Mean-Field methods for Structured Deep-Learning in Computer Vision

THÈSE N^{O} 8677 (2018)

PRÉSENTÉE LE 6 JUILLET 2018

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE VISION PAR ORDINATEUR
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Pierre Bruno BAQUÉ

acceptée sur proposition du jury:

Prof. O. N. A. Svensson, président du jury
Prof. P. Fua, Dr F. Fleuret, directeurs de thèse
Dr V. Koltun, rapporteur
Prof. P. Kumar, rapporteur
Prof. K. Schindler, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Acknowledgements

I would like to thank my thesis advisors Dr. François Fleuret and Prof. Pascal Fua for providing me the best guidance and giving me the opportunity to work in such a fertile environment. I would like to extend my thanks to the members of my thesis committee, who make me the honor to evaluate this thesis.

My special thanks go to my beloved Alizée for giving me strength and faith in the future. This work would not have even started without her. I also thank my parents, sister and family for their unconditional love and support. This thesis is also theirs.

I thank my friends from Toulouse, Paris, Boston and London.

Finally, I want to thank all my friends, from the CVLab and from the different running and cycling groups who contributed to make these last four years among the best in my life.

Abstract

In recent years, Machine Learning based Computer Vision techniques made impressive progress. These algorithms proved particularly efficient for image classification or detection of isolated objects. From a probabilistic perspective, these methods can predict marginals, over single or multiple variables, independently, with high accuracy.

However, in many tasks of practical interest, we need to predict jointly several correlated variables. Practical applications include people detection in crowded scenes, image segmentation, surface reconstruction, 3D pose estimation and others. A large part of the research effort in today's computer-vision community aims at finding task-specific solutions to these problems, while leveraging the power of Deep-Learning based classifiers. In this thesis, we present our journey towards a generic and practical solution based on mean-field (MF) inference.

Mean-field is a Statistical Physics-inspired method which has long been used in Computer-Vision as a variational approximation to posterior distributions over complex Conditional Random Fields. Standard mean-field optimization is based on coordinate descent and in many situations can be impractical. We therefore propose a novel proximal gradient-based approach to optimizing the variational objective. It is naturally parallelizable and easy to implement. We prove its convergence, and then demonstrate that, in practice, it yields faster convergence and often finds better optima than more traditional mean-field optimization techniques.

Then, we show that we can replace the fully factorized distribution of mean-field by a weighted mixture of such distributions, that similarly minimizes the KL-Divergence to the true posterior. Our extension of the clamping method proposed in previous works allows us to both produce a more descriptive approximation of the true posterior and, inspired by the diverse MAP paradigms, fit a mixture of mean-field approximations. We demonstrate that this positively impacts real-world algorithms that initially relied on mean-fields.

One of the important properties of the mean-field inference algorithms is that the closed-form updates are fully differentiable operations. This naturally allows to do parameter learning by simply unrolling multiple iterations of the updates, the so-called back-mean-field algorithm. We derive a novel and efficient structured learning method for multi-modal posterior distribution based on the Multi-Modal Mean-Field approximation, which can be seamlessly combined to modern gradient-based learning methods such as CNNs.

Finally, we explore in more details the specific problem of structured learning and prediction for multiple-people detection in crowded scenes. We then present a mean-field based structured deep-learning detection algorithm that provides state of the art results on our new and challenging Wildtrack dataset.

Keywords: mean-field inference, structured learning, conditional random fields, multi-modal, computer-vision, detection

Résumé

Les techniques d'apprentissage automatique utilisées en vision par ordinateur ont connu des progrès surprenants depuis une décennie. Ces algorithmes se sont révélés particulièrement performants pour la classification d'images et la détection d'objets isolés. Du point de vue du statisticien, ces méthodes permettent de prédire des distributions marginales selon une ou plusieurs variables, de façon indépendantes et avec une grande fiabilité.

Cependant, de nombreuses tâches ayant un intérêt pratique, nécessitent la prédiction conjointe de plusieurs variables fortement corrélées qui sont, liste non exhaustive : la détection de personnes dans des scènes denses, la segmentation d'image, la reconstruction de surface, l'estimation de pose en 3D et d'autres encore. Une grande partie de l'effort de recherche dans notre domaine, est consacrée à l'élaboration de solutions spécifiques à chacun de ces problèmes, tout en s'appuyant sur des outils de classification de base fondés sur des réseaux de neurone profonds. Cette thèse propose de suivre notre cheminement vers une solution pratique et générique basée sur les algorithmes d'inférence en champ-moyen (CM).

Le calcul en champ-moyen est une approche inspirée de la physique statistique, classiquement utilisée en vision par ordinateur pour approximer des distributions postérieures complexes définies par des champs aléatoires conditionnels (CAC). Les méthodes classiques d'inférence en champ-moyen sont basées sur une descente par coordonnées et sont souvent trop coûteuses à utiliser en pratique. Nous introduisons donc un nouvel algorithme d'inférence en champ-moyen pour des CAC arbitraires. Notre approche a de meilleures propriétés de convergence, est mieux comprise d'un point de vue théorique et peut facilement être implémentée sur des infrastructures de calcul parallèle. Après avoir prouvé la convergence de notre méthode, nous montrons qu'elle permet, sur des problèmes concrets, de trouver de meilleures solutions que les méthodes traditionnelles.

Dans un second temps, nous établissons qu'il est possible de remplacer l'approximation en champ-moyen par une superposition de telles distributions, qui comme dans le cas CM, minimise la divergence-KL par rapport à la distribution postérieure. Cette extension de la méthode de clamping, nous permet d'obtenir une approximation plus fidèle du postérieur. Notre algorithme permet de calibrer une superposition d'approximations en champ moyen, étendant ainsi l'approche des Maximum-A-Posteriori variés. Nous prouvons que notre méthode apporte une amélioration pratique significative aux algorithmes pré-existants utilisant les champ-moyens.

L'approche de champ-moyen se distingue par le fait que les itérations utilisées pour le calcul sont des opérations différentiables. Il est ainsi possible d'apprendre les paramètres d'un CAC par rétro-propagation en déroulant les itérations, cette méthode est appelée rétro champ-moyen. Nous proposons une nouvelle méthode pour l'apprentissage structuré de distributions multi-modales, basée sur l'approximation en Champ-Moyen Multi-Modal. Celle-ci peut être facilement combinée à des méthodes d'apprentissage par gradient, telles que les réseaux de neurones à convolution.

Enfin, nous nous concentrerons sur le problème de l'apprentissage structuré dans le cadre de la détection des personnes dans des foules. Nous présenterons un nouvel algorithme, basé sur ces travaux, qui fournit des résultats supérieurs à l'état de l'art dans le domaine.

Mots-Clés : champ-moyen, apprentissage structuré, champs aléatoires conditionnels, multi-modal, vision par ordinateur, détection

Contents

List of Figures

List of Tables

1 Introduction

Structured prediction tasks are ubiquitous to computer vision. Indeed, in many problems where we need to predict several variables jointly, the correlations between the variables may matter more than their individual values. This is true to tasks such as image segmentation, people detection, curvilinear structure delineation or surface reconstruction. Standard modern machine learning techniques such as Deep Neural-Networks, are not explicitly designed to take into account such correlation. Since this is an important limitation, task-specific solutions have been developed over the years to solve it. However, they are often problem focused and do not generalize to other tasks.

Tools from theoretical computer science and machine learning, such as Conditional Random Fields (CRF) were applied to structured prediction. One of many such approaches was to use mean-field inference (MF) to approximate complex posterior distributions defined through CRFs. This works well in some cases but in some others the algorithm does not converge or the naive mean-field approximation is too simplistic.

Data and machine learning can be used to learn the parameters of the CRF which best model the problem at hand. Mean-field inference and back-propagation through unrolled iterations can be employed to infer the best parameters. However, such algorithms were inherently limited by the failure modes of the mean-field algorithm and no generic solution to the structured learning problem based on mean-field had been proposed yet.

In this thesis, we first use mean-field for prediction tasks where the structure is enforced using a manually defined Conditional Random Fields (CRF). We improve the standard

MF algorithm in two different ways. First, we improve the convergence properties of mean-field algorithms for arbitrary potentials. We propose a new MF algorithm that is faster, easily parellelizable and leads to better performances than previous ones. We then propose a Multi-Modal Mean-Field method which extends the standard MF algorithm. By fitting a mixture of fully factorized distribution instead of a single one, we obtain a better approximation to multi-modal posteriors. Finally, we show that these newly introduced tools can help us learn the CRF parameters directly from data.

Since the recent practical success of Deep-Neural Networks in computer vision, the necessity of using Conditional Random Field structures and mean-field inference has been rightfully questioned. We explain below what benefits CRFs can bring in modern computer vision applications. Furthermore, we discuss how the underlying ideas behind mean-field inference can be put to use. We provide details regarding when and where such approaches should be considered.

Enforcing prior knowledge Humans should be able to guide the learning algorithms with prior knowledge. Conditional Random Fields and Probabilistic Graphical Models, can be used by the programmer to input human knowledge about inter-variable relationships in the following ways:

- As human developers and for many tasks of practical interest, we naturally know how to write equations to discriminate between good and bad solutions. It is therefore more intuitive to define the quality of a solution through a scoring or *energy* function, than to manually design an algorithm which directly produces the solution. We can then use off-the-shelf inference algorithms, such as the mean-field one, to obtain solutions through energy minimization.
- In computer vision, the desired energy model often possesses physical local invariance properties. Such prior information can be conveyed by using a graphical structure in the definition of the energy function.
- As explained in more details below, graphical models make it possible to enforce conditional independence properties on the variables. This is another form of prior knowledge that is used to guide the structured prediction or learning tasks.

In all three cases, the pre-defined energy functions can either have all their parameters set manually or some or all of these parameters can be undetermined a priori.

Then, if training data is available, inference can be combined with a structured learning algorithm. This makes it possible to optimize the parameters to best model the data and therefore learn a meaningful model. As we will see below, mean-field is a very powerful way to learn the parameters of the pre-defined CRF.

Mean-Field as adaptive filtering In effect, parallel mean-field inference, similarly to Convolutional Neural Networks, uses a sequence of parallel updates of variables, written as a linear function of the neighbor's features, followed by a softmax non-linearity. This will be discussed extensively in Chapter 3.2.2.

However, there is a notable difference between mean-field and CNN operations. Namely, standard CNNs use the same linear operation to update each variable with respect to its neighbors. Mean-field updates, on the other hand are used to modulate the linear update functions differently for every vertex, through a potential function.

Similar ideas appeared recently in a more general context through adaptative filtering for graph CNNs [Kipf and Welling, 2016, Simonovsky and Komodakis, 2017]. However, mean-fields provide a robust and energy-based way to enforce prior knowledge in the structure of such filters [Krähenbühl and Koltun, 2012].

CRFs for structured learning Conditional Random Fields and Energy based models can be used to model multi-variate probability distributions, when the predicted variables are non-independent.

Let **I** denote the input variable vector, an image in vision for instance and **X** the vector of output variables that we want to predict.

Training methods for Convolutional Neural Networks minimize a predefined fixed loss function, which is usually a L2-loss for a regression task or a Cross-Entropy one for classification. Importantly, this loss-function is separable over variables, for instance,

$$\mathscr{L} = \sum_{(\mathbf{I}^i,\mathbf{X}^i)\in\mathscr{D}} \sum_k -\log\left(\mathscr{F}(\mathbf{I}^i)_{l=X^i_k}\right), \tag{1.1}$$

for categorical variables.

In essence, this means that the network is optimized to predict the mean of continuous variables or the marginals of categorical ones, given the input data. If the output of the network is used directly for prediction tasks, for instance to produce image segmentation, shape reconstruction, multi-object detection and others, this will only be valid if the posterior over $\mathbf{X}$ is such that its multiple components are independent, given the input. In other terms, that $P(\mathbf{X}|\mathbf{I})$, is a fully-factorized distribution.

For many practical applications, this assumption does not hold, as several valid answers can co-exist for a given question. This is what we call ambiguities or *Multi-Modal* posteriors. Examples of such problems range from segmentation of linear structures such as roads or neurons, detection, 3D pose estimation or surface reconstruction. In that case, predicting marginals or mean estimates and using them to produce outputs, do not provide the expected results. This can translate into over-smooth segmentations on top of which an inference method, based on mean-fields [Krähenbühl and Koltun, 2012] or Graph-Cuts, has to be applied. For detection, it is almost always necessary to use a Non-Maximum suppression algorithm [Ren et al., 2015], which can be interpreted as a greedy form of inference, or to use mean-field inference [Baqué et al., 2017a]. For multi-people pose estimation most methods also use a CRF-based post processing using graph-cuts [Pishchulin et al., 2016].

In some other cases, the true posterior is actually fully factorized, but the Neural Network has not enough capacity to extract and convey all the information contained in $\mathbf{I}$. Therefore even in the extreme case where $P(\mathbf{X}|\mathbf{I})$ should be a Dirac distribution – and therefore fully-factorized –, the same distribution, conditioned on the features $feat(\mathbf{I})$, extracted by the network, $P\left(\mathbf{X}|feat(\mathbf{I})\right)$, may not be fully factorized. Therefore a probabilistic model for the correlations between output variables is needed. The augmentation of the capacity of the Neural Networks, used for semantic segmentation for instance, led to a reduction of the performance gain brought by CRFs in recent years. This is evidenced by the numbers reported in Krähenbühl and Koltun [2012] and Chen et al. [2017], where the former was published 5 Years before the later.

It is often useful to learn the parameters, or in other terms, the CRF's potentials, to make them fit a dataset. We will see in this thesis how mean-fields provide a convenient and efficient framework to learn these parameters, even if they are embedded in a Deep architecture.

- $\|.\|$ is the Euclidean norm in $\mathbb{R}^N$.
- For a differentiable function f, ∇f its gradient.
- **X** is a multivariate random variable composed of the random variables $\{X_1, \ldots, X_N\}$
- **I** is a vector of variables, usually an image, that is the input of our prediction models.
- P_θ is a probability distribution parametrized by the vector of variables θ.
- $P_\theta(. \mid \mathbf{I})$ is a conditional distribution with input **I**.
- If f is a functions and Q a probability distribution $\mathbf{E}_{\mathbf{X}\sim Q}[f(\mathbf{X})]$, is the expectancy of $f(\mathbf{X})$ where **X** follows the distribution Q.
- Z_θ is the partition function of the Boltzmann distribution associated to energy E_θ.
- Q is a probability distribution on N independent Bernoulli variables $\{X_1, \ldots, X_N\}$.
- $\mathscr{L}_\theta$ denotes the log-likelihood loss function used for statistical learning.
- T is a temperature parameter for a Boltzmann distribution in the sense of statistical physics.
- $\mathrm{KL}(Q\|P)$ is the Kullback-Leibler divergence between distributions Q and P.

Table 1.1 – Notations

2 Background and related work

Structured learning is ubiquitous to practical applications. Even though this problem has attracted less interest in recent years, many approaches were proposed. Some of the most successful and elegant solutions to it leverage probabilistic graphical models (PGMs).

Throughout this chapter, we will provide a review of PGMs and Conditional Random Field models. We will go through a discussion of the challenges related to inference and parameters learning. Furthermore, we will explain some of the existing approaches to solving them.

We will see why traditional Deep Learning methods are not directly adapted to structured learning problems. Since this is a major shortcoming, many solutions have been proposed in recent years, and we will explore some of them. In particular, we will focus on previous works that combined Deep Learning with CRF models.

Finally, we will review practical applications of the pre-cited methods for computer vision problems.

2.1 Structured Learning

Machine Learning automatically discovers a model which to explain a set of observations, in order to be able to predict new ones.

2.1.1 Learning and Inference

In this thesis, we will assume a *supervised* learning setting. In other terms, we assume that we are given a data-set of observations $\mathscr{D} = \{(\mathbf{x}_s, \mathbf{I}_s)\}_{s=1...D}$, composed of D training samples. For every $\mathbf{I}_s$, usually an input image in computer vision, we observe a corresponding output vector $\mathbf{x}_s$.

In the *parametric learning* setting that is studied in this thesis, we assume that we are able to design a family of parametric conditional distributions $\{P_\theta(.\mid \mathbf{I})\}_{\theta\in\mathscr{P}}$, parametrized by a multivariate vector θ in a parameter space $\mathscr{P}$. The choice of the relevant family of distributions is based on prior knowledge about the problem, as discussed below.

Our goal is then to find a value of θ, such that $\mathbf{x}_s \sim P_\theta(\mathbf{X}|\mathbf{I}_s)$ is a plausible model for the observed samples in $\mathscr{D}$.

One of the most classical approaches is to set it to the maximum likelihood estimator

$$\begin{aligned}\theta &= \underset{\theta}{\operatorname{argmax}} \log\left(\prod_{s=1...D} P_\theta(\mathbf{X}=\mathbf{x}_s|\mathbf{I}_s)\right) \\ &= \underset{\theta}{\operatorname{argmax}} \sum_{s=1...D} \log\left(P_\theta(\mathbf{X}=\mathbf{x}_s|\mathbf{I}_s)\right) ,\end{aligned} \tag{2.1}$$

which is the value of θ such that the probability of observing the outputs, given the inputs, is maximized. In this thesis, we will call the objective function of Equation 2.1, *Loss* function, and the terms of the sample-wise decomposition the *Sample-Loss*. We will therefore write

$$\mathscr{L}_\theta = \sum_{s=1...D} \mathscr{L}_{\theta\,s} .$$

Once θ is chosen, for any new input $\mathbf{I}$ we can produce an estimate of the corresponding $\mathbf{x}$, either via *sampling*

$$\mathbf{x} \sim P_\theta(\mathbf{X}|\mathbf{I}) , \tag{2.2}$$

or via *Maximum-a-posteriori*

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmax}} P_\theta(\mathbf{X} = \mathbf{x}|\mathbf{I}) \,, \tag{2.3}$$

which corresponds to the most likely configuration of the output.

In the field of statistical learning, the task of choosing θ, as in Equation 2.1, given a dataset $\mathcal{D}$, is called *learning*. The one of sampling or computing the Maximum-A-Posteriori, as in Equation 2.3, is called *inference*. As we will see below, these are two closely intricate tasks and a *learning* algorithm relies on an *inference* one.

2.1.2 Univariate Learning

For many problems of practical importance, the output **X** is actually a single variable.

For instance, in the standard regression problem, **X** is a single real random variable X. We can then choose a family of distributions P_θ in the Gaussian form

$$P_\theta(X = x|\mathbf{I}) = \frac{1}{\sqrt{2\Pi}\sigma} \exp\left(-\frac{(x - f_\theta(\mathbf{I}))^2}{2\sigma^2}\right) , \tag{2.4}$$

where $f_\theta(\mathbf{I})$, is a parametric function and σ an arbitrary parameter, fixed or learned with θ. $f_\theta(\mathbf{I})$ is typically a linear function for the linear regression problem, or a Convolutional Neural Networks in other cases such as [Baqué et al., 2018].

Another standard task is the classification one. It has attracted a lot of attention in the last decade. Image classification has become the *de-facto* standard benchmark to compare Convolutional Neural Network (CNN) architectures in computer vision. There, **X** is a categorical variable that takes values in $\{1, \ldots, L\}$, or Bernouilli variable if $L = 2$. In many popular machine Learning models, the parametric family $\{P_\theta\}_{\theta \in \mathcal{P}}$ is taken to be such that

$$P_\theta(\mathbf{X} = l) = \frac{\exp(f_\theta(\mathbf{I})_l)}{\sum\limits_{k=\{1,\ldots,L\}} \exp(f_\theta(\mathbf{I})_k)} , \tag{2.5}$$

where, again $f_\theta(\mathbf{I})$ is typically a linear function in the logistic regression model. In many applications, $f_\theta(\mathbf{I})$ takes the form of a Deep-Convolutional Neural Networks as in the seminal works of [Krizhevsky et al., 2012] and [He et al., 2016].

In this setting, the inference task is trivial. For instance, in the case of the regression model of 2.4,

$$\underset{\mathbf{x}}{\operatorname{argmax}} P_\theta(\mathbf{X}=\mathbf{x}|\mathbf{I}) = f_\theta(\mathbf{I})\,,$$

and in the classification one of 2.7,

$$\underset{\mathbf{x}}{\operatorname{argmax}} P_\theta(\mathbf{X}=\mathbf{x}|\mathbf{I}) = \underset{k}{\operatorname{argmax}} f_\theta(\mathbf{I})_k\,.$$

2.1.3 Multivariate Learning

Many tasks of practical interest necessitate to predict several values at the same time. In other terms, the realizations of our random variable **X**, are not any more single scalar values or single categorical variables, but vectors of size N. Examples of such regression problems range from depth estimation [Eigen et al., 2014] to pressure prediction in Computational Fluid Dynamics [Baqué et al., 2018]. Moreover, standard examples of such classification problems are semantic segmentation [Long et al., 2015] and detection [Ren et al., 2015].

Since that the random variable **X** is vector-valued, the distribution $P_\theta(\mathbf{X}|\mathbf{I})$ has to be a multi-variate one. For regression tasks, a simple approach, which is often used and sometimes works in practice, is to look for P_θ in the family of fully-factorized distribution. In such a case, we take

$$P_\theta(\mathbf{X}=\mathbf{x}\,|\,\mathbf{I}) = \prod_{i\in 1\ldots N} \frac{1}{\sqrt{2\Pi}\sigma_i} \exp(-\frac{\left(x_i - f_\theta(\mathbf{I})_i\right)^2}{2\sigma_i^2})\,, \tag{2.6}$$

where $f_\theta(\mathbf{I})_i$ is a different parametric function for every element i of the output vector.

For classification problems, the same idea applies and we use

$$P_\theta(\mathbf{X}=\mathbf{x}\,|\,\mathbf{I}) = \prod_{i\in 1\ldots N} \frac{\exp(f_\theta(\mathbf{I})_{i,l=x_i})}{\sum\limits_{k=\{1,\ldots,L\}} \exp(f_\theta(\mathbf{I})_{i,k})}\,. \tag{2.7}$$

In both cases, the *sample loss-functions* of Equation 2.1, decompose naturally into a sum of elementary terms, which can be optimized.

Importantly, note that the functions $f_\theta(\mathbf{I})_{i,l=\mathbf{x}_i}$, share a common set of parameters θ. For, instance, for semantic segmentation, every pixel would be classified at the end of a large Convolutional Neural Network. Therefore, features used for the final prediction, are shared between variables.

Therefore the underlying probabilistic model does not assume conditional independence between the variables,

$$P(\mathbf{X}=\mathbf{x}) \neq \prod_{i\in 1\ldots N} P(X_i = x_i)\,.$$

Only their independence given an input **I**,

$$P(\mathbf{X}|\mathbf{I}) = \prod_{i\in 1\ldots N} P(X_i|\mathbf{I})\,,$$

is assumed under this class of models.

This assumption is very restrictive on the family of probability distributions that the network can actually model. The reader can easily think of failure modes of this approach and we will discuss it in more details later.

Another correct point of view is to say that the learned probabilities are simply marginal probabilities

$$P(X_i|\mathbf{I}) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots x_N\in\mathcal{X}} P(\mathbf{X}=x_1,\ldots,x_{i-1},X_i,x_{i+1},\ldots x_N|\mathbf{I})\,.$$

Hence, for a regression task, $f_\theta(\mathbf{I})_i$ will tend to predict the mean values for variable X_i, given I_i. Similarly, for classification, we obtain *marginal* distributions through $\exp(f_\theta(\mathbf{I})_{i,k})$

2.1.4 Structured Learning

In many cases of interest, the conditional independence assumption of the output variables contained in $\mathbf{X}$, does not hold.

Therefore, the Maximum-A-Posteriori inference, or sampling, cannot be carried out independently for all variables using the learned marginals,

$$\underset{\mathbf{x}}{\operatorname{argmax}} P_\theta(\mathbf{X} = \mathbf{x}|\mathbf{I}) \neq \underset{\mathbf{x}}{\operatorname{argmax}} \sum_{i=1,\dots,N} P_\theta(X = x_i|\mathbf{I}) \,,$$

and the difference can be very large. Similarly, sampling from marginal distributions independently may not be representative of true samples drawn from $P_\theta(\mathbf{X}|\mathbf{I})$.

Therefore, we can define *structured prediction* as the task of sampling or finding the most likely configuration of an output which is composed of multiple strongly correlated variables. This task corresponds to an *inference* problem with a non-fully factorized distribution $P_\theta(\mathbf{X}|\mathbf{I})$. *Structured learning* is the task of learning this distribution $P_\theta(\mathbf{X}|\mathbf{I})$ from data.

Since it is an important problem, over the years, many approaches were developed. Earlier attempts include Structured Support Vector Machines [Taskar et al., 2005], which are explained in more details in [Bakir et al., 2007].

Temporal models such as Hidden Markov Models [Rabiner, 1990] or Recurrent Neural Networks [Hochreiter and Schmidhuber, 1997], can be seen as structured learning and prediction algorithms. However, this thesis is considering applications in the domain of computer vision and we will omit works on sequence modeling.

More recently, two Neural Network based sampling models were introduced. Variational Auto-Encoders (VAEs) [Kingma and Welling, 2014] and Generative Adversarial Networks (GANs) [Goodfellow et al., 2014], allow to learn a latent encoding space for a class of images, where sampling can be performed to reconstruct realistic looking images.

In both approaches, P_θ is not defined directly, but rather through a sampling model $\mathbf{X} \sim f_{g_\theta}(\mathbf{Y})$, where f_{g_θ} is a neural network, called decoder or generator and $\mathbf{Y}$, a multivariate independent Gaussian distribution. Since f_{g_θ} is non-invertible, computing directly $P_\theta(\mathbf{X}_s)$ for a sample $\mathbf{X}_s$, is not tractable. Therefore, optimizing P_θ to maximize the log-likelihood of Equation 2.1, is even less tractable. Each of the two models propose an approximation to it.

VAEs propose to use the standard technique of variational approximation – which will be explained in more details in Section 2.3 – over the distribution of latent variables. The main advantage of their approach is that the variational approximation is computed by another Neural Network that is trained explicitly to provide tight approximation bounds.

GAN methods use a different approach. One of the most successful version of this algorithm, the WGAN of Arjovsky et al. [2017], aims at minimizing the Wasserstein distance between P_θ and the empirical data distribution. It leverages the Kantorovich-Rubinstein duality theorem [Villani, 2008] to do so using a *discriminator* neural network f_d and an adversarial competition between f_d and f_g. The *discriminator* learns a discrepancy function to discriminate between ground-truth samples and the ones that were sampled by the generator. The Energy-Based GAN (EBGAN) of Zhao et al. [2016], is related to the CRF framework that we describe below. Indeed, EBGANs use a discrepancy function which takes the form of a Boltzmann probability distribution defined by an Energy function. The Energy function is also used in CRFs to weight the likelihood of a new sample under the learned model.

Both of the pre-cited classes of approach, namely GANs and VAEs, were impressively powerful for image generation tasks Goodfellow et al. [2014] or, more recently surface mesh reconstruction Bagautdinov et al. [2018]. However, despite several attempts, they were less successful at solving more formal, measurable computer vision tasks such as Semantic Segmentation Luc et al. [2016].

However, one of the most popular approaches to structured learning remains the Probabilistic Graphical Models (PGMs) one, which is thoroughly described in Koller and Friedman [2009]. As explained below in more details, such models can be used to represent a complex family of distributions P_θ.

2.2 Conditional Random Fields

Probabilistic Graphical Models (PGMs) are used in computer science to represent and compute probability distributions over multiple variables. By representing random variables as nodes in a graph the practitioner can input prior knowledge about the structure of the distribution of interest, which is particularly convenient in a scarce data situation. The structure of the graph translates conditional independence properties between variables. Furthermore, the graphical model's sparse structure makes it possible to design efficient inference and learning algorithms, even for graphs with a very large number of variables.

The class of PGMs includes two sub-classes of models, Bayesian Networks (BNs) and Conditional Random Fields (CRFs). They respectively correspond to directed and undirected graphical representations. BNs are a powerful model, which induce a hierarchy between variables, where P_θ is defined by a sequence of conditional probabilities. However in this thesis, we focus on parameter learning for CRFs.

2.2.1 Definition

Recall that $\mathbf{X} = (X_1, \ldots, X_N)$ represents prediction variables and $\mathbf{I}$ an input, usually an image in Computer Visions. A CRF relates the ones to the others via a posterior probability distribution

$$P(\mathbf{X}) = \frac{1}{Z} \prod_{c \in \mathscr{C}} \psi_c(\mathbf{X}_c \mid \mathbf{I}) \,, \tag{2.8}$$

which is often rewritten in the exponential form as

$$P(\mathbf{X} \mid \mathbf{I}) = \exp\left(-E(\mathbf{X} \mid \mathbf{I}) - \log(Z(\mathbf{I}))\right) \,, \tag{2.9}$$

where $E(\mathbf{X} \mid \mathbf{I})$ is an energy function that can be decomposed into a sum

$$E(\mathbf{X} \mid \mathbf{I}) = \sum_{c \in \mathscr{C}} \boldsymbol{\phi}_c(\mathbf{X}_c | \mathbf{I}) \,, \tag{2.10}$$

where $\mathscr{C}$ is a subset of indices in $\{1,\ldots,N\}$, called graph-cliques. $\boldsymbol{\phi}_c(\mathbf{X}_c|\mathbf{I})$ are locally defined functions which take as input the values of the subset of variables $\mathbf{X}_c$, where

$$\mathbf{X}_c = \{X_j\}_{j\in\mathscr{C}} .$$

The functions $\boldsymbol{\phi}_c(\cdot|\mathbf{I})$, are called *potential functions*.

Finally, $A(\mathbf{I}) = \log(Z(\mathbf{I}))$ is the log-partition function that normalizes the distribution.

We will sometimes omit the dependency with respect to **I** and if the potential functions do not depend on an external input **I**, then the model is often called a *Markov Random Field*.

In this thesis, we will use the following terminology to define potential functions ϕ_c according to the size of the corresponding clique c

- **Unary potentials** : $|c| = 1$
- **Pairwise potentials** : $|c| = 2$
- **High-Order potentials** : $|c| \geq 3$

2.2.2 Graphical representation and properties

The CRF is often very conveniently represented as a graph. Generic CRFs use a specific form, which is called *factor graph* in order to account for Higher-Order potentials with clique size greater than two. As illustrated in Figure 2.1, a factor graph is a bipartite graph where variable nodes are represented by circles and potential nodes are represented by squares, called factors. Every factor node corresponds to a potential ϕ_c, and is connected to all the variable nodes in the clique c.

Several interesting properties can be derived from the definition of the CRF.

In particular, let us consider two nodes A and B, associated with variables X_A and X_B. Let us assume that **C** is a subset of nodes that forms a vertex-cut, which disconnects

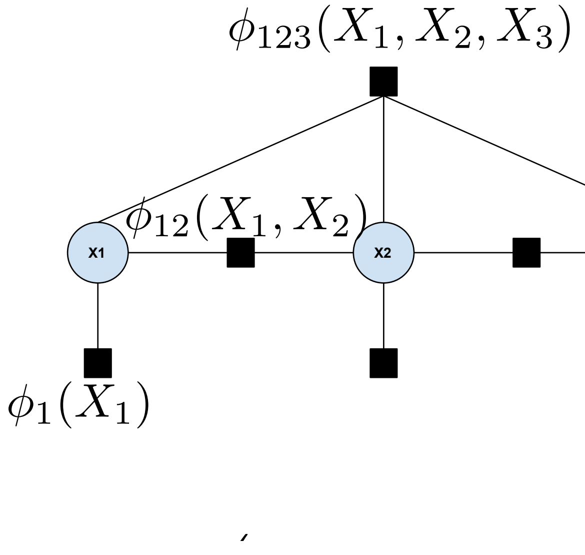


Figure 2.1 – Factor graph representation.

nodes A and B. Then, the following conditional independence property holds:

$$P(X_A, X_B | X_{\mathbf{C}}) = P(X_A | X_{\mathbf{C}}) P(X_B | X_{\mathbf{C}}) \,,$$

where $X_{\mathbf{C}}$ is the set of variables associated to the nodes in **C**.

Consequently, a variable X_A, is independent of any other variable X_B given $X_{\mathbf{C}}$ if **C** contains the set of neighbors of A in the graph.

2.2.3 Exponential family representation and duality

The exponential family is a large class of probability distributions, which is widely used for graphical models.

There are two main representations of exponential family distributions, and both correspond to parameters which are dually related and equivalent. The first representation is given as an exponential potential, parametrized by θ and is a specific form of the one of Equation 2.2. ϕ is the vector of sufficient statistics, which characterizes the family of distributions we are working with. $A(\theta)$ is the log-partition function, which normalizes the distribution

$$P_\theta(\mathbf{X}=\mathbf{x}) = \exp\left(<\phi(\mathbf{x}),\theta> - A_\theta\right) . \tag{2.11}$$

For instance, in the case of a pairwise MRF, we have unary terms and pairwise terms such that :

$$<\phi(\mathbf{x}),\theta>= \sum_{\substack{i=1,\dots,N\\k=1,\dots,L}} x_{i,k}\theta_{i,l} + \sum_{\substack{i=1,\dots,N\\j=1,\dots,N\\k=1,\dots,L\\m=1,\dots,L}} x_{i,k}x_{j,l}\theta_{i,j,k,l}$$

The second representation of the exponential family distribution is the moments representation. We denote by $\mathbf{X}_\mu$ the random variable with sufficient statistics ϕ such that

$$E[\phi(\mathbf{X}_\mu)] = \mu .$$

Interestingly, as explained before, for a binary pairwise CRF, the sufficient statistics is the vector $\phi(x) = (x_{i,l}, x_{i,j,k,l})$. Therefore, the probability distribution is naturally represented in terms of expectancies, variance and covariances of individual variables. Both representations are dually related through a Legendre transform. These ideas are used to travel between both representations throughout the thesis.

More precisely, the Legendre Transform $A^*(\mu)$ of $A(\theta)$ is also the negative entropy of the variable $\mathbf{X}_\mu$ under P_θ. Concretely, it means that :

$$A_\theta = sup\{<\theta,\mu> - A^*(\mu)\}, \mu \in M , \tag{2.12}$$

where M is the set of all *realizable* moment parameters μ. Here, *realizable* means that they respect basic properties of probability distributions about normalization and marginalization. The moment representation corresponds to the value of μ which achieves the optimum in the Legendre transform of Equation 2.12. The inference task described in Section 2.3, can be interpreted as switching from one representation to the other.

2.3 Inference

Let us further assume that P_θ is a probability distribution defined by a CRF, as in Equation 2.4.1. As discussed in 2.1, one of the main challenges of structured learning is the inference one.

Note that, the seemingly simple definition of P_θ from Equation 2.4.1 hides a major difficulty. Indeed, the normalizing partition function Z, is actually computed as the sum of an exponentially large number of terms as

$$Z = \sum_{\mathbf{x} in \mathcal{X}} exp\left(-E(\mathbf{x})\right) ,$$

where $\mathcal{X}$ is the set of all possible configurations of **x**. In other terms, if we work with categorical variables that can take L values, then $|\mathcal{X}| = L^N$.

This remark shows that, even computing marginal probabilities

$$P_\theta(X_A = k|\mathbf{I}) ,$$

becomes a challenging problem, which can only be solved by brute force summation for relatively small CRFs.

Similar challenges apply to Maximum-a-Posteriori (MAP) inference, the task of finding the most likely configuration in $\mathcal{X}$.

Because any propositional satisfiability problem can be represented as a factor graph, we know that, in the general case, we can only hope for approximations of the solution to the inference problems. However, for a restricted class of CRFs, the inference problem can be solved exactly. More precisely, for graphical models having a tree-like structure, a simple iterative marginalization technique can be applied to solve exactly the marginal and MAP inference problems. These algorithms are called respectively *sum-product* and *max-sum* algorithms Koller and Friedman [2009].

A slightly more challenging but also tractable case is the one of graph structures which are close to being trees. Indeed, when a graph can be represented as a tree of small cliques, the *sum-product* and *max-sum* algorithms can be applied on an augmented graphical models where original variables are replaced by cliques. This algorithm, called *junction-tree*, eliminates cycles by clustering variables. The maximum size of clusters that have to be considered to transform a CRF into a tree is called *tree-width* and the complexity of exact inference directly depends on it.

2.3.1 Belief propagation

Quite interestingly, the algorithm described above, which was initially designed for graphs with no loops, has been applied successfully in the loopy setting. This leads to a range of algorithms, whose convergence properties are not fully understood, and which only provide an approximation to the marginals. This method has been developed and used in several fields with different names. It is known as the "Bethe-Peierls approximation" in Physics, the "sum-product" (or "max-sum") in computer science and as "Belief Propagation" (BP) by the machine learning community.

Belief Propagation As mentioned above, the "loopy" belief propagation is inspired from a procedure which is exact on tree-like graphical models. This procedure is relatively simple, it is a systematic recipe to marginalise the distribution.

In this section, for clarity, a denotes a factor index from the graphical model and ∂a, the variable indices in the corresponding clique, or in other terms the adjacent variable nodes in the factor graph representation.

Let us assume for the moment that our probability distribution $\mu(\mathbf{x})$ replaces $P(\mathbf{x})$ in 2.8 and that the corresponding factor graph is a tree. The main, and most important ingredient of the BP algorithm is the set of messages $\nu_{i\to a}$ and $\widehat{\nu}_{a\to i}$ which are exchanged between variable and factor nodes.

These messages are also “local” probability distributions, defined over a single variable x_i. A message $\nu_{i\to a}$ from a variable to a factor node is the marginal distribution of x_i on a “modified” model, where the factor node a has been removed. Similarly, a message $\widehat{\nu}_{a\to i}$ from a factor to a variable node is the marginal of x_i on the model where all factors adjacent to x_i but a have been removed. Intuitively, on trees, this corresponds to marginals coming from different parts of the tree, and the true marginal of x_i can be computed as:

$$\mu(x_i) \propto \prod_{a\in\partial i} \widehat{\nu}_{a\to i}(x_i) \tag{2.13}$$

Using basic marginalization properties of probability distributions, one can easily show that the messages obey to the following fixed point equations:

$$\begin{aligned} \nu_{i\to a} &= \prod_{b\in\partial i\setminus a} \widehat{\nu}_{b\to i}(x_i) \\ \widehat{\nu}_{a\to i} &= \sum_{\mathbf{x}_{\partial a}\setminus i} \psi_a(\mathbf{x}_{\partial a}) \prod_{k\in\partial a\setminus i} \nu_{k\to a}(x_k) \end{aligned} \tag{2.14}$$

Free Energy The free Energy of a system is defined from as its negative log-partition function:

$$F = -T.\log(Z) \tag{2.15}$$

where T is a temperature.

One of the basic results of statistical physics states that the Free Energy is actually the sum of the Expected Energy and the negative entropy:

$$\mathscr{F}(\mu) = \underbrace{\sum_{\mathbf{x}} \mu(\mathbf{x}) \log \prod_{a=1}^{M} \psi_a(\mathbf{x}_{\partial a})}_{\mathscr{E}(\mu)} + \underbrace{T.\sum_{\mathbf{x}} \mu(\mathbf{x}) \log(\mu(\mathbf{x}))}_{-\mathscr{H}(\mu)} \tag{2.16}$$

Bethe Free Energy On tree-like factor graphs, the Bethe Free-Entropy is exactly equal to the Free Entropy. On general graphs, it is only an approximation of it.

The Bethe Free Energy is a function over the set of marginals $\{\mu_i, \mu_a\}$ which are locally consistent

$$\sum_{\mathbf{x}_{\partial a}\setminus i} \mu_a(\mathbf{x}_{\partial a}) = \mu_i(x_i)\,, \tag{2.17}$$

defined as

$$\mathbf{F} = -\sum_{a\in F} \mu_a(\mathbf{x}_{\partial a}) \log \frac{\mu_a(\mathbf{x}_{\partial a})}{\psi_a(\mathbf{x}_{\partial a})} - \sum_{i\in V} (1-|\partial i|)\mu_i(x_i)\log \mu_i(x_i)\,. \tag{2.18}$$

Lemma 1 *Note that the Free Energy can be rewritten in terms of the message passing terms $(\nu, \widehat{\nu})$ instead of marginals μ:*

$$\mathbf{F}(\nu) = -\sum_{a\in F} \mathbf{F}_a(\nu) - \sum_{i\in V} \mathbf{F}_i(\nu) + \sum_{(ia)\in E} \mathbf{F}_{ia}(\nu)\,, \tag{2.19}$$

where

$$\mathbf{F}_a(\nu) = \log\left[\sum_{x_{\partial a}} \psi_a(x_{\partial a}) \prod_{i\in\partial a} \nu_{i\to a}(x_i)\right], \qquad \mathbf{F}_i(\nu) = \log\left[\sum_{x_i}\prod_{b\in\partial i} \widehat{\nu}_{b\to i}(x_i)\right],$$

$$\mathbf{F}_{ai}(\nu) = \log\left[\sum_{x_i} \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)\right].$$

Proof Let us first look at the $\mathbf{F}_a(\nu)$ term.

Looking at the fixed point message passing rule, we know that

$$\sum_{\mathbf{x}_{\partial a}} \psi_a(x_{\partial a}) \prod_{i\in\partial a} \nu_{i\to a}(x_i) = Z_a$$

where Z_a is the local partition function such that $\mu(\mathbf{x}_{\partial a}) = \dfrac{\psi_a(x_{\partial a}) \prod_{i\in\partial a} \nu_{i\to a}(x_i)}{Z_a}$.

Therefore, $\mathbf{F}_a(\nu) = -\log(Z_a)$. Furthermore, we know that for any system, $-\log(Z_a)$ is the free energy function, which can also be expressed as the sum of Energy and negative entropy (see Equation 2.16).

Which means that:

$$\begin{aligned}
\mathbf{F}_a(\nu) &= -\log(Z_a) \\
&= -\sum_{\mathbf{x}_{\partial a}} \mu_a(\mathbf{x}_{\partial a}) \log(\psi_a(\mathbf{x}_{\partial a})) + \sum_{\mathbf{x}_{\partial a}} \mu_a(\mathbf{x}_{\partial a}) \log(\mu_a(\mathbf{x}_{\partial a})) \\
&= \sum_{\mathbf{x}_{\partial a}} \mu_a(\mathbf{x}_{\partial a}) \log\left[\frac{\mu_a(\mathbf{x}_{\partial a})}{\psi_a(\mathbf{x}_{\partial a}})\right]
\end{aligned} \tag{2.20}$$

Now, let us look more precisely at the last term $\mathbf{F}_{ia}(\nu)$. Again, we use the message passing fixed point equation to get:

$$\begin{aligned}
\mathbf{F}_{ia}(\nu) &= \log(\sum_{x_i} \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)) \\
&= \log(\sum_{x_i} \prod_{b\in\partial i\backslash a} \widehat{\nu}_{b\to i}\widehat{\nu}_{a\to i}(x_i)) \\
&= \log(\sum_{x_i} \prod_{b\in\partial i} \widehat{\nu}_{b\to i}) \\
&= \mathbf{F}_i(\nu)
\end{aligned} \tag{2.21}$$

Therefore,

$$-\sum_{i\in V} \mathbf{F}_i(\nu) + \sum_{(ia)\in E} \mathbf{F}_{ia}(\nu) = (|\partial i| - 1)\mathbf{F}_i(\nu)$$

And use for $\mathbf{F}_i(\nu)$ the same method as for $\mathbf{F}_a(\nu)$ to terminate the proof.

To each set of messages corresponds a locally consistent set of marginals that can be computed as

$$\begin{aligned}
\mu_i(x_i) &= \prod_{\{b\in\partial i\}} \widehat{\nu}_{b\to i}(x_i) \\
\mu_a(X_{\partial a}) &= \sum_{\mathbf{x}_{\partial a}} \psi_a(\mathbf{x}_{\partial a}) \prod_{k\in\partial a} \nu_{k\to a}(x_k)\,.
\end{aligned} \tag{2.22}$$

Theorem 2 *There is a one-to-one correspondence between stationary points of the Bethe free entropy function and fixed points of the BP algorithm.*

The first proof of this fact was given in Yedidia et al. [2001].

Proof We provide a proof sketch.

Take the formulation of the Bethe Free Entropy as given in Equation 2.18. We are looking to minimize this expression with respect to the marginals $\mathbf{b}_a$ and $\mathbf{b}_i$. We introduce the Lagrange multipliers to enforce the local consistency of Equation 2.17.

We then differentiate the Lagrangian in order to derive the first order stationarity condition, and obtain a simple condition on the Lagrange multipliers. It turns out, that after an exponential reparametrisation of the Lagrange multipliers, the first order conditions are exactly similar to the BP fixed point equations. In other terms, the BP messages correspond (up to reparametrisation), to the Lagrange multipliers.

Fixed points and clusters. The BP message passing rules and the fact that they lead to exact marginals on tree-like graphs, have been well known for several decades. Their extension to loopy graphs, and the correspondence between BP fixed points and stationary points of the Free entropy are more recent [Yedidia et al., 2001], but this knowledge is widespread within the computer science community. However, people often use BP in our community for loopy graphs, knowing that it might not converge, that several local minima may exist, but without fully understanding when that may be the case. Quite interestingly, some tools from statistical physics let us understand and predict such behaviors.

Indeed, the main assumption that is used to show that the BP equations converge in tree-like graphs is the conditional independence of variables in ∂a, adjacent to a same factor a, when this factor is removed. In large random graphs, when the density of connections is not too large, this assumption is almost verified as neighboring variables, generally become "far apart" when the factor that was linking them is removed. Therefore, if, as it is often the case, correlations between variables decrease on the long range, two variables in ∂a are almost independent once factor a is removed.

In some cases, one can predict how difficult it will be for the BP algorithm to converge or to find a good solution, for an average instance. More precisely, as the connectivity of the graph increases, clusters of local minima of BP equations arise and the optimization through BP iterates becomes more and more hazardous. For more details on the topic, we refer the reader to Mezard and Montanari [2009].

2.3.2 Mean-Field Inference

Mean-Field is another well-known method from statistical physics. In Computer-Science, it has been derived as a special case of *variational inference* (VI) [Kappes et al., 2015]. We will review both techniques below.

Variational Inference Recall that the final goal of parameters learning is, as stated in Equation 2.1 to maximize the probability of the data under the distribution P_θ, with respect to θ. According to Equation 2.8, this is made difficult by the presence of the partition function Z. As studied in section 2.4, one approach to alleviating this problem is to use a *variational upper bound*, using an approximating auxiliary distribution Q, within a tractable, restricted family of distributions.

More precisely, let $\mathscr{Q}$ denote a restricted family of distributions over the same variables $\mathbf{X}$ as P_θ. Furthermore, let us assume the that these distributions are tractable for inference. For instance, they can be fully factorized or be represented by a tree-like CRF. Furthermore, in the derivation we omit the dependence of P_θ in $\mathbf{I}$ for clarity. Then, we can derive the following lower-bound to the log-partition function

$$
\begin{aligned}
A_\theta = \log Z_\theta &= \log \sum_{\mathbf{x}\in\mathscr{X}} exp(-E_\theta(\mathbf{x})) && (2.23)\\
&= \log \sum_{\mathbf{x}\in\mathscr{X}} Q(\mathbf{x}) \frac{exp(-E_\theta(\mathbf{x}))}{Q(\mathbf{x})} && (2.24)\\
&\geq - \sum_{\mathbf{x}\in\mathscr{X}} Q(\mathbf{x}) E_\theta(\mathbf{x}) + \mathscr{H}(Q) && (2.25)\\
&\geq -\mathbf{E}_{\mathbf{X}\sim Q}[E_\theta(\mathbf{x})] + \mathscr{H}(Q) = A_Q\,, && (2.26)
\end{aligned}
$$

where Q is a probability distribution whose support includes the one of P_θ and $\mathscr{H}(Q)$ is its entropy function

$$
\mathscr{H}(Q) = - \sum_{\mathbf{x}\in\mathscr{X}} Q(\mathbf{x}) \log(Q(\mathbf{x}))\,.
$$

Note that we used Jensen's inequality between 2.24 and 2.24.

The approximation error in the log-partition function estimation can then be rewritten as

$$\begin{aligned} A_\theta - A_Q &= \log Z_\theta + \mathbf{E}_{\mathbf{X}\sim Q}[E_\theta(\mathbf{x})] - \mathcal{H}(Q) \qquad (2.27) \\ &= \mathbf{E}_{\mathbf{X}\sim Q}[\log \frac{Q(\mathbf{X})}{P_\theta(\mathbf{X})}] \\ &= \mathrm{KL}(Q \| P_\theta)\,, \qquad (2.28) \end{aligned}$$

where KL denotes the Kullback-Leibler divergence between the variational distribution Q and the original one P.

Variational Inference as distribution approximation The Kullback-Leibler divergence is commonly used to measure distance between probability distributions. More precisely it belongs to the wider family of Bregman divergences [Bregman, 1967] and is defined as

$$\mathrm{KL}(Q||P) = \sum_{\mathbf{x}\in\mathcal{X}} Q(\mathbf{X}=\mathbf{x}) \log \frac{Q(\mathbf{X}=\mathbf{x})}{P(\mathbf{X}=\mathbf{x})}\,. \qquad (2.29)$$

In order to obtain an estimate of the log-partition function that is as accurate as possible, one will be looking for a distribution Q that makes the gap of Equation 2.27 as small as possible. In other terms, Equation 2.28 lets us think of the Variational Inference problem as the one of approximating a complex distribution P_θ by a simpler one Q within a restricted family of distributions $\mathcal{Q}$.

For a given P_θ, the distribution Q_θ stands for

$$Q_\theta = \operatorname*{argmin}_{Q\in\mathcal{Q}} \mathrm{KL}(Q \| P_\theta)\,,$$

which is the optimal variational approximation to P_θ within the restricted family $\mathcal{Q}$.

Note that,

$$\begin{aligned} &\text{if } P_\theta \in \mathcal{Q} \\ &\text{then } P_\theta = Q_\theta \\ &\text{and } \mathrm{KL}(Q_\theta \| P_\theta) = 0\,, \end{aligned}$$

which means that the best approximation to P_θ is itself.

As we will see in this thesis, even when the family $\mathscr{Q}$ is very restricted, Q can sometimes be a good approximation to P_θ. Intuitively, a good Variational Approximation Q_θ will put weight were P_θ already has some weight, but may ignore other likely regions. Figure 2.2, illustrates this fact with Gaussian Approximation.

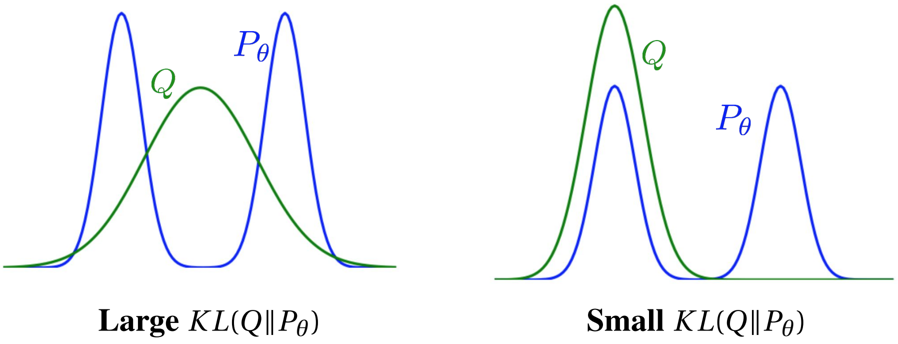


Figure 2.2 – KL-divergence for Gaussian approximation for a mixture of Gaussians P_θ

Computationally, this minimization is only feasible because the term $\log Z_\theta$ in 2.27, does not depend on Q and can therefore be ignored in the optimization process.

Mean-Field inference Mean-Field (MF) inference is a specific form of variational inference for multivariate distributions. The MF algorithms look for an approximation within the restricted family $\mathscr{Q}$ of fully-factorized distributions. Because of the very simple form of the approximating distribution, it is also often called *naive* Mean-Field.

More precisely, recall that we are looking for a probability distribution on a multi-variate variable $\mathbf{X} = (X_1, \ldots, X_N)$. We therefore introduce a distribution Q written as

$$Q(\mathbf{X} = (x_1, \ldots, x_N)) = \prod_{i=1}^{N} Q_i(x_i)\,, \tag{2.30}$$

where $Q_i(\cdot)$ is a mono-dimensional distribution.

For classification types of problems, where each variable X_i is a categorical variable, $Q_i(\cdot)$ is a categorical discrete distribution, which can be parametrized by a vector $\mathbf{q}_i$ of real numbers

$$q_{i,l} = Q_i(X_i = l; \mathbf{q}_i) \text{ for } i \in \{1,\dots,N\}, l \in \{1,\dots,L\}$$

and $\mathbf{q}_i \in \mathcal{M}$ where, $\mathcal{M}$ is the set of parameters verifying

$$\forall i \in \{1,\dots,N\} \sum_{l \in \{1,\dots,L\}} q_{i,l} = 1 \,.$$

The $q_{i,l}$ are estimated by minimizing the KL-divergence of Equation 2.29. To insist on this parametrization, we will indifferently write $Q_i(\mathbf{X}_i; \mathbf{q}_i)$ or $Q_i(\mathbf{X}_i)$.

Since Q is fully factorized, the terms of the KL-divergence can be recombined as a sum of an expected energy, containing as many terms as there are potentials and a convex negative entropy containing one term per variable

$$\mathrm{KL}(Q \| P_\theta) = \sum_{c \in \mathscr{C}} \mathbf{E}_{\mathbf{X} \sim Q}[\phi_c(\mathbf{X})] - \sum_{i=1,\dots,N} \mathscr{H}(Q_i) + \log Z_\theta \,, \tag{2.31}$$

where

$$\mathscr{H}(Q_i) = - \sum_{l \in \{1,\dots,L\}} q_{i,l} \log q_{i,l} \,.$$

We can ignore Z_θ, which does not depend on Q and rewrite the objective function as

$$\mathscr{F}(\mathbf{q}) = \underbrace{-\mathbf{E}_{Q(\boldsymbol{X};\mathbf{q})}[\log P(\mathbf{X} \mid \mathbf{I})]}_{\mathscr{E}(\mathbf{q})} + \underbrace{\mathbf{E}_{Q(\boldsymbol{X};\mathbf{q})}[\log Q(\boldsymbol{X}; \mathbf{q})]}_{-\mathscr{H}(\mathbf{q})} \,, \tag{2.32}$$

which is often called *variational Free-Energy.*

The design of efficient and convergent minimization algorithms for this objective function will be part of the topic of this thesis.

If the variables are continuous ones, we will look for Q in the form of a product of Gaussian densities. In other term, we will then choose to write $Q_i(\cdot)$ as

$$Q_i(x_i) = \frac{1}{\sqrt{2\Pi}\sigma_i} \exp(-\frac{(x_i - \alpha_i)^2}{2\sigma_i^2}) \,,$$

where α_i and σ_i are the parameters to optimize during inference. In this case, Equation 2.31, remains valid, except for the expression of the entropy.

Traditional Mean-Field Algorithm For completeness, we provide a derivation of well-known coordinate descent optimization technique for mean-field updates, similar in spirit to the one of Bishop [2006]. This minimization problem is going to be discussed in more details in Chapter 3, where we derive a new approach.

The traditional Mean-Field algorithm is used to minimize of the variational Free-Energy $\mathscr{F}(\mathbf{q})$ of Equation 2.32, iteratively with respect to $\mathbf{q}$.

The algorithm performs iterations to update a probability distribution Q^t until convergence. $\mathbf{q}_i^t = \{q_{i,1}^t, \ldots, q_{i,L}^t\}$ denotes the subset of parameters corresponding to the variable X_i.

The optimization scheme used is essentially a block coordinate descent over the parameters. Therefore, at iteration t, we choose a variable index i to optimize and the subset of parameters that correspond to all the other variables, which we will denote by $\mathbf{q}_{\setminus i}^t$, remains fixed. At step t we therefore have to solve the simplified optimization problem

$$\begin{aligned} \underset{\mathbf{q}_i}{\text{minimize}} \quad & \mathscr{E}(\mathbf{q}_i, \mathbf{q}_{\setminus i}^t) - \mathscr{H}(\mathbf{q}_i, \mathbf{q}_{\setminus i}^t) \\ \text{subject to} \quad & \sum_l q_{i,l} = 1 \, . \end{aligned} \tag{2.33}$$

Let us first expand the first term of Equation 2.33. We write

$$\begin{aligned} \mathscr{E}(\mathbf{q}_i, \mathbf{q}_{\setminus i}^t) &= -\mathbf{E}_{Q(X;\mathbf{q})}[\log P(\mathbf{X}|\mathbf{I})] \\ &= -\mathbf{E}_{Q(X;\mathbf{q})}\left[\mathbf{E}_{Q(X|\mathbf{q})}[\log P(\mathbf{X}|\mathbf{I})|X_i]\right] \\ &= -\sum_l q_{i,l} \mathbf{E}_{Q(X;\mathbf{q}_{\setminus i})}[\log P(\mathbf{X}|\mathbf{I})|X_i = l] \, . \end{aligned} \tag{2.34}$$

Since $Q(\mathbf{X};\mathbf{q})$ is a product of categorical distributions $Q_i(\mathbf{X}_i;\mathbf{q})$, we can rewrite the second term of Equation 2.33 as

$$\begin{aligned} -\mathscr{H}(\mathbf{q}_i, \mathbf{q}^t_{\setminus i}) &= \sum_{j,l} q_{j,l} \log q_{j,l} \\ &= \sum_{l} q_{i,l} \log q_{i,l} + \underbrace{\sum_{j:j \neq i} \sum_{l} q_{j,l} \log q_{j,l}}_{C_i} , \end{aligned} \tag{2.35}$$

where C_i denotes the constant summand which does not include terms related to X_i.

Let us now define the Lagrangian

$$\begin{aligned} \mathscr{L}(\mathbf{q}_i, \mu_i) &= \mathscr{E}(\mathbf{q}_i, \mathbf{q}^t_{\setminus i}) - \mathscr{H}(\mathbf{q}_i, \mathbf{q}^t_{\setminus i}) - \mu_i(\sum_{l} q_{i,l} - 1) \\ &= -\sum_{l} q_{i,l} \mathbf{E}_{Q(\boldsymbol{X}|\mathbf{q}_{-i})}[\log p(\mathbf{X}|\mathbf{I})|X_i = l] + \sum_{l} q_{i,l} \log q_{i,l} - \mu_i(\sum_{l} q_{i,l} - 1) + C_i \, . \end{aligned} \tag{2.36}$$

where we introduced a dual variable μ_i to account for the optimization constraint. By differentiating with respect to a $q_{i,l}$ we obtain the optimality condition

$$\log q^{\star}_{i,l} = \mathbf{E}_{Q(\boldsymbol{X}|\mathbf{q}_{\setminus i})}[\log p(\mathbf{X}|\mathbf{I})|X_i = l] + \mu_i \, . \tag{2.37}$$

This leads to the standard update rule

$$\forall l \, , q^{\star}_{i,l} \propto \exp\left[\mathbf{E}_{Q(\boldsymbol{X}|\mathbf{q}_{\setminus i})}[\log p(\mathbf{X}|\mathbf{I})|X_i = l]\right] , \tag{2.38}$$

where the normalization constant can be computed from μ_i.

Iteratively applying Equation 2.38 by looping through the variables then guarantees the convergence of $\mathscr{F}$, due to the fact that $\mathscr{F}$ is convex with respect to each $q_{i,l}$ [Bishop, 2006, Baqué et al., 2015].

2.4 Parameters Learning

As discussed in 2.1, we are interested in learning the parameters θ of the distribution $P_\theta(\mathbf{X}|\mathbf{I})$ in order to model a dataset $\{(\mathbf{X}_d, \mathbf{I}_d)\}_{d=1,\ldots,D}$.

When the variables $\mathbf{X}$ that we want to model are multivariate and non-independent, given $\mathbf{I}$, a good way to define an appropriate family of distributions is to use a CRF type of models. Therefore, in this section, we will assume that P_θ takes the form of a CRF as in Equation 2.4.1, conditioned on an input $\mathbf{I}$ and parametrized by θ as

$$P_\theta(\mathbf{X}\,|\,\mathbf{I}) = \exp\left(-\mathscr{E}(\mathbf{X}\,|\,\mathbf{I};\theta) - \log(Z(\mathbf{I};\theta))\right) , \tag{2.39}$$

where $\mathscr{E}(\mathbf{X}\,|\,\mathbf{I})$ is an energy function that can be decomposed into a sum

$$\mathscr{E}(\mathbf{X}\,|\,\mathbf{I};\theta) = \sum_{c\in\mathscr{C}} \boldsymbol{\phi}_c(\mathbf{X}_c|\mathbf{I};\theta) .$$

2.4.1 Maximum likelihood learning

One of the most popular approaches to parameters learning is the maximum likelihood one. In this setting, using the specific form of P_θ defined in Equation 2.39, we will be looking for the maximum likelihood parameter of the loss function $\mathscr{L}_\theta$

$$\underset{\theta}{\operatorname{argmin}} \sum_{s=1\ldots D} -\log(P_\theta(\mathbf{X}=\mathbf{x}_s|\mathbf{I}_s)) .$$

Since there is no closed form solution to this problem, we will be using a gradient-based minimization approach.

$$\nabla_\theta \mathscr{L}_\theta = -\sum_{s=1\ldots D} \nabla_\theta \log(P_\theta(\mathbf{X}=\mathbf{x}_s|\mathbf{I}_s)) . \tag{2.40}$$

For the sake of simplicity, we will further assume that only one sample is given – or that we compute only one term in the gradient of Equation 2.40 –, the full gradient being then recovered by a mere summation. With a very large dataset, or on-line settings, the gradients can be recombined via stochastic gradient descent [Bottou and Bousquet, 2008].

Let us now focus on the challenge of computing the gradient terms

$$\nabla_\theta - \log(P_\theta(\mathbf{X}=\mathbf{x}_s|\mathbf{I}_s)) .$$

According to the specific form of P_θ, given in Equation , we obtain

$$\begin{aligned}\nabla_\theta - \log\left(P_\theta(\mathbf{X} = \mathbf{x}_s|\mathbf{I}_s)\right) &= \nabla_\theta \mathscr{E}(\mathbf{x}_s \mid \mathbf{I}_s;\theta) - \nabla_\theta \log Z(\mathbf{I};\theta) \qquad (2.41)\\ &= \nabla_\theta \mathscr{E}(\mathbf{X} \mid \mathbf{I}_s;\theta) - \frac{\sum_{\mathbf{x}\in\mathscr{X}} \nabla_\theta\left(\mathscr{E}(\mathbf{x} \mid \mathbf{I}_s;\theta)\right)\exp\left(-\mathscr{E}(\mathbf{x} \mid \mathbf{I}_s;\theta)\right)}{Z(\mathbf{I}_s;\theta)}\\ &= \nabla_\theta \mathscr{E}(\mathbf{x}_s \mid \mathbf{I}_s;\theta) - \sum_{\mathbf{x}\in\mathscr{X}} P_\theta(\mathbf{X} = \mathbf{x}|\mathbf{I}_s;\theta)\nabla_\theta\left(\mathscr{E}(\mathbf{x} \mid \mathbf{I}_s;\theta)\right)\\ &= \nabla_\theta \mathscr{E}(\mathbf{x}_s \mid \mathbf{I}_s;\theta) - E_{\mathbf{X}\sim P_\theta}\left[\nabla_\theta \mathscr{E}(\mathbf{X} \mid \mathbf{I}_s;\theta)\right]\ . \qquad (2.42)\end{aligned}$$

The formula obtained in Equation 2.42 is central to likelihood-based structured learning methods. However, this equation hides a major technical difficulty. In order to compute the right-most term of Equation 2.42, we need to estimate $E_{\mathbf{X}\sim P_\theta}\left[\nabla_\theta \mathscr{E}(\mathbf{X} \mid \mathbf{I};\theta)\right]$, which is implicitly an inference problem.

From here, several approaches can be used. One alternative is to try to approximate explicitly $E_{\mathbf{X}\sim P_\theta}\left[\nabla_\theta \mathscr{E}(\mathbf{X} \mid \mathbf{I};\theta)\right]$ by standard inference techniques. Both Belief Propagation (BP) and Variational Inference (VI) methods described in Section 2.3, can be used here for approximate inference. This approach can be slow because of the computational complexity of inference algorithms and the quality of the learning may be limited by the approximation capacity of inference algorithms.

These approaches, and especially the one based on Mean-Field (MF) variational inference, will be discussed in more details in this thesis.

Another approach, which is used in practice to accelerate these algorithms, is based on an approximation of the expectancy of 2.42 via sampling. Ideally, if we had access to a method to sample exactly from the current estimate of the distribution P_θ, we could use it to form an unbiased estimator of $E_{\mathbf{X}\sim P_\theta}$. Using convergence properties of stochastic gradient descent, we could then, at each step, draw a point from the training set $(\mathbf{x}_s, \mathbf{I}_s)$ to compute the left-most term, and a sample from our unbiased P_θ sampler to compute a step.

In order to approximate a perfect sampler for P_θ, we can use Markov chain Monte Carlo (MCMC) or Gibbs sampling methods on CRF Walsh [2004]. However, in theory, one needs to run many MCMC iterations before convergence to obtain good and diverse samples. This has to be done again at each stochastic gradient iteration. It can make sampling based methods very slow in practice.

An idea, called contrastive divergence algorithm, was proposed by Hinton [2002] to solve this problem and accelerate sampling, at the cost of more noisy gradient estimates. The main idea is to initialize the MCMC with the sampled ground truth data-point $\mathbf{x}_s$ and run only a few MCMC iterations from here. The underlying assumption is that the samples from the ground-truth empirical distributions should not be too far from P_θ, and therefore, initializing the iterations with it is better than a random initialization. Again, we will see how this concept relates to our Multi-Modal Mean-Field algorithm.

2.4.2 Back Mean-Field

Recently, other authors developed a more pragmatic approach to the CRF parameters learning problem. Starting from the observation that the CRF is trained in order to be able to then make predictions using inference methods, the recent work of Domke [2013] proposed to directly learn the weights in order to make the variational inference process generate distributions which correctly model the ground truth. In other terms, it means that we are looking for

$$\theta^* = \underset{\theta}{\operatorname{argmin}} - \log Q_\theta(\mathbf{x}_s) \tag{2.43}$$

$$\text{s.t } Q_\theta = \underset{Q \in \mathcal{Q}}{\operatorname{argmin}} \, \mathrm{KL}(Q \| P_\theta)\,. \tag{2.44}$$

In order to optimize the parameters θ, the authors differentiate the mean-field iterations that are used to find Q_θ from Equation 2.44 using chain rule.

They then use this differentiable mapping to compute

$$\frac{\partial \log Q_\theta(x_s)}{\partial \theta},$$

which can finally be used to optimize Equation 2.43 with a gradient descent scheme.

In practice, Domke [2013], uses this method where Q_θ is obtained via naive Mean-Field inference. As we will see, this has severe limitations in terms of structured learning properties, because of the very limited modeling power of the naive MF approach.

However, it comes with the advantage that the practitioner can fine tune the parameters of a predefined CRF, via simple back-propagation through the MF iterations.

2.5 Deep CRFs and Computer Vision

Conditional Random Fields are a very useful modeling tool for structured learning problems. On the other hand, Deep Convolutional Neural Networks have proven their unmatched efficacy for feature extraction, univariate classification and regression tasks in Computer Vision. Very naturally, attempts have been made at combining both approaches in recent years. We will describe the type of models which is most often used and some of the relevant works in the domain.

2.5.1 Deep CRFs

In the previous sections, we described the CRFs as an hyper-graph where variables are connected by potentials. In Section 2.2, we also studied the exponential family CRFs, where the potentials $\phi_c(\mathbf{X}_c \mid \theta)$ can be decomposed as

$$\phi_c(\mathbf{X}_c \mid \theta) = \theta_c \phi_c(\mathbf{X}_c)\,,$$

where ϕ_c is a polynomial function of the clique variables $\mathbf{X}_c$, usually called sufficient statistics.

Here, we go one step further and assume that the potential functions depend parametrically on an image input $\mathbf{I}$, and hence rewrite the energy terms of Equation 2.10 as

$$\phi_c(\mathbf{X}_c \mid \mathbf{I}, \theta) = \theta_c(\mathbf{I}, \omega)\phi_c(\mathbf{X}_c)\,,$$

where θ_c is a Neural Network function and ω its synaptic weights.

In this setting, we replace the objective of learning the CRF parameters θ_c by the one of learning a parametric mapping from an image $\mathbf{I}$ through CNN parameters ω. Most of the times, the synaptic weights ω, will be shared between several neural networks, which will actually be a single one with many heads.

2.5.2 Not end-to-end CRFs as refinement

In many practical examples of structured prediction, a neural network is trained using a standard independent loss as in 2.1.1. However the obtained predictions, which don't take inter-variable correlations into account, are not suitable answers to the prediction problems. A predefined pairwise or higher order CRF inference module can then be used to recover a structured output. We provide below three examples of such cases.

Dense CRF for semantic segmentation This CRF model, introduced by Krähenbühl and Koltun [2011], has been used in many semantic segmentation pipelines and successfully used in conjunction with Deep-Learning based methods Chen et al. [2015]. We will build on top of it in several chapters of this thesis.

In this model, a CNN is trained to predict a marginal distribution over semantic labels independently for each pixel. A *dense* network of pairwise CRF potentials is then considered on top to refine the segmentation, based on pixels' proximity on the image and RGB similarity. More precisely, the pairwise potentials can be written as

$$\phi_{i,j}(x_i, x_j \mid \mathbf{I}; \theta) = \sum_{(k,l)} x_{i,k} x_{j,l} \exp\left(-\frac{\|p_i - p_j\|^2}{2\theta_\alpha^2}\right) \exp\left(-\frac{\|I_i - I_j\|^2}{2\theta_\beta^2}\right), \tag{2.45}$$

where p_i denotes the physical coordinates of the pixel on the image and I_i the rgb color on the original image.

Higher-Order repulsive CRF for detection Conditional random-field techniques have also been used in the context of object detection, even though not always presented as such. The most typical example is the Hough transform framework. The Hough method can be interpreted as a greedy heuristic to find a Maximum-a-Posteriori in a CRF. In that case, the energy has the same form as the objective of a facility location problem [Farahani et al., 2012].

In this class of problems, large clique sizes are involved. The potential terms in the energy function are designed to take a large value if *at least one* of the variables in a clique has label of interest and 0 otherwise. These potentials are useful for detection tasks, where we observe a detection evidence on the image, for instance a non-background pixel, and we want at least one of the detection variables to be *on* to explain this detection. More precisely, the potential functions will be written as

$$\phi_c(\mathbf{x}_c \mid \mathbf{I};\theta) = \prod_{i \in c}(1 - x_i)\theta(\mathbf{I}); \tag{2.46}$$

where $\theta(\mathbf{I})$ is a detection feature function in the image. It can be a background subtraction operation, as in Fleuret et al. [2008], or be based on a more sophisticated classifier, as in Barinova et al. [2012].

Barinova et al. [2012] propose a powerful Heuristic – inspired by standard Operations Research techniques –, in order to solve for a MAP assignment in a repulsive MRF. They apply their method in the context of pedestrian detection with Hough Forests. As the non-maximum-suppression task is fairly simple in the examples they use, their method performs well [Barinova et al., 2012].

In Fleuret et al. [2008], a similar problem is solved by using a mean-field relaxation, which is solved with fixed point iterations.

Other works Very recent approaches to bounding-box detection in images, such as Ren et al. [2015], Liu et al. [2016], use a Non-Maximum-Suppression (NMS) post-processing step to produce final detections. This is needed because the unary classifier produces independent detections which don't take into account the fact that an object has already been detected. The NMS step can then be seen as a MAP inference process in a pairwise repulsive CRF where variables correspond to bounding-boxes and potentials are based on the Intersection-over-Union between bounding-boxes.

Recent approaches to multiple people pose estimation such as the one of Pishchulin et al. [2016], are also using a pairwise CRF to reconstruct skeletons, where unary potentials are based on joint detectors and pairwise ones are based on a learned estimation of compatibility between pairs of joints.

2.5.3 End-to-end trained models

In all the examples described above, the unary and pairwise potentials can sometimes use data-driven models. For instance, unary potentials can stem from trained CNN-based pixel classifiers Chen et al. [2015] or pairwise potentials may be computed from a learned regressor as in Pishchulin et al. [2016]

However, some parts of the parameters CRF parameters always need to be set manually and the model is never trained directly to produce the expected outputs. It therefore often requires a careful manual selection of some parameters, which will for instance weight the importance of the different types of potentials with respect to each other.

Training the model end-to-end requires to apply one of the methods which were described in section 2.4. However, modern Neural Network architectures are very efficiently implemented and are usually orders of magnitude faster than the slow and iterative inference processes described above. Therefore, the practical complexity of inference methods has slowed down their adoption in modern computer vision pipeline which are hardly ever trained end-to-end.

Nevertheless, the work of Zheng et al. [2015] partially solved the problem and proposed an approach based on the back mean-field method described in section 2.4. The authors unroll the naive mean-field inference iterations as a sequence of neural-network layers. They can then back-propagate the gradient over the iterations and learn the parameters of the CRF which have implicitly become Neural Network parameters in this new inference layer.

One of the strengths of this new model was to introduce the concept of adaptive filtering in Neural Networks. The convolutions were not any more performed according to the image distance, but on a higher 5-dimensional space including xy-coordinates and rgb colors in the initial image.

However, as we will see in this thesis, this method remains restricted by the modeling power of the naive mean-field. It fails to learn parameters properly in very ambiguous and multi-modal settings.

3 Principled Parallel Mean-Field Inference for Discrete Random Fields

3.1 Optimizing Mean-Field

As explained in Chapter 2, many Computer Vision problems, ranging from image segmentation to depth estimation from stereo, can be naturally formulated in terms of Conditional Random Fields (CRFs). Solving these problems then requires either estimating the most probable state of the CRF, or the marginal distributions over the unobserved variables. Since there are many such variables, it is usually impossible to get an exact answer, and one must instead look for an approximation.

Mean-field variational inference Wainwright and Jordan [2008] is one of the most effective ways to do approximate inference and has become increasingly popular in our field Saito et al. [2012], Vineet et al. [2014], Krähenbühl and Koltun [2013]. It involves introducing a variational distribution that is a product of terms, typically one per hidden variable. These terms are then estimated by minimizing the Kullback-Leibler (KL) divergence between the variational and the true posterior. The standard scheme is to iteratively update each factor of the distribution one-by-one. This is guaranteed to converge Bishop [2006], Koller and Friedman [2009], but is not very scalable, because all variables have to be updated sequentially. It becomes impractical for realistically-sized problems when there are substantial interactions between the variables. This can be remedied by replacing the sequential updates by parallel ones, often at the cost of failing to converge.

It has nonetheless recently been shown that parallel updates could be done in a provably

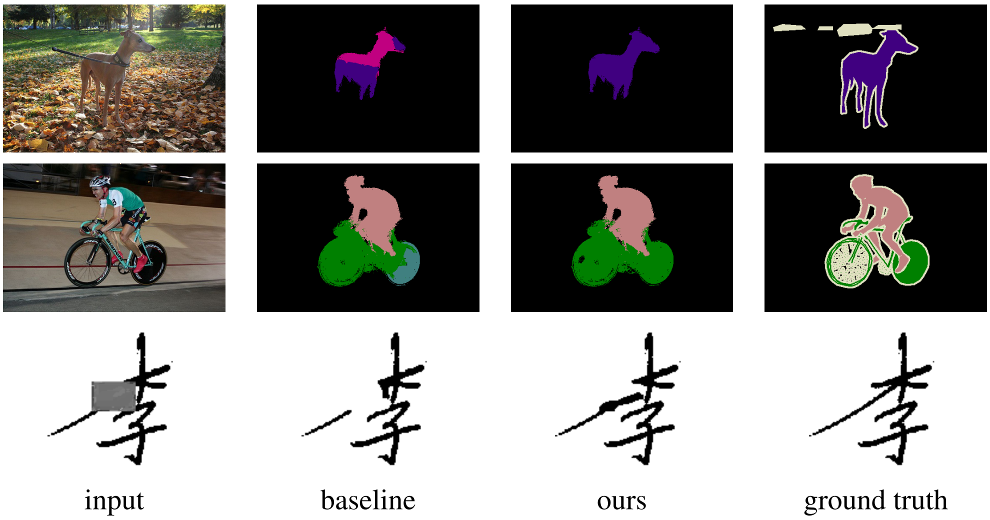


Figure 3.1 – **First two rows:** VOC2012 images in which we outperform a baseline by adding simple co-ocurrence terms, which our optimization scheme, unlike earlier ones, can handle. **Bottom row:** Our scheme also allows us to improve upon a baseline for the purpose of recovering a character from its corrupted version.

convergent way for pairwise CRFs, provided that the potentials are concave Krähenbühl and Koltun [2013]. When they are not, an *ad hoc* heuristic designed to achieve convergence, which essentially smooths steps by averaging between the next and current iterate, has been used over the years. This heuristic is mentioned explicitly in some works Sun et al. [2013], Frostig et al. [2014], or used implicitly in optimization schemes Fleuret et al. [2008], Vineet et al. [2014] by introducing an additional damping parameter.

However, a formal justification for such smoothing is never provided, which we do in this chapter. More specifically, we show that, by damping in the natural parameter space instead of the mean-parameter one, we can reformulate the optimization scheme as a specific form of proximal gradient descent. This yields a theoretically sound and practical way to chose the damping parameters, which guarantees convergence, no matter the shape of the potentials. When they are attractive, we show that our approach is equivalent to that of Krähenbühl and Koltun [2013]. However, even when they are repulsive and can cause the earlier methods to oscillate without ever converging, our scheme still delivers convergence. For example, as shown in Figure 3.1, this allows us to add co-occurrence terms to the model used by a state-of-the-art semantic segmentation method Chen et al. [2015] and improves its results. Furthermore, we retain the simplicity of the closed-form mean-field update rule, which is one of the key strengths of the mean-field approach.

In short, our contribution is threefold:

- We introduce a principled, simple, and efficient approach to performing parallel inference in discrete random fields. We formally prove that it converges and demonstrate that it performs better than state-of-the-art inference methods on realistic Computer Vision tasks such as segmentation and people detection.
- We show that many of the earlier methods can be interpreted as variants of ours. However, we offer a principled way to set its metaparameters.
- We demonstrate how parallel mean-field inference in random fields relates to the gradient descent. This allows us to integrate advanced gradient descent techniques, such as momentum and ADAM Kingma and Ba [2014], which makes mean-field inference even more powerful.

To validate our approach, we first evaluate its performance on a set of standardized benchmarks, which include a range of inference problems and have recently been used to assess inference methods Frostig et al. [2014]. We then demonstrate that the performance improvements we observed carry over to three realistic Compute Vision problems, namely Characters Inpainting, People Detection and Semantic Segmentation. In each case, we show that modifying the optimization scheme while retaining the objective function of state-of-the-art models Fleuret et al. [2008], Nowozin et al. [2011], Chen et al. [2015] yields improved performance and addresses the convergence issues that sometimes arise Vineet et al. [2014].

3.2 Related Work

In this section, we briefly review basic Conditional Random Field (CRF) theory detailed in section 2.2 and the use of mean-field inference to solve the resulting optimization problems. We also give a short introduction into proximal gradient descent algorithms, on which our method is based. Note, in this work, we focus on models involving discrete random variables.

3.2.1 Conditional Random Fields

Let $\mathbf{X} = (X_1, \ldots, X_N)$ represent hidden variables and $\boldsymbol{I}$ represent observed variables. For example, for semantic segmentation, the X_is are taken to be variables representing semantic classes of N pixels, and $\boldsymbol{I}$ represents the observed image evidence.

A Conditional Random Field (CRF) models the relationship between $\mathbf{X}$ and $\mathbf{I}$ in terms of the posterior distribution

$$P(\mathbf{X} \mid \mathbf{I}) = \exp\left(\sum_{c \subset \{1,\ldots,N\}} \boldsymbol{\phi}_c(\mathbf{X}_c \mid \mathbf{I}) - \log Z(\mathbf{I}) \right), \tag{3.1}$$

where $\boldsymbol{\phi}_c(.)$ are non-negative functions known as potentials and $\log Z(\mathbf{I})$ is the log-partition function. It is a constant that we will omit for simplicity since we are mostly concerned by estimating values of $\mathbf{X}$ that maximize $P(\mathbf{X} \mid \mathbf{I})$.

This model is often further simplified by only considering unary and pairwise terms:

$$P(\mathbf{X} \mid \mathbf{I}) \propto \exp\left(\sum_i \phi_i(X_i, I_i) + \sum_{(i,j)} \phi_{ij}(X_i, X_j) \right). \tag{3.2}$$

3.2.2 Mean-Field Inference

Typically, one wants either to estimate the posterior $P(\mathbf{X}|\mathbf{I})$ or to find the vector $\hat{\mathbf{X}}$ that maximizes $P(\mathbf{X}|\mathbf{I})$, which is known as the MAP assignment. Unfortunately, even for the simplified formulation of Equation 3.2, both are intractable for realistic sizes of $\mathbf{X}$. As a result, many approaches settle for approximate solutions. These include sampling methods, such as Gibbs sampling Gelfand and Smith [1990], and deterministic ones such as mean-field variational inference Winn and Bishop [2005], belief propagation Murphy et al. [1999], Minka [2001], Kolmogorov [2015], and others Boykov et al. [2001], Gorelick et al. [2014]. A comprehensive comparison of inference methods in discrete models is provided in Kappes et al. [2015].

Note that, mean-field methods have been shown to combine the advantages of good convergence guarantees Bishop [2006], flexibility with respect to the potential functions that can be handled Saito et al. [2012], and potential for parallelization Krähenbühl and

Koltun [2013]. As a result, they have become very popular in our field. Furthermore, they have recently been shown to yield state-of-the-art performance for several Computer Vision tasks Saito et al. [2012], Vineet et al. [2014], Chen et al. [2015], Zheng et al. [2015].

Mean-field involves introducing a distribution Q of the factorized form

$$Q(\mathbf{X} = (x_1, \ldots, x_N); \mathbf{q}) = \prod_{i=1}^{N} Q_i(X_i = x_i; \mathbf{q}_i)\,, \tag{3.3}$$

where $Q_i(.; \mathbf{q}_i)$ is a categorical distribution with mean parameters $\mathbf{q}_i$. That is,

$$\forall l,\, Q_i(X_i = l; \mathbf{q}_i) = q_{i,l}, \tag{3.4}$$

with $\mathbf{q}$ in the space $\mathcal{M}$ such that $\forall i \in \{1, \ldots, N\}, l \in \{1, \ldots, L\}$, $0 \leq q_{i,l} \leq 1$ and $\forall i, \sum_l q_{i,l} = 1$, where N is often the number of pixels, and L is the number of labels.

Q is then used to approximate $P(\mathbf{X} \mid \mathbf{I})$ by minimizing the KL-divergence:

$$\mathrm{KL}(Q||P) = \sum_{\boldsymbol{x}} Q(\mathbf{X} = \mathbf{x}; \mathbf{q}) \log \frac{Q(\mathbf{X} = \mathbf{x}; \mathbf{q})}{P(\mathbf{X} = \mathbf{x} \mid \mathbf{I})}\,. \tag{3.5}$$

In some cases, this approximation is the desired final result. In others, one seeks a MAP assignment. To this end, a standard method is to select the assignment that maximizes the *approximate* posterior $Q(\mathbf{X}; \mathbf{q})$, which is equivalent to rounding when the X_is are Bernoulli variables. An alternative approach is to draw samples from $Q(\mathbf{X}; \mathbf{q})$.

When minimizing the KL-divergence of Equation 3.5, $Q(\mathbf{X}; \mathbf{q})$ can be reparameterized in terms of its *natural* parameters defined as follows. For each variable X_i and label l, we take the natural parameter $\theta_{i,l}$ to be such that

$$Q(X_i = l; \mathbf{q}_i) = q_{i,l} \propto \exp[-\theta_{i,l}]. \tag{3.6}$$

As we will see below, this parameterization often yields simpler notations and implementations.

Sweep Mean-Field Inference

As seen in section 2.3 the expression of Equation 3.5 is equivalent Bishop [2006] to minimizing

$$\mathscr{F}(\mathbf{q}) = \underbrace{-\mathbf{E}_{Q(X;\mathbf{q})}[\log P(\mathbf{X}\,|\,\mathbf{I})]}_{\mathscr{E}(\mathbf{q})} + \underbrace{\mathbf{E}_{Q(X;\mathbf{q})}[\log Q(X;\mathbf{q})]}_{-\mathscr{H}(\mathbf{q})}, \tag{3.7}$$

with respect to $\mathbf{q} \in \mathscr{M}$. $\mathscr{F}(.)$ is sometimes called the variational free energy. Its first term is the expectation of the energy under $Q(\mathbf{X};\mathbf{q})$, and its second term is the negative entropy, which acts as a regularizer.

One can minimize $\mathscr{F}(\mathbf{q})$ by iteratively updating each $q_{i,l}$ in sequence while keeping the others fixed Bishop [2006]. Each update involves setting $q_{i,l}$ to

$$q^{\star}_{i,l} \propto \exp\left[\mathbf{E}_{Q(\mathbf{X}/X_i;\mathbf{q})}\left[\log P(\mathbf{X}\,|\,\mathbf{I})\right]\right]. \tag{3.8}$$

This coordinate descent procedure, which we will call `SWEEP`, is guaranteed to converge to a local minimum of $\mathscr{F}$ Bishop [2006]. However, it tends to be very slow for realistic image sizes and impractical for many Computer Vision problems Vineet et al. [2014], Krähenbühl and Koltun [2013]. Namely, in the case of dense random fields, it involves re-computing a large number of expectations (one per factor adjacent to the variable) after each sequential update. Filter-based mean-field inference Krähenbühl and Koltun [2011] attempts to reduce the complexity of these updates, but it effectively performs parallel updates, which we will describe below.

Parallel Mean-Field Inference

To obtain reasonable efficiency in practice, Computer Vision practitioners often perform the updates of Equation 3.8 in parallel as opposed to sequentially. Not only does it avoid having to reevaluate a large number of factors after each update, it also allows the use of vectorized instructions and GPUs, both of which can have a dramatic impact on the computation speed.

Unfortunately, these parallel updates invalidate the convergence guarantees and in practice often lead to undesirable oscillations in the objective. Several approaches to remedying this problem have been proposed, which we review below.

Damping A natural way to improve convergence is to replace the updates of Equation 3.8 by a damped version, expressed as

$$q_{i,l}^{t+1} = (1-\eta)\cdot q_{i,l}^{t} + \eta \cdot q_{i,l}^{\star}, \tag{3.9}$$

where t denotes the current iteration, $q_{i,l}^{\star}$ is the result of solving the optimization problem of Equation 3.8, and η is a heuristically chosen damping parameter. This damping is explicitly mentioned in papers such as the ones of Sun et al. [2013], Frostig et al. [2014]. In Vineet et al. [2014], convergence issues are mentioned and a damping parameter is provided in the publicly available code. Similarly, in Fleuret et al. [2008], the algorithm relies on mean-field optimization with repulsive terms. The need for damping is not explicitly discussed in the paper, but the publicly available code also includes a damping.

Damping delivers satisfactory results in many cases, but does not formally guarantee convergence. It may fail if the parameter η is not carefully chosen, and sometimes changed at different stages of the optimization. In all the approaches that we are aware of, this is done heuristically. We will refer to this type of methods as `ADHOC`.

Concave potentials A principled way to address the convergence issue for the pairwise random fields is offered in Krähenbühl and Koltun [2013], and we refer to the corresponding algorithm as `FULL-PARALLEL`. However, authors restrict their potentials ϕ_{ij} of Equation 3.2 to be concave, which in some cases is reasonable, but as we will show in Section 3.4, many Computer Vision models violate this requirement. By contrast, our approach is similarly principled but without additional constraints. In practice it works for higher-order, or, equivalently, non-pairwise potentials.

3.2.3 Proximal Gradient Descent

Let F be a generic objective function of the form $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$, where g is a regularizer, and $\mathbf{x}_t$ is the value of the optimized variable at iteration t of a minimization procedure on a constraint set $\mathscr{X}$. Proximal gradient descent, also known as composite mirror-descent Duchi et al. [2010], is an iterative method that relies on the update rule

$$\mathbf{x}^{t+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathscr{X}} \{\langle \mathbf{x}, \nabla f(\mathbf{x}^t)\rangle + g(\mathbf{x}) + \lambda \Psi(\mathbf{x}, \mathbf{x}^t)\} \ , \tag{3.10}$$

where Ψ is a non-negative *proximal* function that satisfies $\Psi(\mathbf{x}, \mathbf{x}^t) = 0$ if and only if $\mathbf{x} = \mathbf{x}^t$, and $\lambda > 0$ is a scalar parameter. g contains the terms of the objective function that do not need to be approximated to the first order, while still allowing efficient computation of update of Equation 3.10. Ψ can be understood as a distance function that accounts for the geometry of $\mathscr{X}$ Teboulle [1992] while also making it possible to compute the update of Equation 3.10 efficiently. λ can then be thought of as the inverse of the step size.

As shown in Section 3.3.1, our algorithm is a version of proximal gradient descent in which Ψ is based on the KL-divergence and allows automated step-size adaptation as the optimization progresses. Recently, a variational approach that also relies on the KL-divergence as the proximal function has been proposed Khan et al. [2015]. This thesis explores the connection between the KL-proximal method and the Stochastic Variational Inference Amari [1998], Hoffman et al. [2013]. However, the method presented there is not directly applicable to discrete random fields, especially for the Vision problems we consider. Moreover, it does not allow for step size adaptation, which often yields better performance, as we demonstrate in our experiments.

3.3 Method

As discussed in the previous section, the goal of mean-field inference is to

$$\operatorname*{minimize}_{\mathbf{q}\in\mathscr{M}} \mathscr{F}(\mathbf{q}) \tag{3.11}$$

where $\mathscr{F}$ is the variational free energy of Equation 3.7. Performing sequential updates of the $q_{i,l}$ is guaranteed to converge, but can be slow. Parallel updates are usually much faster, but the optimization procedure may fail to converge.

In this section, we introduce our approach to guaranteeing convergence whatever the shape of the pairwise potentials. To this end, we rely on proximal gradient descent as described in Section 3.3.1 and formulate the proximal function Ψ in terms of the KL-divergence. This is motivated by the fact that it is more adapted to measuring the distance between probability distributions than the usual L2 norm, while being independent of how the distribution is parameterized.

We will show that this both guarantees convergence and yields a principled way to obtain a closed form damped update equation equivalent to Equation 3.9.

3.3.1 Proximal Gradient for Mean-Field Inference

In our approach to minimizing the variational free energy of Equation 3.7, we treat $\mathscr{E}$ as the function f of Equation 3.10 and the negative entropy $-\mathscr{H}$ as the regularizer g. This choice stems from the fact that $-\mathscr{H}$ is separable, and therefore, can be minimized in parallel in Equation 3.10, without using a first order approximation. Also, $-\mathscr{H}$ being the regularizer g means that we do not need to look at its derivatives with respect to the mean-parameters, which are not well behaved when they approach zero. We then define

$$\Psi^t(\mathbf{q}, \mathbf{q}^t) = \sum_i \sum_l d^t_{i,l} q_{i,l} \log \frac{q_{i,l}}{q^t_{i,l}} = \mathbf{D}^t \odot \mathrm{KL}(\mathbf{q} || \mathbf{q}^t) \,, \tag{3.12}$$

where KL is the non-negative KL-divergence, which is a natural choice for a distance between distributions. $\mathbf{D}^t$ is a diagonal matrix with positive diagonal elements $d^t_{i,l}$s, which we introduce to allow for anisotropic scaling of the proximal KL-divergence term. As will be discussed below, different choices of the $d^t_{i,l}$s yield different variants of our algorithms. Note however that, Ψ^t is a valid proximal function.

The update of Equation 3.10 then becomes

$$\mathbf{q}^{t+1} = \operatorname*{argmin}_{\mathbf{q} \in \mathscr{M}} \{ \langle \mathbf{q}, \nabla \mathscr{E}(\mathbf{q}^t) \rangle - \mathscr{H}(\mathbf{q}) + \mathbf{D}^t \odot \mathrm{KL}(\mathbf{q} || \mathbf{q}^t) \} \,. \tag{3.13}$$

This computation can be performed independently for each index $i \in \{1, \ldots, N\}$. Furthermore, as we prove in section 3.3.2, it can be done in closed form and can be written as

$$\begin{aligned} q_{i,l}^{t+1} \quad \propto \exp[\quad & \eta_{i,l}^{t} \cdot \mathbf{E}_{Q(\mathbf{X}/X_i=l;\mathbf{q})}\left[\log P(\mathbf{X}|\mathbf{I})\right] \\ & + (1-\eta_{i,l}^{t}) \cdot \log q_{i,l}^{t}] \,, \end{aligned} \tag{3.14}$$

where $\eta_{i,l}^{t} = \dfrac{1}{1+d_{i,l}^{t}}$. Eq 3.14 can be rewritten as

$$\theta_{i,l}^{t+1} = \eta_{i,l}^{t} \cdot \theta_{i,l}^{\star} + (1-\eta_{i,l}^{t}) \cdot \theta_{i,l}^{t} \ , \tag{3.15}$$

where $\theta_{i,l}^{\star} = -\mathbf{E}_{Q(\mathbf{X}/X_i;\mathbf{q})}\left[\log P(\mathbf{X}|\mathbf{I})\right]$ now is a natural parameter, like those of Equation 3.6. In other words, we have replaced the heuristic update rule of Equation 3.9 in the space of mean parameters by a principled one in the space of natural ones. As we will see, this yields performance and convergence improvements in most cases. As for the stopping criteria, one can define one based on the value of the objective, or, in practice, run inference for a fixed number of iterations.

3.3.2 Derivation of the closed form update

We will now derive the closed-form update rule for the KL-proximal gradient descent introduced in the previous section.

Let us now consider the proximal gradient update,

$$\underset{q \in \mathcal{M}}{\text{minimize}} \left\{ \langle \mathbf{q}, \nabla \mathcal{E}(\mathbf{q}^t) \rangle - \mathcal{H}(\mathbf{q}) + \mathbf{D}^t \odot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t) \right\} \ , \tag{3.16}$$

where the first and the second terms are the expected energy and negative entropy respectively, and the last term is the proximal term. It can be written as

$$\mathbf{D}^t \odot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t) = \sum_{i,l} d_{i,l} \cdot q_{i,l} \log \frac{q_{i,l}}{q_{i,l}^{t}} \ , \tag{3.17}$$

where $\mathbf{D}^t$ is a diagonal matrix with non-zero elements $d_{i,l}$.

Our goal is to derive a closed-form update for all the mean parameters $q_{i,l}$, or, alternatively, for all the natural parameters $\theta_{i,l}$. We can then write down the partial derivative of the expected energy with respect to any $q_{i,l}$ as

$$\nabla\mathcal{E}(\mathbf{q}^t)_{i,l} = \frac{\partial\mathcal{E}(\mathbf{q}^t)}{\partial q_{i,l}} = \mathbf{E}_{Q(\mathbf{X}|\mathbf{q}^t_{\setminus i})}[\log p(\mathbf{X}|\mathbf{I})|X_i = l] \,. \tag{3.18}$$

Note, that both our objective $\mathcal{F}$ *and* the constraints $\mathbf{q} \in \mathcal{M}$ are separable over the variables $X_1, \ldots, X_N$, which makes it possible to minimize independently for each X_i. In other words, our goal is to solve for all i

$$\underset{\mathbf{q}_i}{\text{minimize}} \qquad \sum_l q_{i,l}\nabla\mathcal{E}(\mathbf{q}^t)_{i,l} + \sum_l q_{i,l}\log q_{i,l} + d_i^t \sum_l q_{i,l}\log\frac{q_{i,l}}{q^t_{i,l}} \,, \tag{3.19}$$

$$\text{subject to} \qquad \sum_l q_{i,l} = 1 \tag{3.20}$$

Similarly to the sweep updates described previously, we convert each problem to an unconstrained one by introducing the Lagrangian

$$\begin{aligned}\mathcal{L}(\mathbf{q}_i, \mu_i) = &\sum_l q_{i,l}\nabla\mathcal{E}(\mathbf{q}^t)_{i,l} + \sum_l q_{i,l}\log q_{i,l} \,, \\ &+ d_i^t \sum_l q_{i,l}\log\frac{q_{i,l}}{q^t_{i,l}} - \mu_i\left(\sum_l q_{i,l} - 1\right) \,,\end{aligned} \tag{3.21}$$

where μ_i is a corresponding Lagrange multiplier.

We then differentiate it with respect to $q_{i,l}$, $\forall i, l$

$$(1 + d_i^t)\log q^\star_{i,l} = \mathbf{E}_{Q(X|\mathbf{q}_{-i})}[\log p(\mathbf{X}|\mathbf{I})|X_i = l] + d_i^t \log q^t_{i,l} + \mu_i \,, \tag{3.22}$$

which in turn leads to the update rule

$$q^{t+1}_{i,l} \propto \exp\left[\eta_i^t \cdot \mathbf{E}_{Q(X|\mathbf{q}_{-i})}[\log p(\mathbf{X}|\mathbf{I})|X_i = l] + (1 - \eta_i^t)\cdot \log q^t_{i,l}\right] \,, \tag{3.23}$$

where $\eta_i^t = \frac{1}{1+d_i^t}$, and normalization constant can be obtained from μ_i.

3.3.3 Fixed Step Size

The simplest way to instantiate our algorithm is to fix all the $d_{i,l}^t$s of Equation 3.12 to the same value d and to write

$$\forall t\,,\, \mathbf{D}^t = \mathbf{D} = d\mathbb{1} \Rightarrow \forall t, i, l, \eta_{i,l}^t = \frac{1}{1+d}\,, \tag{3.24}$$

where $\eta_{i,l}^t$ plays the same role as the damping factor of Equation 3.9. We now show that this is guaranteed to converge when the proximal term is given enough weight.

In our mean-field settings, $\mathscr{E}(\mathbf{q})$ is a polynomial function of the mean-parameters vector $\mathbf{q}$. Therefore, one can always find some positive real number L such that the gradient of $\mathscr{E}$ is *L-Lipschitz continuous*. We prove below that this property implies that our proximal gradient descent scheme is guaranteed to converge for any fixed matrix $D = d\mathbb{1}$ such that $d > L$.

Intuitively, when updating the value of $\mathbf{q}^t$ to $\mathbf{q}^{t+1}$, the magnitude of the gradient change stays controlled and thus the coordinate-wise optimum $\theta_{i,l}^\star = -\nabla\mathscr{E}(\mathbf{q}^t)_{i,l}$ will also be changing smoothly across iterations. As a result, L is the key value to understand oscillations. In practice, our goal is to find its smallest possible value to allow steps as large as possible while guaranteeing convergence.

Lemma 3 *The gradient of the proximal term at the current iteration point* $\nabla_{\mathbf{q}}\mathbf{D}^t \odot KL(\mathbf{q}||\mathbf{q}^t)|_{\mathbf{q}=\mathbf{q}^t}$ *is orthogonal to* $\mathscr{M}$.

Proof Let's write down the gradient:

$$\nabla_{\mathbf{q}}\mathbf{D}^t \odot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t) = (d_1^t \cdot \nabla_{\mathbf{q}_1}\mathrm{KL}(\mathbf{q}_1||\mathbf{q}_1^t), \ldots, d_N^t \nabla_{\mathbf{q}_N}\mathrm{KL}(\mathbf{q}_N||\mathbf{q}_N^t))\,, \tag{3.25}$$

with each component containing:

$$\nabla_{\mathbf{q}_i}\mathrm{KL}(\mathbf{q}_i||\mathbf{q}_i^t) = (\log\frac{q_{i,1}}{q_{i,1}^t} + 1, \ldots, \log\frac{q_{i,M}}{q_{i,M}^t} + 1)\,. \tag{3.26}$$

The partial gradient at the current iteration point $\mathbf{q}_i^t$ is the all-ones vector:

$$\nabla_{\mathbf{q}_i} \mathrm{KL}(\mathbf{q}_i || \mathbf{q}_i^t)|_{\mathbf{q}_i = \mathbf{q}_i^t} = (1, \ldots, 1)\,, \tag{3.27}$$

which is obviously orthogonal to the hyperplane defined by the constraint $\sum_l q_{i,l} = 1$. Thus, $d_i^t \nabla_{\mathbf{q}_i} \mathrm{KL}(\mathbf{q}_i || \mathbf{q}_i^t)|_{\mathbf{q}_i = \mathbf{q}_i^t}$ is also orthogonal to this hyperplane, and we easily obtain the orthogonality of the product vector $\nabla_{\mathbf{q}} \mathbf{D}^t \odot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t)|_{\mathbf{q}=\mathbf{q}^t}$ to $\mathcal{M}$.

Lemma 4 *For all* $\mathbf{q}^t$ *in* $\mathcal{M}$*,*

$$\forall \mathbf{q} \in \mathcal{M},\, \mathbf{D}^t \cdot KL(\mathbf{q}^{t+1} || \mathbf{q}^t) \geq \frac{L}{2} \|\mathbf{q} - \mathbf{q}^t\|_2^2\,.$$

Proof Note that the Hessian of the KL-proximal term is diagonal with

$$\forall \mathbf{q} \in \mathcal{M},\, \frac{\partial^2 \mathbf{D}^t \cdot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t)}{\partial q_{i,l}^2}|_{\mathbf{q}} = \frac{d_{i,l}^t}{q_{i,l}} \geq L\,. \tag{3.28}$$

Therefore, the proximal term is L-strongly convex on $\mathcal{M}$. For all $\mathbf{q}^t$ in $\mathcal{M}$,

$$\forall \mathbf{q} \in \mathcal{M},\, \mathbf{D}^t \cdot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t) \geq \langle \nabla_{\mathbf{q}} \mathbf{D}^t \odot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t)|_{\mathbf{q}=\mathbf{q}^t}, \mathbf{q} - \mathbf{q}^t \rangle + \frac{L}{2} \|\mathbf{q} - \mathbf{q}^t\|_2^2\,. \tag{3.29}$$

The first term of the right hand side is null according to the orthogonality property 3. Which leads to

$$\forall \mathbf{q} \in \mathcal{M},\; \mathbf{D}^t \cdot \mathrm{KL}(\mathbf{q}^{t+1} || \mathbf{q}^t) \geq \frac{L}{2} \|\mathbf{q} - \mathbf{q}^t\|_2^2\,. \tag{3.30}$$

We will now demonstrate, that under certain assumptions, applying updates of Eq. 3.23 lead to a decrease in objective at each iteration.

Theorem 5 *If* $\mathcal{E}$ *is L-Lipschitz gradient on* $\mathcal{M}$*, and that* d_i^t*s are chosen such that* $d_i^t \geq L$*,* $\forall\, t, i$*. Then the objective function is decreasing at each step.*

Proof Let us assume that $\mathcal{E}$ is L-Lipschitz gradient on $\mathcal{M}$ and that $d_i^t \geq L, \forall t, i$. Then, we can show that the value of the objective function $\mathcal{E}(\mathbf{q}^{t+1}) - \mathcal{H}(\mathbf{q}^{t+1})$ at step $t+1$ has to be smaller than $\mathcal{E}(\mathbf{q}^t) - \mathcal{H}(\mathbf{q}^t)$

$$\begin{aligned}
\mathcal{E}(\mathbf{q}^t) - \mathcal{H}(\mathbf{q}^t) &\geq \underset{\mathbf{q}}{\operatorname{argmin}} \left[\mathcal{E}(\mathbf{q}^t) + \langle(\mathbf{q} - \mathbf{q}^t), \nabla\mathcal{E}(\mathbf{q}^t)\rangle - \mathcal{H}(\mathbf{q}) + \mathbf{D}^t \cdot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t)\right] && (3.31) \\
&\geq \mathcal{E}(\mathbf{q}^t) + \langle(\mathbf{q}^{t+1} - \mathbf{q}^t), \nabla\mathcal{E}(\mathbf{q}^t)\rangle - \mathcal{H}(\mathbf{q}^{t+1}) + \mathbf{D}^t \cdot \mathrm{KL}(\mathbf{q}^{t+1}||\mathbf{q}^t) && (3.32) \\
&\geq \mathcal{E}(\mathbf{q}^t) + \langle(\mathbf{q}^{t+1} - \mathbf{q}^t), \nabla\mathcal{E}(\mathbf{q}^t)\rangle - \mathcal{H}(\mathbf{q}^{t+1}) + \frac{L}{2}\|\mathbf{q}^{t+1} - \mathbf{q}^t\|_2^2 && (3.33) \\
&\geq \mathcal{E}(\mathbf{q}^{t+1}) - \mathcal{H}(\mathbf{q}^{t+1}) && (3.34)
\end{aligned}$$

where step Equation 3.32 comes from the fact that by definition $\mathbf{q}^{t+1}$ realizes the minimum, Equation 3.33 holds by strong-convexity lower bound 4 and Equation 3.34 holds by L-Lipschitz gradient property of $\mathcal{E}$.

In the pairwise case, the Hessian of the objective function is a constant matrix, which we call potential matrix. Therefore, the highest eigenvalue of the potential matrix is a valid Lipschitz constant and efficient methods allow to compute it for moderately sized problems.

In fact, the convergence result presented in Krähenbühl and Koltun [2013] is strongly related to this. Namely, assuming that the potential matrix is negative semi-definite, is equivalent to assuming that $L < 0$ in our formulation. This directly corresponds to the concavity assumptions on the potentials in Krähenbühl and Koltun [2013]. Therefore, under the assumptions of Krähenbühl and Koltun [2013], our algorithm leads to $\eta = 1$, corresponding to the fully-parallel update procedure. In that sense, our procedure is a generalization of the one proposed by Krähenbühl and Koltun [2013].

In the non-pairwise case, the Hessian is not constant, and the calculation of the Lipschitz constant is not trivial. For each specific problem, bounds should be derived using the particular shape of the CRF at hand.

3.3.4 Adaptive Step Size

Note that the Hessian of the KL-proximal term is diagonal with

$$\frac{\partial^2 \mathbf{D}^t \cdot \mathrm{KL}(\mathbf{q}||\mathbf{q}^t)}{\partial q_{i,l}^2}|_{\mathbf{q}=\mathbf{q}^t} = \frac{d_{i,l}^t}{q_{i,l}^t} \,. \tag{3.35}$$

Therefore, when some of the $q_{i,l}$s get close to 0, the elements of the Hessian may become very large, especially when using a constant value for the $d_{i,l}^t$ as suggested above. When that happens, the local KL-approximation remains a valid upper bound of the objective function, but not a tight enough one, which results in step sizes that are too small for fast convergence.

This can be reduced by choosing a matrix $\mathbf{D}^t$ that compensates for this. A simple way to do this would be to scale the $d_{i,l}^t$ proportionally to $\max(q_{i,0}, \ldots, q_{i,L_i-1})$ to start compensating for diagonal terms. However, this method is still sub-optimal because it ignores the fact that all our variables lie inside the simplex $\mathscr{M}$. A better alternative is to bound from below the proximal term by a quadratic function, but on $\mathscr{M}$ rather than on $\mathbb{R}^n$.

In this chapter, we only apply this method to the binary case, for which we set

$$d_{i,0}^t = d_{i,1}^t = q_{i,0}^t q_{i,1}^t \cdot d \,, \tag{3.36}$$

were d is an additional parameter that should be set close to L. Extending this approach to the multi-label case will be a topic for future work. In Section 3.3.5, we provide a different alternative to performing adaptive anisotropic updates in all settings.

Intuitively, when the current parameters are close to the borders of the simplex, the mean parameters are less sensitive to natural parameters, which, therefore, need less damping. We demonstrate in our experiments that it provides a way to choose the step size without tuning.

3.3.5 Momentum

Our approach can easily be extended to incorporate techniques that are known to speed-up gradient descent and help to avoid local minima, such as the classic momentum method Polyak [1964] or the more recent ADAM technique Kingma and Ba [2014]. The momentum method involves averaging the gradients of the objective $f(\mathbf{x})$ over the iterations in a *momentum* vector $\mathbf{m}$ and use it as the direction for the update instead of simply following the current gradient. To integrate it into our framework, we replace the gradient $\nabla\mathcal{E}$ in Equation 3.13 by its rolling exponentially weighted average $\mathbf{m}$ computed as

$$\mathbf{m}^{t+1} \quad = \quad \gamma_1 \mathbf{m}^t + (1-\gamma_1)\nabla\mathcal{E}(\mathbf{q}^t)\,, \tag{3.37}$$

with the exponential decay parameter $\gamma_1 \in [0;1]$. This substitution brings the following update rule

$$\theta_{i,l}^{t+1} \quad = \quad \eta \cdot m_{i,l}^t + (1-\eta)\cdot\theta_{i,l}^t\,. \tag{3.38}$$

We will refer to this approach as `OURS-MOMENTUM`.

3.3.6 ADAM

The ADAM method Kingma and Ba [2014] has become very popular in deep learning. Our framework makes it easy to use for mean-field inference as well by appropriately choosing the matrix $\mathbf{D}^t$ at each step and combining it with the momentum technique.

We define the averaged second moment vector $\mathbf{v}$ of the natural gradient as

$$v_{i,l}^{t+1} \quad = \quad \gamma_2[\theta_{i,l}^t + \nabla\mathcal{E}(\mathbf{q}^t)_{i,l}]^2 + (1-\gamma_2)v_{i,l}^t\,, \tag{3.39}$$

where $\mathbf{v}$ is initialized to a strictly positive value and $\gamma_2 \in [0;1]$ is an exponential memory parameter for **v**.

Then, the $\mathbf{D}^t$ matrix is defined through each of its diagonal entries as

$$d_{i,l}^t = \sqrt{v_{i,l}^{t+1}}d + \epsilon - 1\,, \tag{3.40}$$

where ϵ is a fixed parameters and d controls the damping. We will refer to this method as `OURS-ADAM`.

Intuitively it is good at exploring parameter space thanks to a form of auto-annealing of the gradient. The natural gradient $\theta_t + \nabla\mathscr{E}(\mathbf{q}^t)$ is zero at a local minimum of the objective function Hoffman et al. [2013]. Therefore, close to a minimum, the proximal term $\mathbf{D}^t$ becomes small, thus allowing more exploration of the space. On the other hand, after a long period of exploration with large natural gradients, more damping will tend to make the algorithm converge.

3.4 Experimental Evaluation

In this section, we evaluate our method on a variety of inference problems and demonstrate that in most cases it yields faster convergence and better minima. All the code, including our efficient GPU mean-field inference framework, will be made publicly available.

3.4.1 Baselines and Variants

We compare several variants of our approach to some of the baselines we introduced in the related work section. The baselines we consider are as follows:

- `SWEEP`. As discussed in Section 3.2.2, it involves sequential coordinate descent Bishop [2006] and is not always computationally tractable for large problems.
- `ADHOC`. As discussed in Section 3.2.2, it performs parallel updates with the *ad hoc* damping parameter η of Equation 3.9 chosen manually.
- `FULL-PARALLEL`. As also discussed in Section 3.2.2, it relies on the inference described in Krähenbühl and Koltun [2013]. For example, the popular `densecrf`

framework Krähenbühl and Koltun [2011] uses this approach.

We compare to these the following variants of our approach:

- OURS-FIXED. Damping occurs in the space of natural parameters instead of mean ones as described in Section 3.3.3.
- OURS-ADAPTIVE. Adaptive and anisotropic damping in the space of natural parameters as described in Section 3.3.4.
- OURS-MOMENTUM. Similar to OURS-ADAPTIVE, but using the momentum method instead of ordinary gradient descent, as described in Section 3.3.5. We use the same parameter value $\gamma_1 = 0.95$ for all datasets.
- OURS-ADAM. Similar to OURS-ADAPTIVE but using the ADAM method instead of ordinary gradient as described in Section 3.3.6. We use the same parameters as in the original publication Kingma and Ba [2014], $\gamma_1 = 0.99$, $\gamma_2 = 0.999$ and $\epsilon =$1E-8 for all datasets.

All four methods involve a parameter $\eta = \frac{1}{1+d}$, defined in Equation 3.24 for OURS-FIXED, Equation 3.36 for OURS-ADAPTIVE, Equation 3.38 for OURS-MOMENTUM and Equation 3.40 for OURS-ADAM. Additionally, in Section 3.4.3 and Figure 3.2 we demonstrate that our method is less sensitive to the choice of this parameter than its competitors.

3.4.2 Experimental Setup

We evaluated all the methods first on a set of standardized benchmarks Frostig et al. [2014]: DBN, containing 108 instances of deep belief networks (on average 920 variables), GRID, containing 21 instances of two-dimensional grids (1600 variables), and SEG, containing 100 instances of segmentation problems (230 variables), where each instance is represented as a binary pairwise random field.

We then consider three realistic Computer Vision tasks that all involve minimizing a functional of the form given in Equation 3.7. We describe them below.

Characters Inpainting We consider character inpainting, formulated as a binary pairwise random field, Decision Tree Fields (DTF, Nowozin et al. [2011]). The dataset contains 100 test instances of occluded characters, and the goal is to restore the occluded part, as shown in the last row of Figure 3.1. We use pre-computed potentials provided by Nowozin et al. [2011]. Note, that this model consists of data-driven potentials, and includes both short and long-range interactions, which makes it particularly interesting from the optimization perspective.

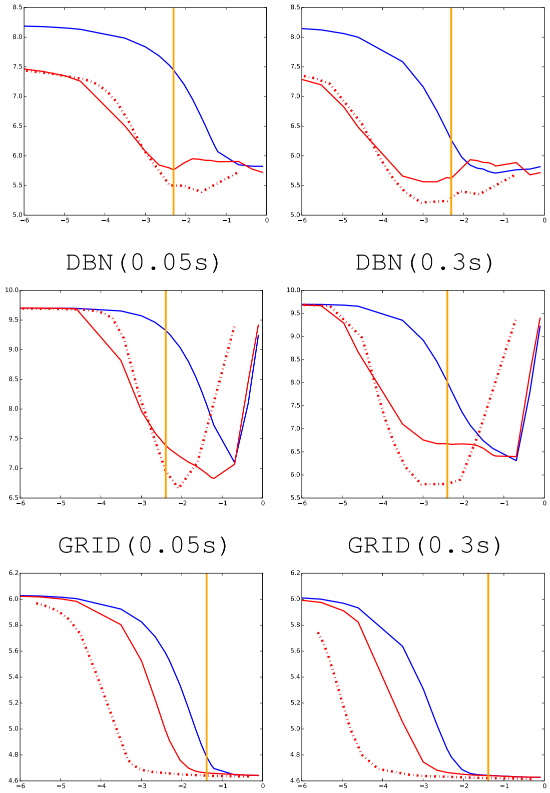
People Detection We consider detecting upright people in a multi-camera settings, using the Probabilistic Occupancy Map approach (POM, Fleuret et al. [2008]), that relies on a random field with high-order repulsive potentials, which models background subtraction signal given the presences of people in the environment. We evaluate it on the ISSIA D'Orazio et al. [2009] dataset, which contains 3000 frames of a football game, captured by 6 cameras located on two sides of the field. The original work Fleuret et al. [2008] does not explicitly mention it, but the publicly available implementation uses the `ADHOC` damping method. We implement all our methods and remaining baselines directly in this code of Fleuret et al. [2008].

Semantic Segmentation We consider semantic segmentation on PASCAL VOC 2012 dataset Everingham et al. [2012], which defines 20 object classes and 1 background class. We based our evaluation on DeepLab-CRF model Chen et al. [2015], which is currently one of the best-performing methods. This model uses CNNs to obtain unary potentials, and then employs `densecrf` of Krähenbühl and Koltun [2013] with dense pairwise potentials. However, this basic CRF model does not contain any strong repulsive terms, and thus we expect `densecrf`'s standard inference, `FULL-PARALLEL`, to work well. To improve performance, we additionally introduced co-occurrence potentials Vineet et al. [2014], which, as we will show, violate the conditions assumed in `densecrf`, but can still be successfully handled by our method. Intuitively, these co-occurrence terms put priors on the sets of classes that can appear together. We made minor modifications of `densecrf` to support both our inference and co-occurrence potentials.

We performed all the experiments on Intel(R) Xeon(R) CPU E5-2680 2.50GHz, and a GPU GeForce GTX TITAN X (12GB GRAM).

	DBN			GRID			SEG		
method	0.05s	0.30s	1.00s	0.05s	0.30s	1.00s	0.05s	0.30s	1.00s
SWEEP	-112.94	-2088.07	-2138.13	-5540.59	-16675.55	-18592.26	78.81	75.50	75.50
FULL-PARALLEL	-1952.52	-1951.54	-1942.86	-2564.39	-2777.33	-2439.08	75.66	75.66	75.66
ADHOC	-2047.31	-2047.31	-2047.31	-18345.42	-18348.80	-18349.03	76.10	75.66	75.66
OURS-FIXED	-2081.91	-2081.91	-2081.91	-18213.81	-18219.42	-18219.45	77.17	75.61	75.61
OURS-ADAPTIVE	-2125.48	-2130.61	-2130.61	-18245.93	-18252.48	-18252.48	77.68	75.64	75.61
OURS-MOMENTUM	**-2260.98**	**-2362.14**	**-2374.51**	-18143.48	**-19074.45**	**-19184.37**	74.35	73.75	73.75
OURS-ADAM	-2107.98	-2107.93	-2107.93	**-18617.06**	-18732.59	-18740.36	**72.37**	**72.32**	**72.32**

Table 3.1 – Results for KL minimization for three benchmark datasets Frostig et al. [2014]: DBN (deep belief networks), GRID (two-dimensional grids), SEG (binary segmentation). All the numbers are KL divergence (lower is better) averaged over the instances.



DBN(0.05s) DBN(0.3s)

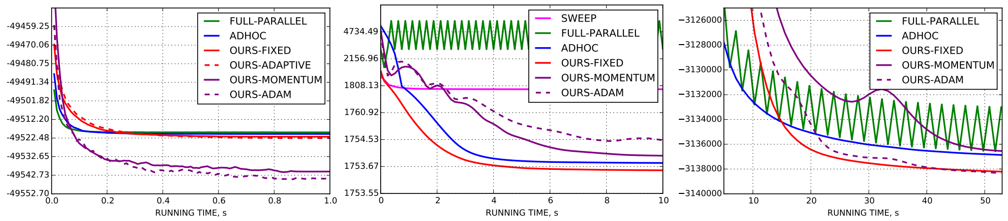
GRID(0.05s) GRID(0.3s)

SEG(0.05s) SEG(0.3s)

Figure 3.2 – Sensitivity of OURS-FIXED (red) and OURS-ADAPTIVE (dashed red) vs ADHOC (blue) to the damping parameter $\eta = \frac{1}{1+d}$. We report KL-divergence (lower is better) vs the value of the parameter, both in log-space.

3.4.3 Comparative Results

In order to understand how the methods behave in practical settings, when the available computational time is limited, we evaluate all methods for several computational *budgets*.



(a) Characters Inpainting (b) People Detection (c) Semantic Segmentation

Figure 3.3 – Convergence results. (a) OURS−ADAM and OURS−MOMENTUM converge very fast to a much better minima. (b) OURS−FIXED outperforms ADHOC both in terms of speed of convergence and the value of the objective. (c) OURS−ADAM and OURS−FIXED show the best performance. The former converges a bit slower, but in the end provide slightly better minima. ADHOC for this dataset converges rather fast, but fails to find a better optima.

The shortest budget corresponds to the early-stopping scenario after few iterations, the longest one roughly models the time until convergence, and the middle one is around 20-30% of the longest.

Benchmarks Quantitative results are given in Table 3.1. Our methods systematically outperform the ADHOC damping method. The SWEEP method usually provides good performance, but is generally slow due to its sequential nature.

Figure 3.2 shows that our methods are less sensitive to damping parameter changes than ADHOC. In Figure 3.2, the vertical orange lines corresponds to the choice of the damping parameter according to $d = L$, which can be computed directly by the power-method. Interstingly, for the GRID dataset, which includes strong repulsive potentials, algorithms do not produce reasonable results when no damping is applied. On the other hand, for the segmentation task, SEG, all the algorithms work well even without damping, in accordance with the results of Krähenbühl and Koltun [2013] or Section 3.3.3.

Characters Inpainting Quantitative results in terms of average pixel accuracy and KL-divergence are given in Table 3.2 and Figure 3.3 (a). Our method, especially when used with more advanced gradient descent schemes, outperforms all the baselines. SWEEP shows relatively good performance, but does not scale as well in terms of the running time. See the bottom row of Figure 3.1 for an example of a result.

	0.05s		0.3s		3s	
method	KL	PA	KL	PA	KL	PA
SWEEP	-6342.56	54.57	-25233.54	58.38	-49519.33	62.50
FULL-PARALLEL	**-49516.98**	60.99	-49519.27	62.00	-49519.33	62.05
ADHOC	-49514.27	61.46	-49520.09	62.15	-49520.20	62.17
OURS-FIXED	-49505.59	60.99	-49520.33	62.26	-49521.71	62.35
OURS-ADAPTIVE	-49503.43	60.93	-49520.14	62.32	-49522.49	62.60
OURS-MOMENTUM	-49513.57	63.69	-49536.67	65.26	-49540.76	65.95
OURS-ADAM	-49516.02	**65.36**	**-49538.84**	**67.03**	**-49544.58**	**67.12**

Table 3.2 – Results for characters inpainting problem Nowozin et al. [2011] based on DTFs. PA is the pixel accuracy for the occluded region (bigger is better). Our methods outperform the baselines by a margin of 3-5%. Since FULL-PARALLEL is not damped, it gets to low KL-divergence value quickly, however the actual solution is significantly worse.

	0.5s		1.3s		5s	
method	KL	MODA	KL	MODA	KL	MODA
SWEEP	1865.43	**0.630**	1795.66	0.656	1795.60	0.656
FULL-PARALLEL	2573.79	0.000	2573.79	0.000	8500.90	0.030
ADHOC	2573.79	0.308	1760.02	0.781	1753.71	**0.829**
OURS-FIXED	**1783.63**	0.626	**1754.55**	**0.802**	**1753.63**	**0.829**
OURS-MOMENTUM	1931.36	0.040	1797.19	0.650	1753.83	0.826
OURS-ADAM	2008.52	0.021	1813.66	0.501	1754.52	0.824

Table 3.3 – Results for people detection task D'Orazio et al. [2009] based on POM Fleuret et al. [2008]. OURS-FIXED outperforms the baselines and adaptive methods. This means that this problem does not require more sophisticated parameter exploration techniques.

People Detection Quantitative results, presented in Table 3.3 and Figure 3.3 (b), demonstrate that our method with a fixed step size, OURS-FIXED, brings both faster convergence and better performance. Thanks to our optimization scheme, the time required to get a Multiple Object Detection Accuracy (MODA, Bernardin and Stiefelhagen [2008]) within 3% of the value at convergence is reduced by a factor of two. This can

	5s		15s		50s	
method	KL	I/U	KL	I/U	KL	I/U
FULL-PARALLEL [o]	–		–		–	
		67.18		67.70		68.00
OURS-ADAM [o]	–		–		–	
		66.45		67.50		68.07
FULL-PARALLEL	**-**		-		-	
	3129799	67.21	3134437	67.72	3133010	68.01
ADHOC	-		-		-	
	3129469	67.19	3134557	67.73	3136865	68.04
OURS-FIXED	-		**-**		-	
	3100079	**67.76**	**3135225**	**68.18**	3138206	68.44
OURS-MOMENTUM	-		-		-	
	3060405	66.20	3128121	67.39	3136543	68.18
OURS-ADAM	-		-		**-**	
	3091787	67.08	3131624	68.02	**3138335**	**68.47**

Table 3.4 – Results for semantic segmentation problem Everingham et al. [2012] based on DeepLab-CRF Chen et al. [2015]. For all the budgets, our method obtains better segmentation accuracy. Again, FULL-PARALLEL obtains lower KL faster, with a price of reduced performance. On the top, we provide results for the original DeepLab-CRF model without co-occurrence potentials (denoted by [o]), for which the KL divergence has therefore a different meaning and is not shown.

be of big practical importance for surveillance applications of the algorithm BenShitrit et al. [2014], Bagautdinov et al. [2015], in which it is required to run in real-time. SWEEP exhibits much worse performance than our parallel method because of its greedy behavior.

Semantic Segmentation Quantitative results are presented in Table 3.4 and Figure 3.3 (c). We observe that a similar oscillation issue as noted by Vineet et al. [2014] starts happening when the FULL-PARALLEL method is used in conjunction with co-occurrence potentials, producing even worse results than without those. Using our convergent inference method fixes oscillations and provides an improvement of 0.5% in the average Intersection over Union measure (I/U) compared to the basic method without co-occurrence. What it represents is a big improvement in performance, as the ones shown in Fig 3.1, for at least 30-40 images out of total 1449. Note also, that we obtain this improvement with minimal changes in the original code. By contrast, authors Chen et al. [2015] get similar or smaller improvements by significantly augmenting the training set or by exploiting multi-scale features, which leads to additional computational burden.

3.5 Chapter Conclusion

We have presented a principled and efficient way to do parallel mean-field inference in discrete random fields. We have demonstrated that proximal gradient descent is a powerful theoretical framework for mean-field inference, which unifies and sheds light on existing approaches. Moreover, it naturally allows to incorporate existing adaptive gradient descent techniques, such as ADAM, to mean-field methods. As shown in our experiments, it often brings dramatic improvements in performance. Additionally, we have demonstrated, that our approach is less sensitive to the choice of parameters.

Our method makes it possible to use mean-field inference with a wider range of potential functions, which was previously unachievable due to the lack of convergent optimization. This new optimization method will be used as a new standard throughout this thesis.

4 Multi-Modal Mean-Fields via Cardinality-Based Clamping

4.1 Introduction

The mean-field (MF) modeling technique has been central to statistical physics for a century. Its ability to handle stochastic models involving millions of variables and dense graphs has attracted much attention in the computer vision community. It is routinely used for tasks as diverse as detection [Fleuret et al., 2008, Bagautdinov et al., 2015], segmentation [Saito et al., 2012, Krähenbühl and Koltun, 2013, Chen et al., 2015, Zheng et al., 2015], denoising [Cho et al., 2000, Nowozin et al., 2011, Li and Zemel, 2014], depth from stereo [Fransens et al., 2006, Krähenbühl and Koltun, 2013] and pose-estimation [Vineet et al., 2013].

MF approximates a “true” probability distribution by a fully-factorized one that is easy to encode and manipulate [Koller and Friedman, 2009]. The true distribution is usually defined in practice through a Conditional Random Field (CRF), and may not be representable explicitly, as it involves complex inter-dependencies between variables. In such a case the MF approximation is an extremely useful tool.

While this drastic approximation often conveys the information of interest, usually the marginal distributions, the true distribution may concentrate on configurations that are very different, equally likely, and that cannot be jointly encoded by a product law. Section 4.3 depicts such a case where groups of variables are correlated and may take one among many values with equal probability. In this situation, MF will simply pick one valid configuration, which we call a mode, and ignore the others. So-called structured mean-field methods Saul and Jordan [1995], Bouchard-Côté and Jordan [2009] can help overcome this limitation. This can be effective but requires arbitrary choices in the design of a simplified sub-graph for each new problem, which can be impractical especially if the initial CRF is very densely connected.

Here we introduce a novel way to automatically add structure to the MF approximation and show how it can be used to return several potentially valid answers in ambiguous situations. Instead of relying on a single fully factorized probability distribution, we introduce a mixture of such distributions, which we will refer to as *Multi-Modal Mean Field* (MMMF).

We compute this MMMF by partitioning the state space into subsets in which a standard MF approximation suffices. This is similar in spirit to the approach of Weller and Domke [2015] but a key difference is that our clamping acts simultaneously on arbitrarily sized groups of variables, as opposed to one at a time. We will show that when dealing with large CRFs with strong correlations, this is essential. The key to the efficiency of MMMF is how we choose these groups. To this end, we introduce a temperature parameter that controls how much we smooth the original probability distribution before the MF approximation. By doing so for several temperatures, we spot groups of variables that may take different labels in different modes of the distribution. We then force the optimizer to explore alternative solutions by clamping them, that is, forcing them to take different values. Our temperature-based approach, unlike the one of Weller and Domke [2015], does not require *a priori* knowledge of the CRF structure and is therefore compatible with "black box" models.

In the remainder of the chapter, we will describe both MF and MMMF in more details. We will then demonstrate that MMMF outperforms both MF and the clamping method of Weller and Domke [2015] on a range of tasks.

4.2 Related Work

Conditional Random Fields (CRFs) are often used to represent correlations between variables Wang et al. [2013]. Mean-field inference is a means to approximate them in a computationally efficient way. We briefly review both techniques below.

4.2.1 Conditional Random Fields

As will be shown in Section 4.3, the mean-field approximation model sometimes comes at the cost of downplaying the dependencies between variables. The *DivMBest* method Ramakrishna and Batra [2012], Batra et al. [2012] addresses this issue starting from the following observation: When looking for an assignment in a graphical model, the resulting MAP is not necessarily the best because the probabilistic model may not capture all that is known about the problem. Furthermore, optimizers can get stuck in local minima. The proposed solution is to sequentially find several local optima and force them to be different from each other by introducing diversity constraints in the objective function. It has recently been shown that it is provably more effective to solve for diverse MAPs jointly but under the same set of constraints Kirillov et al. [2015]. However, none of these methods provide a generic and practical way to choose local constraints to be enforced over variable sub-groups. Furthermore, they only return a set of MAPs. By contrast, our approach yields a multi-modal approximation of the posterior distribution, which is a much richer description and which we will show to be useful.

Another approach to improving the MF approximation is to decompose it into a mixture of product laws by "clamping" some of the variables to fixed values, and finding for each set of values the best factorized distribution under the resulting deterministic conditioning. By summing the resulting approximations of the partition function, one can provably improve the approximation of the true partition function Weller and Domke [2015]. This procedure can then be repeated iteratively by clamping successive variables but is only practical for relatively small CRFs. At each iteration, the variable to be clamped is chosen on the basis of the graphical model weights, which requires intimate knowledge about its internals, which is not always available.

Our own approach is in the same spirit but can clamp multiple variables at a time without requiring any knowledge of the graph structure or weights.

Finally, *DivMBest* approaches do not provide a way to choose the best solution without looking at the ground-truth, except for the one of Yadollahpour et al. [2013] that relies on training a new classifier for that purpose. By contrast, we will show that the multi modal Bayesian nature of our output induces a principled way to use temporal consistency to solve directly practical problems.

4.3 Motivation

To motivate our approach, we present here a toy example that illustrates a typical failure mode of the standard MF technique, which ours is designed to prevent. Figure 4.1 depicts a CRF where each pixel represents a binary variable connected to its neighbors by attractive pairwise potentials.

For the sake of illustration, we split the grid into four zones as follows. The attractive terms are weak on left side but strong on the right. Similarly, in the top part, the unary terms favor value of 1 while being completely random in the bottom part.

The unary potentials are depicted at the top left of Figure 4.1 and the result of the standard MF approximation at the bottom in terms of the probability of the pixels being assigned the label 1. In the bottom right corner of the grid, because the interaction potentials are strong, all pixels end up being assigned high probabilities of being 1 by MF, where they could just as well have all been assigned high probabilities to be zero. We explain below how our MMMF algorithm can produce *two* equally likely modes, one with all pixels being zero with high probability and the other with all pixel being one with high probability.

4.4 Multi-Modal Mean-Fields

Given a CRF defined with respect to a graphical model and the probability $P(\mathbf{X} = \mathbf{x})$, recall that $\mathscr{X}$ denotes the set of all possible states of the vector $\mathbf{x}$. The standard MF approximation only models a single mode of the P, as discussed in Section **??**. We therefore propose to create a richer representation that accounts for potential multiple modes by replacing the fully factorized distribution of Equation 2.30 by a weighted mixture of such distributions that better minimizes the KL-divergence to P.

The potential roadblock is the increased difficulty of the minimization problem. In this

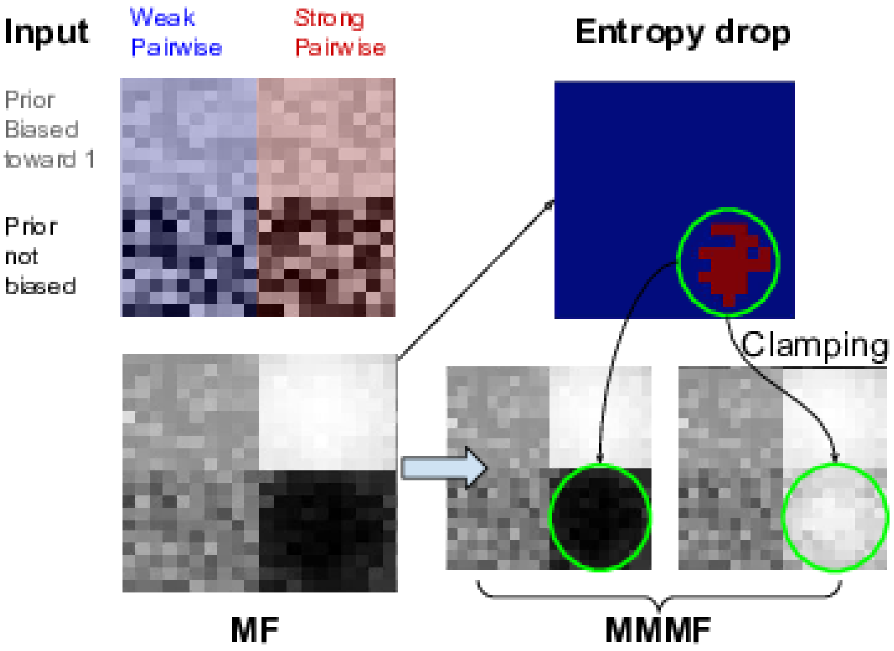


Figure 4.1 – A typical failure mode of MF resolved by MMMF. Grey levels indicate marginal probabilities, under the prior (Input) and under the product laws (MF and MMMF).

section, we present an overview of our approach to solving it, and discuss its key aspects in the following two.

Formally, let us assume that we have partitioned $\mathscr{X}$ into disjoint subsets $\mathscr{X}_k$ for $1 \le k \le K$. We replace the original mean-field (MF) approximation by one of the form

$$\begin{aligned} P(\mathbf{X}=\mathbf{x}) \;\approx\; Q_{MM}(\mathbf{X}=\mathbf{x}) &= \sum_k m_k Q_k(\mathbf{x})\,, \\ Q_k(\mathbf{x}) &= \prod_i \mathbf{q}_i^k(x_i)\,, \end{aligned} \tag{4.1}$$

where Q_k is a MF approximation for the states $\mathbf{x} \in \mathscr{X}_k$ with individual probabilities $\mathbf{q}_i^k$ that variable i can take value x_i in a set of labels $\mathscr{L}$, and m_k is the probability that a state belongs to $\mathscr{X}_k$.

We can evaluate the m_k and q_i^k values by minimizing the KL-divergence between Q_{MM} and P. The key to making this computation tractable is to guarantee that we can evaluate the $\mathbf{q}_i^k$ parameters on each subset separately by performing a standard MF approximation for each. One way to achieve that is to constrain the support of the Q_k distributions to be disjoint, that is,

$$\forall k \neq k', Q_{k'}(\mathscr{X}_k) \quad = \quad 0\,. \tag{4.2}$$

In other words, each MF approximation is *specialized* on a subset $\mathscr{X}_k$ of the state space and is computed to minimize the KL-Divergence there. In practice, we enrich our approximation by recursively splitting a set of states $\mathscr{X}_k$ among our partition $\mathscr{X}_1,\ldots,\mathscr{X}_K$ into two subsets $\mathscr{X}_k^1$ and $\mathscr{X}_k^2$ to obtain the new partition $\mathscr{X}_1,\ldots,\mathscr{X}_{k-1},\mathscr{X}_k^1,\mathscr{X}_k^2,\mathscr{X}_{k+1},\ldots,\mathscr{X}_K$, which is then reindexed from 1 to $K+1$. Initially, $\mathscr{X}_k$ represents the whole state space. Then we take it to be the newly created subset in a breadth-first order until a preset number of subsets has been reached. Each time, the algorithm proceeds through the following steps:

- It finds groups of variables likely to have different values in different modes of the distribution using an entropy-based criterion for the $\mathbf{q}_i^k$.
- It partitions the set into two disjoint subsets according to a clause that sets a threshold on the number of variables in this group that take a specific label. $\mathscr{X}_k^1$ will contain the states among $\mathscr{X}_k$ that meet this clause and $\mathscr{X}_k^2$ the others.
- It performs an MF approximation within each subset independently to compute parameters $\mathbf{q}_i^{k,1}$ and $\mathbf{q}_i^{k,2}$ for each of them. This is done by a standard MF approximation, to which we add the disjointness constraint 4.2.

This yields a binary tree whose leaves are the $\mathscr{X}_k$ subsets forming the desired state-space partition. Given this partition, we can finally evaluate the m_k. In Section 4.5, we introduce our cardinality based criterion and show that it makes minimization of the KL-divergence possible. In Section 4.6, we show how our entropy-based criterion selects, at each iteration, the groups of variables on which the clauses depend.

4.5 Partitioning the State Space

In this section, we describe the cardinality-based criterion we use to recursively split state spaces and explain why it allows efficient optimization of the KL-divergence $\text{KL}(Q_{MM} \| P)$, where Q_{MM} is the mixture of Equation 4.1.

4.5.1 Cardinality Based Clamping

The state space partition $\mathcal{X}_{k\,,\,1 \leq k \leq K}$ introduced above is at the heart of our approximation and its quality and tractability critically depend on how well chosen it is. In Weller and Domke [2015], each split is obtained by clamping to zero or one the value of a single binary variable. In other words, given a set of states $\mathcal{X}_k$ to be split, it is broken into subsets $\mathcal{X}_k^1 = \{\mathbf{x} \in \mathcal{X}_k | x_i = 0\}$ and $\mathcal{X}_k^2 = \{\mathbf{x} \in \mathcal{X}_k | x_i = 1\}$, where i is the index of a specific variable. To compute a mean-field approximation to P on each of these subspaces, one only needs to perform a standard mean-field approximation while constraining the $\mathbf{q}_i$ probability assigned to the clamped variable to be either zero or one. However, this is limiting for the large and dense CRFs used in practice because clamping only one variable among many at a time may have very little influence overall. Pushing the solution towards a qualitatively different minimum that corresponds to a distinct mode may require simultaneously clamping many variables.

To remedy this, we retain the clamping idea but apply it to groups of variables instead of individual ones so as to find new modes of the posterior while keeping the estimation of the parameters m_k and q_i^k computationally tractable. More specifically, given a set of states $\mathcal{X}_k$ to be split, we will say that the split into $\mathcal{X}_k^1$ and $\mathcal{X}_k^2$ is cardinality-based if

$$\mathcal{X}_k^1 = \{\mathbf{x} \in \mathcal{X}_k \text{ s.t. } \sum_{u=1\ldots L} \mathbb{1}(\mathbf{x}_{i_u} = v_u) \geq C\}\,, \tag{4.3}$$

$$\mathcal{X}_k^2 = \{\mathbf{x} \in \mathcal{X}_k \text{ s.t. } \sum_{u=1\ldots L} \mathbb{1}(\mathbf{x}_{i_u} = v_u) < C\}\,, \tag{4.4}$$

where the $i_1, \ldots, i_L$ denote groups of variables that are chosen by the entropy-based criterion and $v_1, \ldots, v_L$ is a set of labels in $\mathcal{L}$. In other words, in one of the splits, more than C of the variables have the assigned values and in the other less than C do. For example, for semantic segmentation $\mathcal{X}_k^1$ would be the set of all segmentations in $\mathcal{X}_k$ for which at least C pixels in a region take a given label, and $\mathscr{X}_k^2$ the set of all segmentations for which less than C pixels do.

We will refer to this approach as *cardinality clamping* and will propose a practical way to select appropriate $i_1,\ldots,i_L$ and $v_1,\ldots,v_L$ for each split in Section 4.6.

4.5.2 Instantiating the Multi-Modal Approximation

The *cardinality clamping* scheme introduced above yields a state space partition $\mathscr{X}_{k\,,\,1\leq k\leq K}$. We now show that given such a partition, minimizing the KL-divergence $\mathrm{KL}Q_{MM}\|P)$ using the multi-modal approximation of Equation 4.1 under the disjointness constraint, becomes tractable.

In practice, we relax the constraint 4.2 to *near* disjointness

$$\forall k\neq k', Q_{k'}(\mathscr{X}_k)\leq\epsilon\,, \tag{4.5}$$

where ϵ is a small constant. It makes the optimization problem better behaved and removes the need to tightly constrain any individual variable, while retaining the ability to compute the KL divergence up to $\mathscr{O}(\epsilon\log(\epsilon))$.

Let $\hat{m}$ and $\hat{q}$ stand for all the m_k and q_i^k parameters that appear in Equation 4.1. We compute them as

$$\begin{aligned}\min_{\hat{m},\hat{q}}\mathrm{KL}(Q_{MM}\|P) &= \min_{\hat{m},\hat{q}}\sum_{\mathbf{x}\in\mathscr{X}}\sum_{k\leq K}m_kQ_k(\mathbf{x})\log\left(\frac{Q_{MM}(\mathbf{x})}{P(\mathbf{x})}\right)\\ &\equiv \min_{\hat{m}}\sum_{k\leq K}m_k\log(m_k)-\sum_{k\leq K}m_kA_k\,, & (4.6)\\ \text{where}\qquad A_k &= \max_{q_i^k,i=1\ldots N}\sum_{\mathbf{x}\in\mathscr{X}}Q_k(\mathbf{x})\log\left(\frac{e^{-E(\mathbf{x})}}{Q_k(\mathbf{x})}\right) & (4.7)\end{aligned}$$

where A_k is maximized under the near-disjointness constraint of Equation 4.9.

As proved formally in the next section, the second equality of Equation 4.6 is valid up to a constant and after neglecting a term of order $\mathscr{O}(\epsilon\log\epsilon)$ which appears under the *near* disjointness assumption of the supports. Given the A_k terms of Equation 4.7 and under the constraints that the mixture probabilities $\hat{m}$ sum to one, we must have

$$m_k = \frac{e^{A_k}}{\sum\limits_{k' \leq K} e^{A_{k'}}} , \tag{4.8}$$

and we now turn to the computation of these A_k terms. We formulate it in terms of a constrained optimization problem as follows.

4.5.3 Minimising the KL-Divergence

Let us see how the KL-Divergence between Q_{MM} and P of Equation 4.7, can be minimised with respect to the parameters m_k and to the distributions Q_k, leading to Equation 9. We reformulate the minimisation problem up to a constant approximation factor of order $\epsilon \log(\epsilon)$.

First, remember that our minimisation problem enforces the near-disjointness condition,

$$\forall k \neq k' \sum_{\mathbf{x} \in \mathscr{X}'_k} Q_k(\mathbf{x}) \leq \epsilon , \tag{4.9}$$

between the elements of the mixture.

Let us then prove the following useful Lemma.

Lemma 6 *For all mixture element $k \leq K$,*

$$\sum_{\mathbf{x} \in \mathscr{X}} Q_k(\mathbf{x}) \log \left(\sum_{k' \leq K} m_{k'} Q_{k'}(\mathbf{x}) \right) = \sum_{\mathbf{x} \in \mathscr{X}} Q_k(\mathbf{x}) \log (m_k Q_k(\mathbf{x})) + \mathscr{O}(\epsilon \log \epsilon) . \tag{4.10}$$

Proof Let k be the index of a mixture component $k \leq K$, and let us denote the approximation error

$$\delta_k = \sum_{\mathbf{x} \in \mathscr{X}} Q_k(\mathbf{x}) \log \left(\sum_{k' \leq K} m_{k'} Q_{k'}(\mathbf{x}) \right) - \sum_{\mathbf{x} \in \mathscr{X}} Q_k(\mathbf{x}) \log (m_k Q_k(\mathbf{x})) . \tag{4.11}$$

Then, we use the near-disjointness condition to bound δ_k,

$$\delta_k \leq \underbrace{\sum_{\mathbf{x}\in\mathcal{X}_k} Q_k(\mathbf{x}) \log\left(1 + \frac{\sum\limits_{k'\neq k} m_{k'} Q_{k'}(\mathbf{x})}{Q_k(\mathbf{x})}\right)}_{I} + \underbrace{\sum_{\mathbf{x}\in\mathcal{X}\setminus\mathcal{X}_k} Q_k(\mathbf{x}) \log\left(1 + \frac{\sum\limits_{k'\neq k} m_{k'} Q_{k'}(\mathbf{x})}{Q_k(\mathbf{x})}\right)}_{J} \tag{4.12}$$

We first use the well known inequality $\log(1+x) \leq x$ in order to upper bound I,

$$I \leq \sum_{\mathbf{x}\in\mathcal{X}_k} Q_k(\mathbf{x}) \frac{\sum\limits_{k'\neq k} m_{k'} Q_{k'}(\mathbf{x})}{Q_k(\mathbf{x})} \tag{4.13}$$

$$\leq \sum_{k'\neq k} \sum_{\mathbf{x}\in\mathcal{X}_k} m_{k'} Q_{k'}(\mathbf{x}) \tag{4.14}$$

$$\leq \sum_{k'\neq k} \epsilon \tag{4.15}$$

$$\leq \mathcal{O}(\epsilon)\,. \tag{4.16}$$

The second term, J, can then be upper-bounded using the fact that the $m_{k'}$ and $Q_{k'}$ are mixture weights and probabilities and hence $\sum\limits_{k'\neq k} m_{k'} Q_{k'}(\mathbf{x}) \leq 1$ for all $\mathbf{x}$. Therefore,

$$J \leq \sum_{\mathbf{x}\in\mathcal{X}\setminus\mathcal{X}_k} Q_k(\mathbf{x}) \log\left(1 + \frac{1}{Q_k(\mathbf{x})}\right) \tag{4.17}$$

$$\leq \sum_{\mathbf{x}\in\mathcal{X}\setminus\mathcal{X}_k} -Q_k(\mathbf{x}) \log\left(Q_k(\mathbf{x})\right) \tag{4.18}$$

$$\leq \sum_{k'\neq k} \sum_{\mathbf{x}\in\mathcal{X}_{k'}} -Q_k(\mathbf{x}) \log\left(Q_k(\mathbf{x})\right)\,. \tag{4.19}$$

Furthermore, for all $k' \neq k$, the near-disjointness condition enforces that $\sum\limits_{\mathbf{x}\in\mathcal{X}_{k'}} Q_k(\mathbf{x}) \leq \epsilon$. Under this constraint, on each of the subsets $\mathcal{X}_{k'}$, the maximal entropy is reached if $Q_k(\mathbf{x}) = \dfrac{\epsilon}{|\mathcal{X}_{k'}|}$ for all $\mathbf{x}$ in $\mathcal{X}_{k'}$. And, therefore

$$\sum_{\mathbf{x}\in\mathcal{X}_{k'}} -Q_k(\mathbf{x}) \log\left(Q_k(\mathbf{x})\right) \leq \epsilon \log\left(\frac{|\mathcal{X}_k'|}{\epsilon}\right) \tag{4.20}$$

$$\leq \mathcal{O}(\epsilon \log \epsilon) + \mathcal{O}(\epsilon)\,, \tag{4.21}$$

where the factor $\log(|\mathcal{X}_k|)$, which is of the order of the number of variables, has been integrated in the constant.

Hence,

$$
\begin{aligned}
J &\leq \sum_{k' \neq k} \sum_{\mathbf{x} \in \mathcal{X}_{k'}} -Q_k(\mathbf{x}) \log\left(Q_k(\mathbf{x})\right) && (4.22)\\
&\leq \mathcal{O}(\epsilon \log \epsilon) + \mathcal{O}(\epsilon)\,, && (4.23)\\
& && (4.24)
\end{aligned}
$$

which terminates the proof.

We can then move on to the minimisation of the KL-Divergence

$$
\begin{aligned}
\min_{\hat{m},\hat{q}} \mathrm{KL}(Q_{MM} \| P) &= \min_{\hat{m},\hat{q}} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{k \leq K} Q_{MM}(\mathbf{x}) \log\left(\frac{Q_{MM}(\mathbf{x})}{P(\mathbf{x})}\right) && (4.25)\\
&= \min_{\hat{m},\hat{q}} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{k \leq K} Q_{MM}(\mathbf{x}) \log\left(\frac{Q_{MM}(\mathbf{x})}{e^{-E(\mathbf{x})}}\right) + \log(Z) && (4.26)\\
&= \min_{\hat{m},\hat{q}} \sum_{k \leq K} \sum_{\mathbf{x} \in \mathcal{X}} m_k Q_k(\mathbf{x}) \log\left(\frac{\sum_{k' \leq K} m_{k'} Q_{k'}(\mathbf{x})}{e^{-E(\mathbf{x})}}\right) + \log(Z) && (4.27)\\
&= \min_{\hat{m},\hat{q}} \sum_{k \leq K} \sum_{\mathbf{x} \in \mathcal{X}} m_k Q_k(\mathbf{x}) \log\left(\frac{m_k Q_k(\mathbf{x})}{e^{-E(\mathbf{x})}}\right) + \log(Z) + \mathcal{O}(\epsilon \log \epsilon) && (4.28)\\
&= \min_{\hat{m}} \left[\sum_{k \leq K} m_k \log m_k + \sum_{k \leq K} \min_{q_k} \sum_{\mathbf{x} \in \mathcal{X}} m_k Q_k(\mathbf{x}) \log\left(\frac{Q_k(\mathbf{x})}{e^{-E(\mathbf{x})}}\right)\right] + \log(Z) + \mathcal{O}(\epsilon \log \epsilon && (4.29)\\
&= \min_{\hat{m}} \sum_{k \leq K} m_k \log(m_k) - \sum_{k \leq K} m_k A_k + \log(Z) + \mathcal{O}(\epsilon \log \epsilon)\,, && (4.30)
\end{aligned}
$$

where,

$$
A_k = \max_{q_i^k, i=1\ldots N} \sum_{\mathbf{x} \in \mathcal{X}} Q_k(\mathbf{x}) \log\left(\frac{e^{-E(\mathbf{x})}}{Q_k(\mathbf{x})}\right).
$$

Equation 4.28 is obtained using Lemma 6.

Assuming that we are able to compute A_k, for all k, the minimisation of this KL-Divergence with respect to parameters m_k, under the nomalisation constraint

$$\sum_{k \leq K} m_k = 1 \,, \tag{4.31}$$

is then straightforward and leads to

$$m_k = \frac{e^{A_k}}{\sum\limits_{k' \leq K} e^{A_{k'}}} \,. \tag{4.32}$$

Handling Two Modes

Let us first consider the case where we generate only two modes modeled by $Q_1(\mathbf{x}) = \prod \mathbf{q}_i^1(x_i)$ and $Q_2(\mathbf{x}) = \prod \mathbf{q}_i^2(x_i)$ and we seek to estimate the q_i^1 probabilities. The q_i^2 probabilities are evaluated similarly.

Recall from Section 4.5.2 that the q_i^1 must be such that the A_1 term of Equation 4.7 is maximized subject to the near disjointness constraint of Equation 4.9, which becomes

$$Q_1\left(\sum_{u=1\ldots L} \mathbb{1}(\mathbf{X}_{i_u} = v_u) < C\right) \leq \epsilon \,, \tag{4.33}$$

under our cardinality-based clamping scheme defined by Equation 4.4. Performing this maximization using a standard Lagrangian Dual procedure Boyd and Vandenberghe [2004] requires evaluating the constraint and its derivatives. Despite the potentially exponentially large number of terms involved, we can do this in one of two ways. In both cases, the Lagrangian Dual procedure reduces to a series of unconstrained mean-field minimizations with well known additional potentials.

1. When C is close to 0 or to L, the Lagrangian term can be treated as a specific form of pattern-based higher-order potentials, as in Vineet et al. [2014], Fleuret et al. [2008], Kohli and Rother [2012], Arnab et al. [2015].

2. When C is both substantially greater than zero and smaller than L, we treat

$\sum_{u=1...L} \mathbb{1}(\mathbf{X}_{i_u} = v_u)$ as a large sum of independent random variables under Q_1. We therefore use a Gaussian approximation to replace the cardinality constraint by a simpler linear one, and finally add unary potentials to the MF problem.

We will encounter the first situation when tracking pedestrians and the second when performing semantic segmentation, as will be discussed in the results section.

Handling an Arbitrary Number of Nodes

Recall from Section 4.5 that, in the general case, there can be an arbitrary number of modes. They correspond to the leaves of a binary tree created by a succession of cardinality-based splits. Let us therefore consider mode k for $1 \leq k \leq K$. Let B be the set of branching points on the path leading to it. The *near* disjointness 4.9, can be enforced with only $|B|$ constraints. For each $b \in B$, there is a list of variables $i_1^b, \ldots, i_{L^b}^b$, a list of values $v_1^b, \ldots, v_{L^b}^b$, a cardinality threshold C^b, and a sign for the inequality $\geq_b$ that define a constraint

$$Q_k \left(\sum_{u=1...L^b} \mathbb{1}(\mathbf{X}_{i_u^b} = v_u^b) \geq_b C^b \right) \leq \epsilon \tag{4.34}$$

of the same form as that of Equation 4.33. It ensures disjointness with all the modes in the subtree on the side of b that mode k does not belong to. Therefore, we can solve the constrained maximization problem of Equation 4.7, as in Section 4.5.3, but with $|B|$ constraints instead of only one.

4.6 Selecting Variables to Clamp

We now present an approach to choosing the variables $i_1, \ldots, i_L$ and the values $v_1, \ldots, v_L$, which define the cardinality splits of Eqs. 4.3 and 4.4, that relies on phase transitions in the graphical model.

To this end, we first introduce a *temperature* parameter in our model that lets us smooth the probability distribution we want to approximate. This well known parameter for physicists Kadanoff [2009] was used in a different context in vision by Premachandran et al. [2014]. We study its influence on the corresponding MF approximation and how we can exploit the resulting behavior to select appropriate values for our variables.

4.6.1 Temperature and its Influence on Convexity

We take the temperature T to be a number that we use to redefine the probability distribution of Eq. 3.1 as

$$P^T(\mathbf{X}=\mathbf{x}) = \frac{1}{Z^T} e^{-\frac{1}{T}\mathbf{E}(\mathbf{x})}, \tag{4.35}$$

where Z^T is the partition function that normalizes P^T so that its integral is one. For $T=1$, P^T reduces to P. As T goes to infinity, it always yields the same Maximum-A-Posteriori value but becomes increasingly smooth. When performing the MF approximation at high T, the first term of the KL-Divergence, the convex negative entropy, dominates and makes the problem convex. As T decreases, the second term of the KL-Divergence, the expected energy, becomes dominant, the function stops being convex, and local minima can start to appear. In the next section, we introduce a physics-inspired proof that, in the case of a dense Gaussian CRF Krähenbühl and Koltun [2013], we can approximate and upper-bound, in closed-form, the *critical temperature* T_c at which the KL divergence stops being convex. We validate experimentally this prediction, using directly the *denseCRF* code from Krähenbühl and Koltun [2013]. This makes it easy to define a temperature range $[1, T_{max}]$ within which to look for T_c. For a generic CRF, no such computation may be possible and the range must be determined empirically.

4.6.2 Computing the Critical Temperature for the Dense Gaussian CRFs

We first compute analytically the phase transition temperature parameter T_c of 6.2 where the KL-Divergence stops being convex. In the first part *Analytical Derivation*, we make strong assumptions in order to be able to obtain a closed form estimation of T_c. We then explain how this result helps understanding real cases. In the second part *Experimental Analysis*, in order to justify our assumptions, we run experiments under three regimes, one where our assumptions are strictly verified, one which corresponds to a real-life scenario and an intermediate one. This set of experiments shows that our strong assumptions provide a valuable insight for practical applications.

Analytical derivation Let us take probability distribution P to be defined by a dense Gaussian CRF Krähenbühl and Koltun [2013]. In order to make computation tractable, we assume that the RGB distance between pixels is uniform and equal to d_{rgb}. Therefore the RGB Kernel is constant with value

$$\theta_{rgb} = e^{\frac{-d_{rgb}^2}{2\sigma_{rgb}}} . \tag{4.36}$$

We consider the case where we have only two possible labels and the same unary potential on all the variables. Even if this assumption sounds strong, we can expect them to be locally valid. Formally, on a $N \times N$ dense grid, the energy function is defined as

$$\begin{aligned} E(\mathbf{x}) = & \frac{\Gamma\theta_{rgb}}{2\pi\sigma^2} \sum_{(i,j),(i',j')} \mathbb{1}[\mathbf{x}_{(i,j)} \neq \mathbf{x}_{(i',j')}] e^{-\frac{\|(i,j)-(i',j')\|^2}{2\sigma}} \\ & + \sum_{(i,j)} U_{(i,j)} \mathbb{1}[\mathbf{x}_{(i,j)} = 0] , \end{aligned}$$

where σ controls the range of the correlations and $U_{(i,j)}$ is a unary potential.

Since that we assumed that all the variables receive the same unary U, all the variables are undiscernibles. Furthermore, the pairwise potentials are attractive, we therefore expect all the mean-field parameters $q_{i,j} = Q(\mathbf{x}_{i,j} = 0)$ to have the same value at the fixed point solution of the Mean-Field. Therefore, we designate this common parameter q^T and we can try to find analytically the Mean-Field fixed point for q^T corresponding to a temperature T.

At convergence, the parameter q^T will have to satisfy

$$\begin{aligned}\log(q^T) &= \mathbb{E}_Q(E(\mathbf{x})|x_i = 0)\\ &= -\frac{\Gamma\theta_{rgb}}{2\pi\sigma^2 T}\sum_{(i,j)\in\mathbb{Z}\times\mathbb{Z}}(1-q^T)e^{-\frac{\|(i,j)\|^2}{2\sigma}} - \frac{U}{T}\\ &= -\frac{(1-q^T)\Gamma\theta_{rgb}+U}{T}\end{aligned}$$

Hence, we obtain the fixed point equation

$$\tilde{q}^T = \frac{1}{2}\tanh\left(\frac{\tilde{q}^T\Gamma\theta_{rgb}-U}{T}\right), \tag{4.37}$$

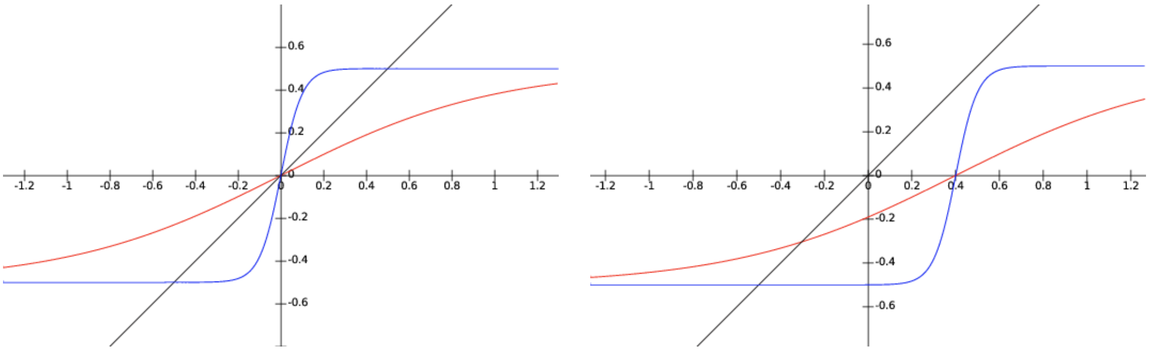
where $\tilde{q}^T = q^T - 0.5$. As depicted in Figure 4.2, when unaries are 0 (on the left) there are two distinct regimes for the solutions of this equation. For high T, there is only one stable solution at $\tilde{q} = 0$. For low T, there are two distinct stable solutions where $\tilde{q}$ is close to -0.5 or 0.5. The temperature threshold T_c where the transition happens, corresponds to the solution of

$$\frac{1}{2}\frac{d\tanh(\frac{\tilde{q}\Gamma\theta_{rgb}}{T})}{d\tilde{q}}|_{\tilde{q}=0} = 1\,, \tag{4.38}$$

and hence $T_c = \frac{\Gamma\theta_{rgb}}{2}$. For real images, we have $\theta_{rgb} \leq 1$, and therefore, $T_c = \frac{\Gamma}{2}$ can be used to upper-bound the true critical temperature.

When unaries are non-zero, there is no closed form solution for T_c, however, from Equation 4.37, we can show that the smaller the unaries (U), the lower the critical temperature will be. This is intuitively justified in Fig. 4.2.

The authors of Weller and Domke [2015], use several heuristics which basically consist in looking for high correlations and low unaries directly in the potentials of the graphical model, in order to find good variables to clamp. We, instead use a criterium based on the critical temperature in order to spot these.



Without unaries

With unaries

Figure 4.2 – $\tanh(\frac{\tilde{q}\Gamma - U}{T})$ for two temperatures. Low T (blue) and High T (red).

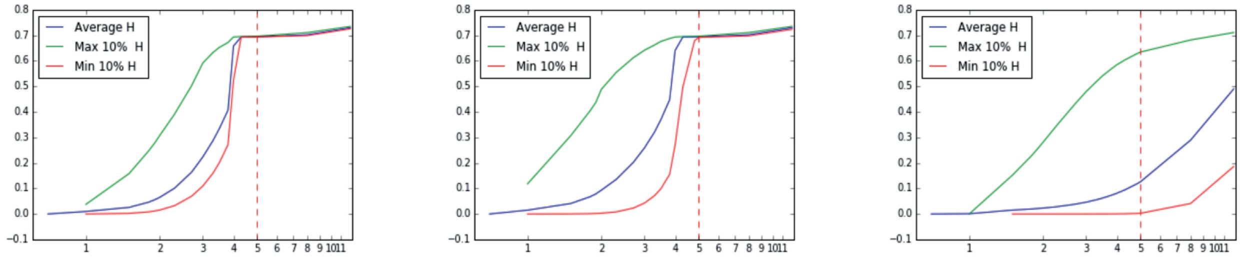
Experimental analysis We use the dense CRF implementation of Krähenbühl and Koltun [2013] to verify the phase transition experimentally for $\Gamma = 10$. In our experiments, we used the three following settings, which range from the stylised example used for calculation to real semantic segmentation problems:

- **Model 1:** We use a uniform rgb image $d_{rgb} = 0$. Two classes without unary potentials. This is exactly the model used for the derivations with $\theta_{rgb} = 1$ and $U = 0$.
- **Model 2:** Gaussian potentials defined over image coordinates distance + RGB distance. Two classes without unary potentials. In other words, $\theta_{rgb} \leq 1$.
- **Model 3:** Gaussian potentials defined over image coordinates distance + RGB distance. Two classes with unary potentials produced by a CNN. This is a real-life scenario.

Fig. 4.3 shows that, as expected, two regimes appear for Model 1, before and after $T = 5$. We see that our prediction remains completely valid for Model 2, some non-uniform regions fall under the regime $\theta_{rgb} \leq 1$ and therefore the 10 % highest entropy percentile transitions slightly earlier. For Model 3, however, we see that the minimal and average entropy remain low even for $T > 5$. This is well explained by the fact that large regions of the image receive strong unary potentials from one class or the other, and therefore fall under the case "with unaries" of Fig. 4.2 where the U parameter cannot be ignored.

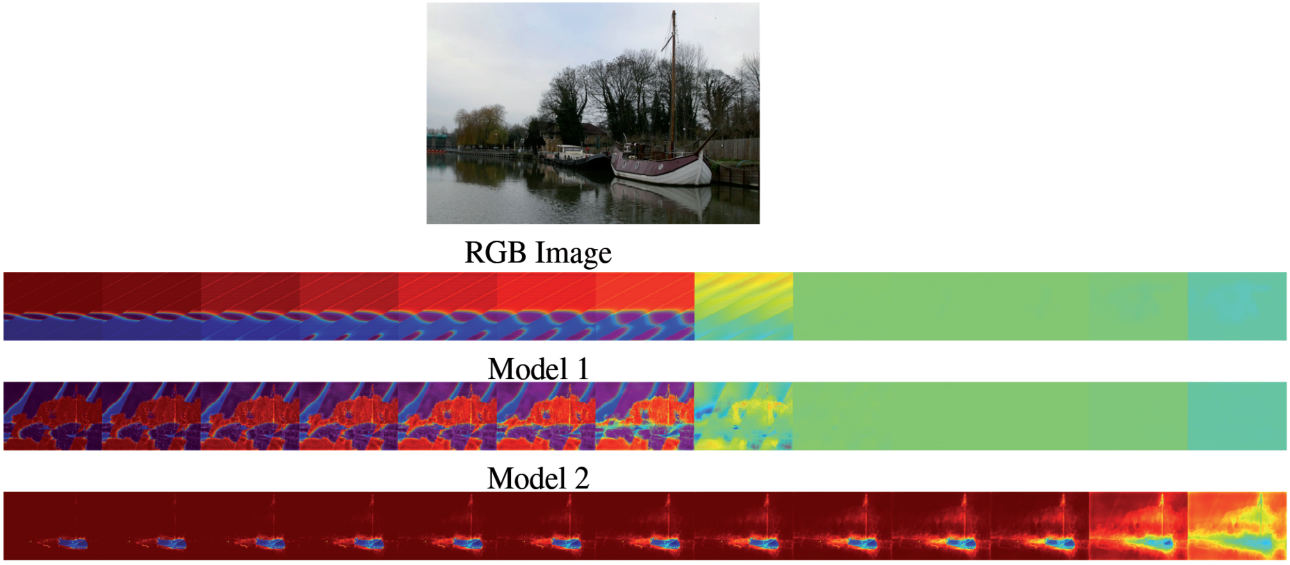
However, some uncertain regions receive unary potentials of same value for both labels, and therefore undergo a phase transition as predicted by our calculation. That is why the maximal entropy behaves similarly to Model 2. Our algorithm precisely targets these uncertain regions.

Interestingly, we see that in practice, the users of DenseCRF choose the Γ and T parameters in order to be in a Multi-Modal regime, but close to the phase transition. For instance in the public releases of Chen et al. [2015] and Zheng et al. [2015], the Gaussian kernel is set with $T = 1$ and $\Gamma = 3$.



Model 1 Model 2 Model 3

Figure 4.3 – Entropy as a function of temperature.



RGB Image

Model 1

Model 2

Model 3

Figure 4.4 – Evolution of MF probability for background label when temperature increases

4.6.3 Entropy-Based Splitting

We describe here our approach to splitting $\mathscr{X}$ into $\mathscr{X}_1$ and $\mathscr{X}_2$ at the root node of the tree. The subsequent splits are done in exactly the same way. The variables to be clamped are those whose value change from one local minimum to another so that we can force the exploration of both minima.

To find them, we start at T_{max}, a temperature high enough for the KL divergence to be convex and progressively reduce it. For each successive temperature, we perform the MF approximation starting with the estimate for the previous one to speed up the computation. When looking at the resulting set of approximations starting from the lowest temperature ones $T = 1$, a telltale sign of increasing convexity is that the assignment of some variables that were very definite suddenly becomes uncertain. Intuitively, this happens when the CRF terms that bind variables is overcome by the entropy terms that encourage uncertainty. In physical terms, this can be viewed as a local phase-transition Kadanoff [2009].

Let T be a temperature greater than 1 and let Q^T and Q^1 be the corresponding Mean Field approximations, with their marginal probabilities q_i^T and q_i^1 for each variable i. To detect such phase transitions, we compute

$$\delta_i(T) = \mathbb{1}[\mathscr{H}(q_i^T) > h_{high}]\mathbb{1}[\mathscr{H}(q_i^1) < h_{low}]\,, \tag{4.39}$$

for all i, where $\mathscr{H}$ denotes the individual entropy.

All variables and labels with positive δ_i become candidates for clamping. If there are none, we increase the temperature. If there are several, we can either pick one at random or use domain knowledge to pick the most suitable subset and values as will be discussed in the Results Section.

4.7 Pseudo-code for the Multi-Modal Mean-Fields algorithm

Algorithm 1 summarises the operations to split one mode into two, or, in other words, to obtain the two additional constraints which are used to define the two newly created subsets. Algorithm 2 summarises the operations to obtain the Multi-Modal Mean Field Distribution by constructing the whole Tree.

In Algorithm 2, $ConstraintTree$, is taken to be a Tree in the form of a list of constraints, one for each branching-point, or leaf,—except for the root—, in a breadth first order. The function `pathto`($nNode$), returns the set of indices corresponding to the branching points on the path to the branching point, or leaf with index $nNode$, including index $nNode$ itself.

4.8 Results

We first use synthetic data to demonstrate that MMMF can approximate a multi-modal probability density function better than both standard MF and the recent approach of Weller and Domke [2015], which also relies on clamping to explore multiple modes. We then demonstrate that this translates to an actual performance gain for two real-world algorithms—one for people detection Fleuret et al. [2008] and the other for segmentation Chen et al. [2015], Yu and Koltun [2016]—both relying on a traditional Mean Field approach. We will make all our code and test datasets publicly available.

The parameters that control MMMF are the number of modes we use, the cardinality threshold C at each split, the ϵ value of Equation 4.9, the entropy thresholds h_{low} and h_{high} of Equation 4.39, and the temperature $T_{\max}$ introduced in Section 4.6. In all our experiments, we use $\epsilon = 10^{-4}$, $h_{low} = 0.3$, and $h_{high} = 0.7$. As discussed in Section 4.6, when the CRF is a dense Gaussian CRF, we can approximate and upper bound the critical temperature T_c in closed-form and we simply take $T_{\max}$ to be this upper bound to guarantee that $T_{\max} > T_c$. Otherwise, we choose $T_{\max}$ empirically on a small validation-set and fix it during testing.

Algorithm 1 Function:`Split`($ConstraintList$)

Input:

$E(\mathbf{x})$: An Energy function defined by a CRF;

`SolveMF`($E, ConstraintList$): A Mean Field solver with cardinality constraint.;

$Temperatures$: A list of temperatures in increasing order;

$\mathcal{H}_{low}, \mathcal{H}_{high}$: Entropy thresholds for the phase transition. 0.3 and 0.6 here.

C: A cardinality threshold

Output:

$LeftConstraints$: A triplet containing a list of variables, clamped to value, -C

$RightConstraints$: A triplet containing a list of variables, clamped to value, C

```
$Q^{T_0} \leftarrow$ SolveMF($E$)
for T in $Temperatures$ do
    $Q^T \leftarrow$ SolveMF($\frac{E}{T}$, $ConstraintList$)
    $i_{list} \leftarrow [.]$
    $v_{list} \leftarrow [.]$
    for index in 1...len($Q^t$), v in $labels$ do
        if $\mathbb{1}[\mathcal{H}(q^T_{index}) > 0.6]\mathbb{1}[\mathcal{H}(q^{T_0}_{index}) < 0.3]\mathbb{1}[q^{T_0}_{index,v} > 0.5] = 1$ then
            $i_{list}$.append(index),$v_{list}$.append(v)
        end if
    end for
    if len($i_{list}$) $> 0$ then
        exit for loop
    end if
end for
$LeftConstraints = i_{list}, v_{list}, -C$
$RightConstraints = i_{list}, v_{list}, C$
return $LeftConstraints$,$RightConstraints$
```

Algorithm 2 Compute Multi-Modal Mean Field

Input:

$E(\mathbf{x})$: An Energy function defined on a CRF;
`SolveMF`$(E, ConstraintList)$: A Mean Field solver with cardinality constraint;
`Split`$(ConstraintList)$: Alg. 1. A function that computes the new constraints.
$NModes$: A target for the number of modes in the Multi-Modal Mean Field

Output:

$Qlist$: A list of Mean Field distributions in the form of a table of marginals
$mlist$: A list of probabilities, one for each mode

```
$ConstraintTree = [.]$
We first build the tree by adding constraints.
while $nNode < NModes$ do
    $ConstraintList = [.]$
    for $p$ in pathto($nNode$) do
        $ConstraintList$.append(ConstraintTree[p])
    end for
    $LeftConstraints, RightConstraints \leftarrow$ Split($ConstraintList$)
    $ConstraintTree$.append($LeftConstraints$)
    $ConstraintTree$.append($RightConstraints$)
end while
We now turn to the computation of on MF distribution per leaf.
$Qlist = [.], Zlist = [.], mlist = [.]$
for mode in $0 \ldots NModes$ do
    $ConstraintList = [.]$
    for $p$ in pathto($mode + NModes - 1$) do
        $ConstraintList$.append(ConstraintTree[p])
    end for
    Q,Z $\leftarrow$ SolveMF(E,ConstraintList)
    $Qlist$.append($Q$)
    $Zlist$.append($Z$)
end for
Finally, we compute the mode probabilities.
for $mode$ in $0 \ldots NModes$ do
    $mlist$.append($\frac{Zlist[mode]}{\sum Zlist}$)
end for
return $Qlist$, $mlist$
```

4.8.1 Synthetic Data

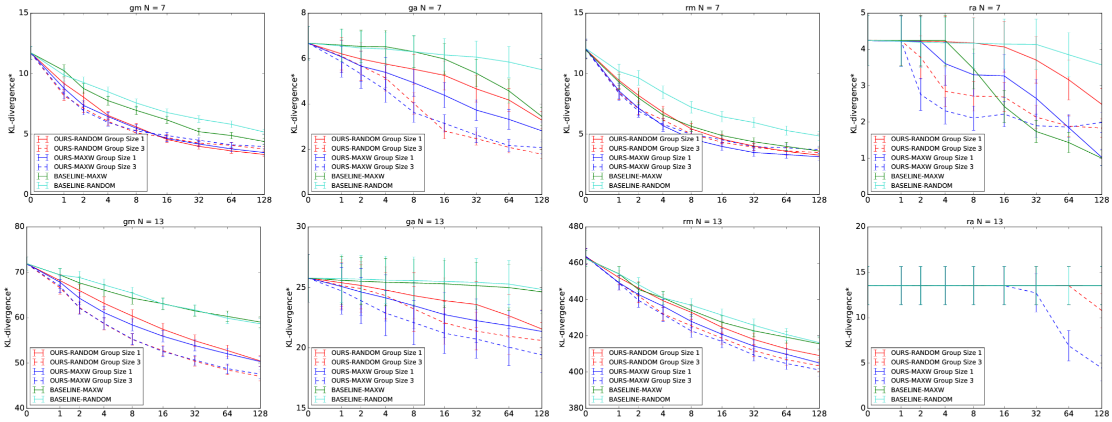
To demonstrate that our approach minimizes the KL-Divergence better than both standard MF and the clamping one of Weller and Domke [2015], we use the same experimental protocol to generate conditional random fields with random weights as in Eaton and Ghahrmani [2009], Weller and Jebara [2014], Weller and Domke [2015]. Our task is then to find the MMMF approximation with lowest KL-Divergence for any given number of nodes. When that number is one, it reduces to MF. Because it involves randomly chosen positive and negative weights, this problem effectively mimics difficult real-world ones with repulsive terms, uncontrolled loops, and strong correlations.

In Figure 4.5, we plot the KL-Divergence as a function of the number of modes used to approximate the distribution on the standard benchmarks. These modes are obtained using either our entropy-based criterion as described in Section 4.6, or the MaxW one of Weller and Domke [2015], which we will refer to as **BASELINE-MAXW**. It involves sequentially clamping the variable having the largest sum of absolute values of pairwise potentials for edges linking it to its neighbors. It was shown to be one of the best methods among several others, which all performed roughly similarly. In our experiments, we used the phase-transition criterion of Section 4.6 to select candidate variables to clamp. We then either randomly chose the group of L variables to clamp or used the MaxW criterion of Weller and Domke [2015] to select the best L variables. We will refer to the first as **OURS-RANDOM** and to the second as **OURS-MAXW**. Finally, in all cases, $C = L$ and the values v_u correspond to the ones taken by the MAP of the mode split.

In Figure 4.5, we plot the resulting curves for $L = 1$ and $L = 3$, evaluated on 100 instances. **OURS-RANDOM** performs better than the method **BASELINE-MAXW** in most cases, even though it does not use any knowledge of the CRF internals, and **OURS-MAXW**, which does, performs even better. The results on the 13×13 grid demonstrate the advantage of clamping variables by groups when the CRF gets larger.

4.8.2 Multi-modal Probabilistic Occupancy Maps

The Probabilistic Occupancy Map (POM) method Fleuret et al. [2008] relies on mean-field inference for pedestrian detection. More specifically, given several cameras with



Mixed grid　Attractive grid　Mixed random　Attractive random

Figure 4.5 – KL-divergence using either our clamping method or that of Weller and Domke [2015] averaged over 100 trials. The vertical bars represent standard deviations. **Attractive** means that pairwise terms are drawn uniformly from $[0,6]$ whereas **Repulsive** means drawn from $[-6,6]$. **Grid** indicates a grid topology for the CRF, whereas **Random** indicates that the connections are chosen randomly such that there are as many as in the grids. We ran our experiments with both 7×7 and 13×13 variables CRFs.

overlapping fields of view of a discretized ground plane, the algorithm first performs background subtraction. It then estimates the probabilities of occupancy at every discrete location as the marginals of a product law minimizing the KL divergence from the "true" conditional posterior distribution, formulated as in Equation 3.1 by defining an energy function. Its value is computed by using a generative model: It represents humans as simple cylinders projecting to rectangles in the various images. Given the probability of presence or absence of people at different locations and known camera models, this produces synthetic images whose proximity to the corresponding background subtraction images is measured and used to define the energy.

This algorithm is usually very robust but can fail when multiple interpretations of a background subtraction image are possible. This stems from the limited modeling power of the standard MF approximation. We show here that, in such cases, replacing MF by MMMF while retaining the rest of the framework yields multiple interpretations, among which the correct one is usually to be found.

Figure 4.6 depicts what happens when we replace MF by MMMF to approximate the true posterior, while changing nothing else to the algorithm. To generate new branches of the binary tree of Section 4.5, we find potential variables to clamp as described in Section 4.6. Among those, we clamp the one with the largest entropy gap—$\mathscr{H}(q_i^T) - \mathscr{H}(q_i^1)$, using the notations of Equation 4.39—and its neighbors on the grid. When evaluating our cardinality constraint, we take C to be 1, meaning that one branch of the tree corresponds to no one in the neighborhood of the selected location and the other to at least one person being present in this neighborhood. Since we typically create those locations by discretizing the ground plane into $10cm \times 10cm$ grid cells, this forces the two newly instantiated modes to be significantly different as opposed to featuring the same detection shifted by a few centimeters. In Figure 4.6, we plot the results as dotted curves representing the MODA scores as functions of the distance threshold used to compute them Bernardin and Stiefelhagen [2008]. In all cases, we used 4 modes for the MMMF approximation and followed the DivMBest evaluation metric Batra et al. [2012] to produce a score by selecting among the 4 detection maps corresponding to each mode the one yielding the highest MODA score. This produces red dotted MMMF curves that are systematically above the blue dotted MF.

However, to turn this improvement into a practical technique, we need a way to choose among the 4 possible interpretations without using the ground truth. We use temporal consistency to jointly find the best sequence of modes, and reconstruct trajectories from this sequence. In the original algorithm, the POMs computed at successive instants were used to produce consistent trajectories using the a K-Shortest Path (KSP) algorithm Berclaz et al. [2011]. This involves building a graph in which each ground location at each time step corresponds to a node and neighboring locations at consecutive time steps are connected. KSP then finds a set of node-disjoint shortest paths in this graph where the cost of going through a location is proportional to the negative log-probability of the location in the POM Suurballe [1974]. Since MMMF produces multiple POMs, we then solve a multiple shortest-path problem in this new graph, with the additional constraint that at each time step all the paths have to go through copies of the nodes corresponding to the same mode, as described in more details in the next section.

The solid blue lines in Figure 4.6 depict the MODA scores when using KSP and the red ones the multi-modal version, which we label as KSP*. The MMMF curves are again above the MF ones. This makes sense because ambiguous situations rarely persist for more than a few a frames. As a result, enforcing temporal consistency eliminates them.

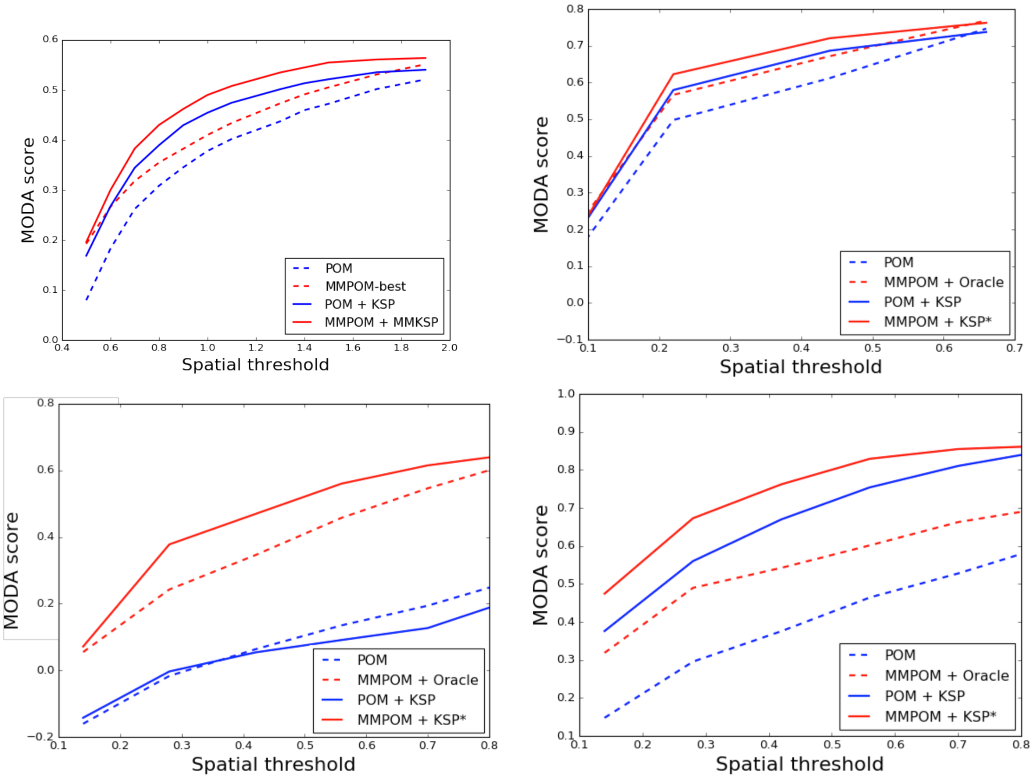
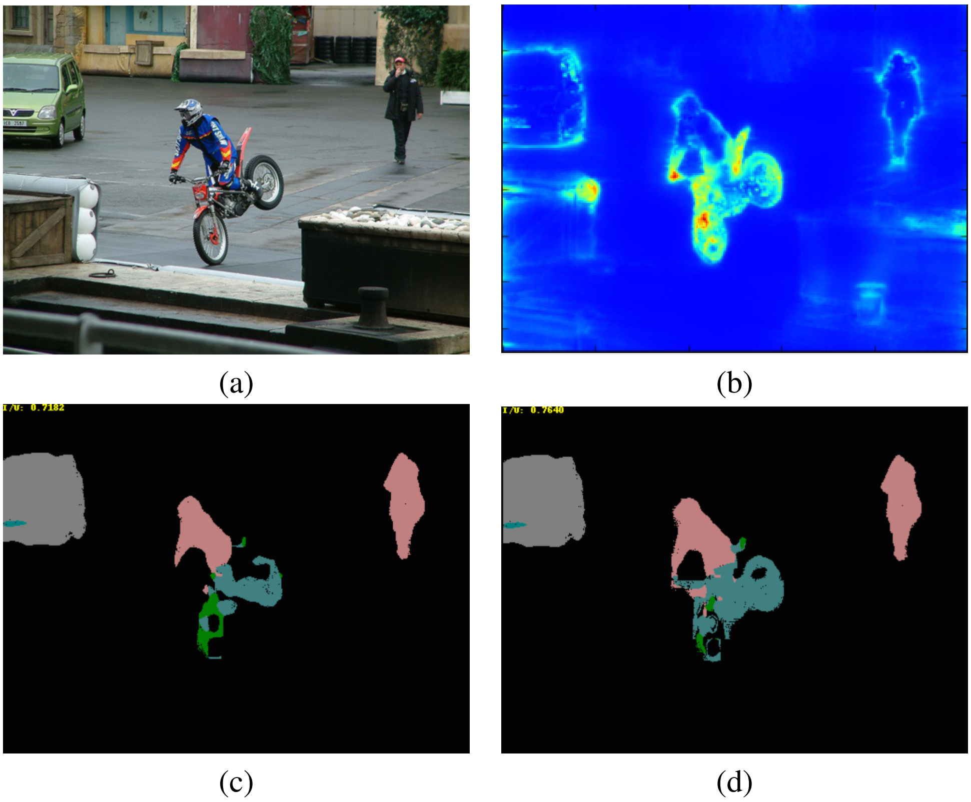


Figure 4.6 – Replacing MF by MMMF in the POM algorithm Fleuret et al. [2008]. The blue curves are MODA scores Bernardin and Stiefelhagen [2008] obtained using MF and the red ones scores using MMMF. They are shown as solid lines when temporal consistency was enforced and as dotted lines otherwise. Note that the red MMMF lines are above corresponding blue MF ones in all cases. (a) 1000 frames from the MVL5 Mandeljc et al. [2012] dataset using a single camera. (b) 400 frames from the Terrace dataset Berclaz et al. [2011] using two cameras. (c) 80 frames of the EPFL-Lab dataset Berclaz et al. [2011] using a single camera. (d) 80 frames from the EPFL-Lab dataset Berclaz et al. [2011] using two cameras.

4.8.3 Multi-Modal Semantic Segmentation

CRF-based semantic segmentation is one of best known application of MF inference in Computer Vision and many recent algorithms rely on dense CRF's Krähenbühl and Koltun [2013] for this purpose. We demonstrate here that our MMMF approximation can enhance the inference component of two such recent algorithms Chen et al. [2015], Yu and Koltun [2016] on the Pascal VOC 2012 segmentation dataset and the MPI video segmentation one Galasso et al. [2013].

Individual VOC Images We write the posterior in terms of the CRF of Chen et al. [2015], which we try to approximate. To create a branch of the binary tree of Section 4.5, we first find the potential variables to clamp as described in Section 4.6. As in 4.8.2, we select the ones in the sliding window with the largest entropy gap, $\mathscr{H}(q_i^T) - \mathscr{H}(q_i^1)$. We then take C to be $L/2$ when evaluating our cardinality constraint, meaning that we seek the dominant label among the selected variables and split the state space into those for which more than half these variables take this value and those in which less than half do.



(a) (b)

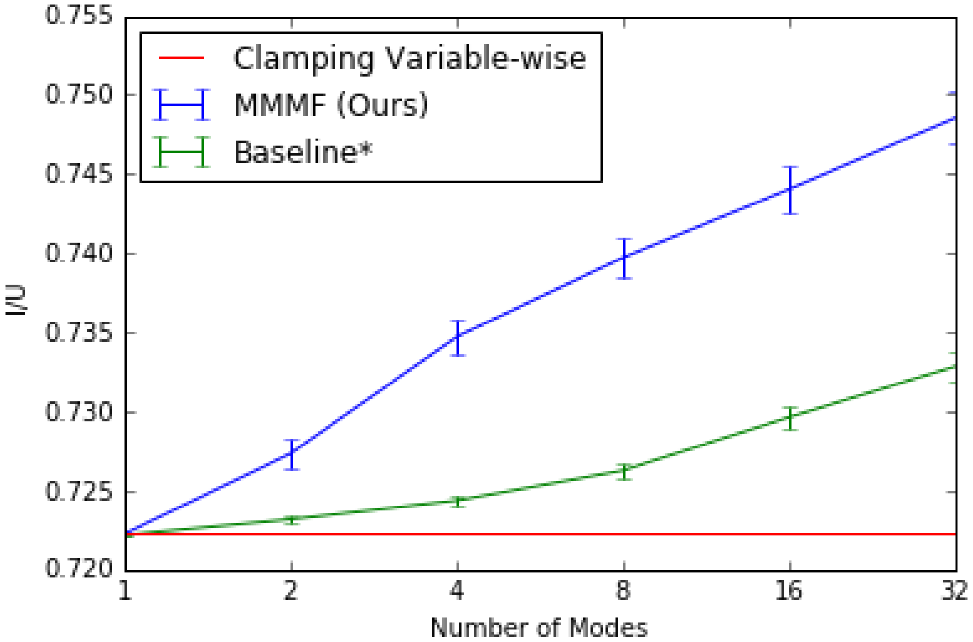
(c) (d)

Figure 4.7 – Qualitative semantic segmentation. (a) Original image. (b) Entropy gap. (c) Labels with maximum a Posteriori Probability after MF approximation. (d) Labels with maximum a Posteriori Probability for the best mode of the MMMF approximation.

Figure 4.7 illustrates the results on an image of the VOC dataset. To evaluate such results quantitatively, we first use the DivMBest metric Batra et al. [2012], as we did in Section 4.8.2. We assume we have an oracle that can select the best mode of our multi-modal approximation by looking at the ground truth. Figure 4.8 depicts the results on the validation set of the VOC 2012 Pascal dataset in terms of the average intersection over union (IU) score as a function of the number of modes. When only 1 mode is used, the result boils down to standard MF inference as in Chen et al. [2015]. Using

32 yields a 2.5% improvement over the MF approximation. This may seem small until one considers that we *only* modify the algorithm's inference engine and leave the unary terms unchanged. In Chen et al. [2015], Zheng et al. [2015], this engine has been shown to contribute approximately 3% to the overall performance, which means that we almost double its effectiveness. For analysis purposes, we implemented two baselines:

- Instead of clamping groups of variables, we only clamp the variable with the maximum entropy gap at each step. As depicted by the red curve in Figure 4.8, this has absolutely no effect and illustrates the importance of clamping groups of variable instead of single ones as in Weller and Domke [2015].
- The DivMBest approach Batra et al. [2012] first computes a MAP and then adds a penalty term to the energy function to find another MAP that is different from the first. It then repeats the process. We adapted this approach for MF inference. The green curve in Figure 4.8 depicts the result, which MMMF outperforms by 1.5%.



IU

score for the best mode

Figure 4.8 – Quantitative semantic segmentation on VOC 2012. IU score for best mode as a function of the number of modes. MMMF in blue, baselines in red and green.

Method	Mean IOU
MF	44.9%
Weller and Domke [2015] + Temp	44.9%
MMMF + Temporal	47.3%
MMMF-Best	53.2%

Table 4.1 – Quantitative semantic segmentation MPI dataset Galasso et al. [2013].

Semantic Video Segmentation. We ran the same experiment on the images of the MPI video segmentation dataset Galasso et al. [2013] using the CRF of Yu and Koltun [2016]. In this case, we can exploit temporal consistency to avoid having to use an oracle and nevertheless get an exploitable result, as we did in Section 4.8.2. Furthermore, we can do this in spite of the relatively low frame-rate of about 1Hz.

More specifically, we first define a compatibility measure between consecutive modes based on label probabilities of matching key-points, which we compute using a key-point matching algorithm Revaud et al. [2016]. We then compute a shortest path over the sequence of modes, taking into account individual mode probabilities given by Equation **??**. Finally, we use only the MAP corresponding to the mode chosen by the shortest path algorithm to produce the segmentation. In Figure 4.1, we again report the results in terms of IU score. This time the improvement is around 2.4%, which indicates that imposing temporal consistency very substantially improves the quality of the inference. To the best of our knowledge, other state of the art video semantic segmentation methods are not applicable for such image sequences. Hur and Roth [2016] requires non-moving scenes and a super-pixel decomposition, which prevents using all the dense CRF-based image segmentors. Kundu et al. [2016] was only applied to street scenes and requires a much higher frame rate to provide an accurate flow estimation.

4.9 K-Shortest Path algorithm for the Multi-Modal Probabilistic Occupancy Maps

We present here the algorithm we use to reconstruct tracks from the Multi-Modal Probabilistic Occupancy Maps (MMPOMs) of Section 4.8

KSP In the original algorithm of Berclaz et al. [2011], the POMs computed at successive instants were used to produce consistent trajectories using the a K-Shortest Path (KSP) algorithm Suurballe [1974]. This involves building a graph in which each ground location at each time step corresponds to a node and neighboring locations at consecutive time steps are connected. KSP then finds a set of node-disjoint shortest paths in this graph where the cost of going through a location is proportional to the negative log-probability of the location in the POM Berclaz et al. [2011]. The KSP problem can be solved in linear time and an efficient implementation is available online.

KSP for Multi-Modal POM Since MMMF produces multiple POMs, one for each mode, at each time-step, we duplicate the KSP graph nodes, once for each mode as well. Each node is then connected to each copy of neighboring locations from previous and following time steps. We then solve a multiple shortest-path problem in this new graph, with the additional constraint that at each time step all the paths have to go through copies of the nodes corresponding to the same mode. This larger problem is NP-Hard and cannot be solved by a polynomial algorithm such as KSP. We therefore use the Gurobi Mixed-Integer Linear Program solver Gurobi Optimization [2016].

More precisely, let us assume that we have a sequence of Multi-Modal POMs Q_k^t and mode probabilities m_k^t for $t \in \{1, \ldots, T\}$ representing time-steps and $k \in \{1, \ldots, K\}$ representing different modes. Each Q_k^t is materialized through a vector of probabilities of presence $q_{k,i}^t$, where each $i \leq N$ is indexes a location on the tracking grid.

Using the grid topology, we define a neighborhood around each variable, which corresponds to the maximal distance a walking person can make on a grid in one time step. Let us denote by $\mathcal{N}_i$ the set of indices corresponding to locations in the neighbourhood of i. The topology is fixed and hence $\mathcal{N}_i$ does not depend on the time steps. We define the following log-likelihood costs.

Using a Log-Likelihood penalty, we define the following costs:

- $C_{k,i}^t = \log\left(\frac{1 - q_{k,i}^t}{q_{k,i}^t}\right)$, representing the cost of going through variable i at time t if mode k is chosen.

- $C_k^t = \log\left(\frac{1-m_k^t}{m_k^t}\right)$, representing the cost of choosing mode k at time t.

We solve for an optimization problem involving the following variables:

- $x_{k,i,l,j}^t$ is a binary flow variable that should be 1 if a person was located in i at t and moved to j at $t+1$, while modes k and l were respectively chosen at time t and $t+1$.
- y_k^t is a binary variable that indicates whether mode k is selected at time t.

We can then rewrite the Multi-Modal K-Shortest Path problem as the following program, were we always assume that $t \leq T$ stands for a time step, $k \leq K$ and $l \leq K$ stand for mode indices, and $i \leq N$ and $j \leq N$ stand for grid locations:

$$
\begin{array}{lll}
\min & \sum\limits_{t,k} C_k^t y_k^t + \sum\limits_{t,k,l\leq K} \sum\limits_{i,j\in\mathcal{N}_i} C_{k,i}^t x_{k,i,l,j}^t & \\
\text{s.t.} & \forall (t,k,i)\,,\ \sum\limits_{l,j\in\mathcal{N}_i} x_{l,j,k,i}^{t-1} = \sum\limits_{l,j\in\mathcal{N}_i} x_{k,i,l,j}^t & \texttt{flow conservation} \\
 & \forall (t,k,i)\,,\ \sum\limits_{l,j\in\mathcal{N}_i} x_{k,i,l,j}^t \leq y_k^t & \texttt{disjoint paths + selected mode} \\
 & \forall t\,,\ \sum\limits_{k} y_k^t = 1 & \texttt{selecting one mode} \\
 & \forall t,k,i,l,j\,,\ 0 \leq x_{k,i,l,j}^t \leq 1 & \\
 & \forall t,k\,,\ y_k^t \in \{0,1\} &
\end{array}
\tag{4.40}
$$

KSP prunning However, the problem as written above, may involve several tens millions of flow variables and therefore becomes intractable, even for the best MILP solvers. We therefore first prune the graph to drastically reduce its size.

The obvious strategy would be by thresholding the POMs and removing all the outgoing and incoming edges from locations which have probabilities below q_{thresh}. However,

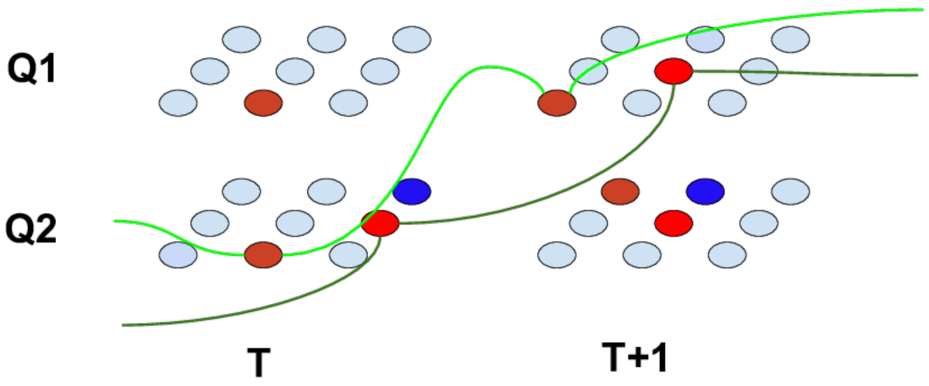


Figure 4.9 – Illustration of the output of our K-Shortest Path algorithm in the case of multiple modes.

this would be self-defeating as one of the main strengths of the KSP formulation is to be very robust to missing-detections and be able to reconstruct a track even if a detection is completely lost for several frames.

We therefore resort to a different strategy. More precisely, we initially relax the constraint `disjoint paths + selected mode`, to a simple disjoint path constraint, and remove the constraint `selecting one mode`. We therefore obtain a relaxed problem

$$
\begin{aligned}
\min \quad & \sum_{t,k} \sum_{t,k,l \le K} \sum_{i,j \in \mathcal{N}_i} C^t_{k,i} x^t_{k,i,l,j} & & \\
\text{s.t.} \quad & \forall (t,k,i)\,, \sum_{l,j \in \mathcal{N}_i} x^{t-1}_{l,j,k,i} = \sum_{l,j \in \mathcal{N}_i} x^t_{k,i,l,j} & & \texttt{flow conservation} \\
& \forall (t,k,i)\,, \sum_{l,j \in \mathcal{N}_i} x^t_{k,i,l,j} \le 1 & & \texttt{disjoint paths} \\
& \forall t,k,i,l,j\,,\ 0 \le x^t_{k,i,l,j} \le 1 & &
\end{aligned}
\tag{4.41}
$$

which is nothing but a vanilla K-Shortest Path Problem. It can be solved using our linear-time KSP algorithm. This KSP problem will output a very large number of paths, going through all the different modes simultaneously. From, this output, we extract the set of grid locations which are used, in any mode, at each time step, and select them as our potential locations in the final program. In our current implementation, we add to these locations, the ones for which $q^t_{k,i} \geq q_{thresh}$ for any mode at time-step t.

We can finally solve Program 4.40, where non-selected locations are pruned from the flow graph. We don't know if our strategy, based on a relaxation and pruning, provides a guaranteed optimal solution to 4.40, but this is an interesting question.

4.10 Conclusion

We have shown that our MMMF aproach makes it possible to add structure to the standard MF approximation of CRFs and to increase the performance of algorithms that depend on it. In effect, our algorithm creates several alternative MF approximations with probabilities assigned to them, which effectively models complex situations in which more than one interpretation is possible.

In future chapters, we will see how MMMF can be integrated into structured learning architectures through the Back Mean-Field procedure.

5 Improved parameters learning

5.1 Practical Challenges of Learning through Mean-Fields

We have been focusing so far on inference problems, assuming that the parameters of the model were fixed. In many problems this assumption is reasonable, and the parameters can be set manually to express prior knowledge or geometric properties of the predictions, such as continuity or local smoothness.

However, these parameters cannot always be chosen manually and it is crucial to be able to learn them from data. The learned parameters can range from a relatively small number of values encoding potentials at a low level, to a higher level parameterization, such as the weights of a deep neural net.

In previous works [Arnab et al., 2018], the mean-field algorithm has often been used for parameters learning using a very pragmatic method. The back mean-field algorithm optimizes the weights directly to make the approximate variational distribution, computed by the MF inference algorithm, fit the observed data. This method, despite its popularity, is often limited in practice and suffers from the existence of many modes in the posterior.

We therefore derive a new learning method, that leverages on the Multi-Modal Mean-Field approach of Chapter 4. We show how this learning method bridges the two previous classes of approach to parameters learning, described in section 2.4. More precisely, we propose a flexible new learning framework, which creates a continuum between contrastive divergence and back mean-field based techniques. We call this new method

Multi-Modal Back Mean-Field.

However, the learning precision brought by this new approach sometimes comes at the cost of an increased computational complexity. In order to alleviate this issue, we finally derive a simple but efficient approximation to the Multi-Modal Mean-Field Learning approach, which, as the back mean-field one, can be seamlessly integrated in Deep-Learning pipelines, but brings substantial improvements to structured learning tasks.

5.2 Related Work

Lets us assume that we observe D data-points $\{(\mathbf{x}^1, \mathbf{I}^1), \ldots, (\mathbf{x}^D, \mathbf{I}^D)\} = \mathcal{D}$. The task of learning the CRF parameters can be expressed as the maximization of the log-likelihood of the observed data under the CRF model parametrized by θ,

$$\begin{aligned} \theta^* &= \max_{\theta} \sum_{(\mathbf{x},\mathbf{I}) \in \mathcal{D}} \log(P(\mathbf{X} = \mathbf{x} \mid \mathbf{I}; \theta)) && (5.1) \\ &= \max_{\theta} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} E(\mathbf{x} \mid \mathbf{I}; \theta) - \log(Z_\theta)\,. && (5.2) \end{aligned}$$

As discussed in Chapter 2 one of the ways explored in the literature to solve this task is to approximate Z_θ via a variational bound, sampling, or other approximate inference methods. Nevertheless, sampling can be very slow and other inference methods intractable in many cases. Variational Inference techniques, such as the mean-field one, can be used to approximate the partition function.

The methods presented above use an explicit energy model, which is only related to the mean-field algorithm through the energy function parameters which defines the KL-Divergence. However, because of the highly non-convex nature of the KL-Divergence function, and the instability of the MF algorithm with respect to the initialization and step size, the situation is slightly more complex. Indeed, a "good" energy function – such that the corresponding posteriors fit the observed data –, is not always going to be easily optimizable at inference time. Therefore, the output of the inference algorithm might not be a good approximation of the posterior itself.

Therefore, Domke [2013] proposed a more direct approach. Namely, instead of maximizing the log-likelihood of the data under the true posterior distribution, the back mean-field maximizes the log-likelihood under the mean-field approximation,

$$\theta^* = \max_{\theta} \sum_{(\mathbf{x},\mathbf{I})\in\mathscr{D}} \log(Q(\mathbf{X}=\mathbf{x} \mid \mathbf{I};\theta)) \tag{5.3}$$

$$= \max_{\theta} \sum_{(\mathbf{x},\mathbf{I})\in\mathscr{D}} \sum_{i} \log Q_i(\mathbf{X}_i = \mathbf{x}_i \mid \mathbf{I};\theta)\,, \tag{5.4}$$

where $Q(\mathbf{X}=\mathbf{x} \mid \mathbf{I};\theta)$ depends on the parameters θ through the optimization problem

$$Q_\theta = \operatorname*{argmin}_{Q\in\mathscr{Q}} \mathrm{KL}(Q\|P_\theta)\,. \tag{5.5}$$

Note that, to compute Q_θ from the CRF potential functions ϕ, we use the gradient descent method described in Chapter 3.2.2. By unrolling those iterations, we obtain a differentiable mapping from θ to Q_θ. Therefore, assuming that the potential function $\phi(. \mid \mathbf{I},\theta)$ is a differentiable parametrization, θ can be optimized via stochastic gradient descent.

In practice, it means that mean-field iterations can be "unrolled" at the end of a Neural Network architecture and gradient information can be Back-Propagated through it.

Unrolling iterations as a part of the Neural Network requires to store the activations during the forward pass – values of Q^t at each iteration in this context –, in order to compute the Backward pass. This is memory inefficient. Furthermore, in practice, it turns out that the gradient vanishes very quickly over the backward computation, because of the normalization of the MF iterations, which act like sigmoid transfer functions. Therefore, in practice, we can restrict the back-propagation to the last few iterations, without loss of efficiency.

Despite some practical successes, this method suffers from the multi-modality of the MF objective function. Intuitively, even if the learned distribution P_θ is a good model for the empirical data, the MF distribution Q_θ, computed at inference time may concentrate on a single mode of P_θ. Therefore, if the drawn sample $(\mathbf{x}_s,\mathbf{I}_s)$, belongs to another mode which is not modeled by Q, the sample learning loss will be high, resulting in an update on θ, even though P_θ is a good model. We will see how this can be remedied using

Multi-Modal Back Mean-Field.

5.3 Method

5.3.1 Multi-Modal Back Mean-Field

In this section, the Multi-Modal Mean-Field method presented in Chapter 4, is leveraged to improve parameters learning in CRFs. We call this novel procedure *Multi-Modal Back-Mean-Field*, and, similarly to the case of the MF approximation, its goal is to maximize the log-likelihood of the observations under the approximation of the posterior, which is a MMMF distribution in this case.

Intuitively, one of the main weaknesses of the standard back-MF learning method comes from the problem of multi-modality. More precisely, if the MF approximation converged to a mode which is very different from the one the ground truth sample is actually belonging to, the gradients computed will be wrong and might mislead the parameters update. By using the tree structure we augment the chances to model correctly the mode the sample belongs to, and therefore get more reliable updates. Furthermore, by weighting mode probabilities according to the likelihood of samples belonging to it, we can strengthen the parameters learning.

As above, lets us assume that we observe D data-points $\{(\mathbf{x}^1,\mathbf{I}^1),\ldots,(\mathbf{x}^D,\mathbf{I}^D)\} = \mathscr{D}$. Our goal is therefore to find the CRF parameters θ^* which maximizes the data likelihood under $Q_{MM}(.\,|\,\mathbf{I},\theta)$.

Our main observation is that each data-point $\mathbf{x}$ actually belongs to a single element of the state space partition $\{\mathscr{X}_1,\ldots,\mathscr{X}_K\}$ introduced in section 4.4. Therefore, for every data-point, it brings a drastic simplification of the log-likelihood formula

$$\theta^* = \max_{\theta} \sum_{(\mathbf{x},\mathbf{I})\in\mathcal{D}} \log(Q_{MM}(\mathbf{X}=\mathbf{x}\mid\mathbf{I},\theta)) \tag{5.6}$$

$$= \max_{\theta} \sum_{(\mathbf{x},\mathbf{I})\in\mathcal{D}} \log\left(\sum_{k} m_k Q_k(\mathbf{X}=\mathbf{x}\mid\mathbf{I},\theta)\right) \tag{5.7}$$

$$= \max_{\theta} \sum_{(\mathbf{x},\mathbf{I})\in\mathcal{D}} \left[\log m_{k(\mathbf{x})}(\mathbf{I},\theta) + \log Q_{k(\mathbf{x})}(\mathbf{X}=\mathbf{x}\mid\mathbf{I},\theta)\right], \tag{5.8}$$

where m_k is defined in Equation 4.8 and $k(\mathbf{x})$ is the index k of the partition function such that $\mathbf{x}\in\mathcal{X}_k$.

Each one of the two terms of the sum in Equation 5.8 is computed as a differentiable function of the CRF parameters θ and the objective is therefore optimized via stochastic gradient descent, as in the standard back mean-field case.

The first term of Equation 5.8, weights the probability of the mode the ground-truth belongs to, compared to other modes. It is therefore similar in spirit to the approximations used in the standard contrastive divergence-based CRF training methods presented in section 2.4. We will even see below that, in some cases, they are exactly equivalent.

On the other hand, the second term of Equation 5.8, stems from a direct training method where we optimize the weights of the CRF to make the clamped MF approximation fit the data on the non-clamped terms. We will even see below that the standard back mean-field can be seen as a special case of or method.

5.3.2 Connection to other CRF training methods

. We explain below how multiple standard CRF training methods can be recovered as specific-cases of ours.

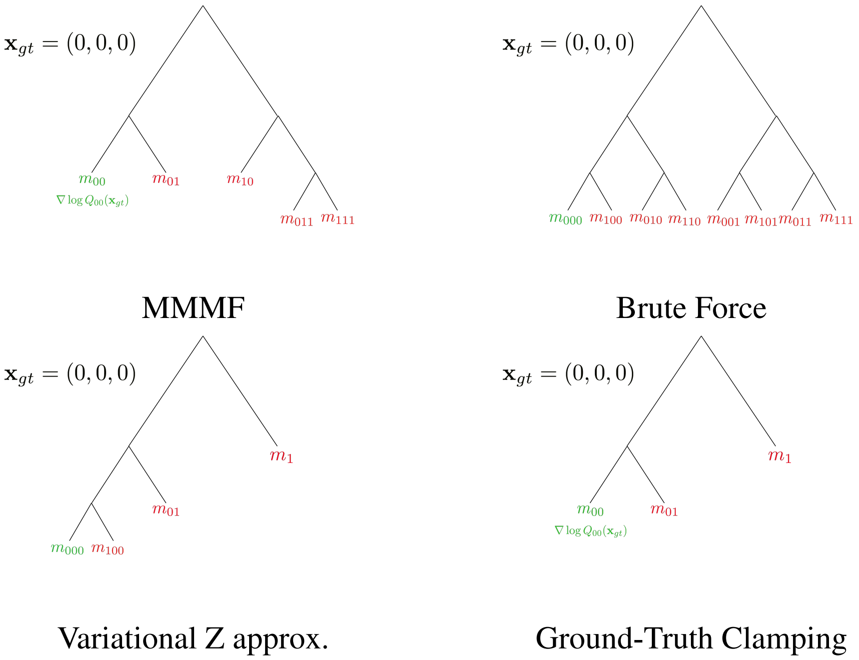
i **Back mean-field:** It is trivial to see that the standard Back Mean-Field method corresponds to ours where only a single mode is explored.

ii **Variational log-partition function approximation and contrastive divergence**: A popular way of training conditional random fields is to use a variational approximation of the log-partition function of Equation 5.2. This approximation makes the gradient computation tractable and one can proceed with optimization. Actually, it turns out that the specific configuration of our method where the branch of the clamping tree containing the ground truth element is completely explored and no other branch is, corresponds to this scenario.

No gradient information would then be propagated through the second term of Equation 5.8. The first term is then computed as the difference between the energy of the probability of the ground truth element and a mean-field approximation of the log-partition function.

Interestingly, variants of our algorithms include exploring branches starting closer to the ground truth sample. This technique is a variant of the *contrastive divergence* one, where the ground truth sample is used to guide the sampling of P_θ.

iii Brute force: On the opposite, the brute force training method can be achieved in our framework by exploring completely every branch. This will lead to an exact computation of Z_θ in Equation 5.2 and therefore an exact log-likelihood gradient.


$\mathbf{x}_{gt} = (0,0,0)$
m_{00}
$\nabla \log Q_{00}(\mathbf{x}_{gt})$
m_{01}
m_{10}
m_{011}
m_{111}
$\mathbf{x}_{gt} = (0,0,0)$
m_{000}
m_{100}
m_{010}
m_{110}
m_{001}
m_{101}
m_{011}
m_{111}
$\mathbf{x}_{gt} = (0,0,0)$
m_1
m_{01}
m_{000}
m_{100}
$\mathbf{x}_{gt} = (0,0,0)$
m_1
m_{00}
m_{01}
$\nabla \log Q_{00}(\mathbf{x}_{gt})$

MMMF

Brute Force

Variational Z approx.

Ground-Truth Clamping

Figure 5.1 – Illustration of the different versions of Multi-Modal Back Mean-Field and connections to other methods. The ground truth label $\mathbf{x}_{gt}$, is (000).

5.4 A new end-to-end training method for CRF-CNN

In this section, we present a new parameters learning method for CRFs, which is based on the Multi-Modal Back Mean-Field and is as efficient as the Back Mean-Field algorithm.

5.4.1 Efficient Multi-Modal training for CRF

The Multi-Modal Mean-Field presented in Chapter 4, provides a good approximation of the posterior at inference time. However, the sophisticated heuristics used to compute this approximation, make it sometimes difficult to use in practice for learning. This is especially true when the Conditional Random Field is combined with a Neural Network which parametrizes its potentials. Then, learning through gradient descent on the objective of Equation 5.8, can become very expensive as a new MMMF approximation must be computed for each gradient step. Even though it is superior in theory, it is not as efficient and easy to use as the standard back-MF one of Domke [2013].

However, note that Equation 5.8, is composed of two terms. Those two terms are not supposed to compensate each other and play different roles toward the same goal. The first one reweights the modes to make the one containing the data more likely, while the second one makes the MF approximation fit the data inside the mode. In theory, at convergence, both terms should have a gradient which is null.

Therefore, in practice, we can choose to ignore the first term and focus on the other. It means that we only train the CRF potentials such that the clamped-MF approximation, where some variables have been clamped to ground-truth, fits the data well. The problem could be that, when not clamped, the MF approximation converges to completely wrong modes, but this does not prevent our method to work well in practice.

At inference time, the MF iterations which were trained with clamping, will naturally converge to a mode of the true posterior, unlike those trained via standard back-MF. Effectively, for Multi-Modal Posterior distributions, the results will be more realistic and sampling from the obtained Mean-Field distributions, will produce more realistic looking results.

5.4.2 A generic structured learning framework for CNNs

In previous works, mean-field iterations were unrolled at the end of a Neural Network in order to refine the output. We propose a new generic Neural Network layer, based on the clamped mean-field algorithm presented above. Our new layer has the same advantage as previous unrolled mean-field approaches, namely, it introduces adaptive filtering in the Neural Network pipeline. Besides, our new generic layer has a better capacity to predict structured outputs directly.

As depicted in Figure 5.2, we use a first Neural Network to produce a set of unary potentials and a second one to produce pairwise ones. The pairwise potentials can take multiple forms and multiple parametrizations. By default, we assume that each vertex is connected to a fixed number of neighbors and that the CNN produces a vector of pairwise terms. More sophisticated adaptive strategies such as in Krähenbühl and Koltun [2011] or Simonovsky and Komodakis [2017], can be used.

For each training sample, we draw a fixed clamping mask with a fixed rate. It corresponds to a set of indices $\mathscr{C}$ in $\{1,\ldots,N\}$, where N is the number of vertices in the CRF. In practice, we observed that clamping 20% of the variables works well. We then unroll the MF iterations as is usually done, except that, after each iteration, the vertices selected by the clamping mask, are fixed to ground truth.

5.5 Experimental Evaluation

5.5.1 Learning People Detection with Multi-Modal Mean-Fields

We now show how we can use the training methods presented above to improve the results of our CRF Mean-Field based Multi-Camera detection algorithm of Baqué et al. [2017b], presented in more details in the last chapter of this thesis.

The core motivation behind our original approach is to properly handle occlusions, while still leveraging the power of CNNs. To do so, we model the interactions between multiple people who occlude each other but may not be physically close to each other. Our solution is to introduce an *observation space*; a generative model for observations

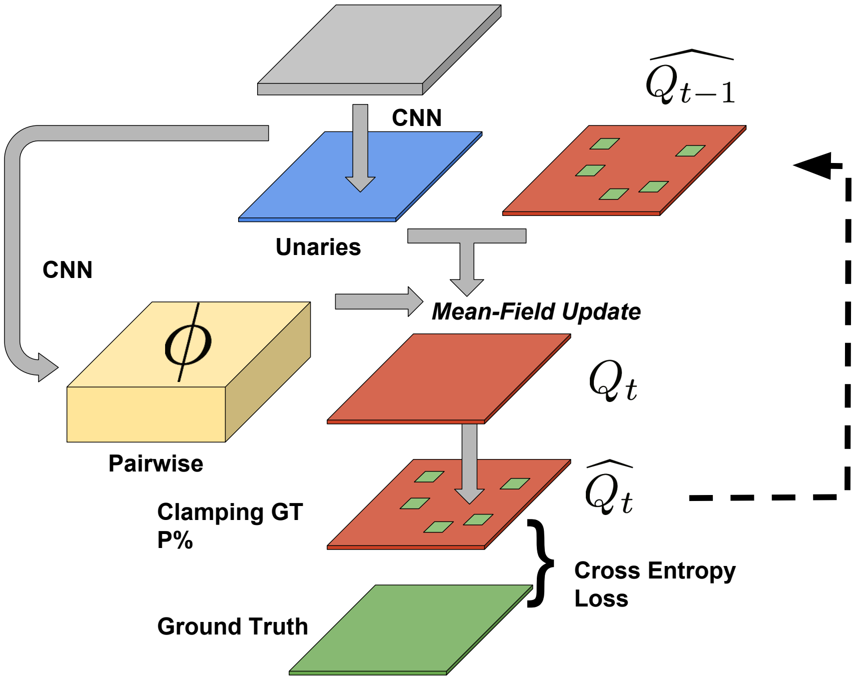


Figure 5.2 – A clamped MF iteration in a Convolutional Neural Network.

given where people are located in the ground plane; and a discriminative model that predicts expected observations from the images. We then define a loss function that measures how different the CNN predictions are from those generated by the model. Finally, we use a Mean-Field approach with respect to probabilities of presence in the ground plane to minimize this loss. We cast this computation in terms of minimizing the energy of a Conditional Random Field in which the interactions between nodes are non-local because the people who occlude each other may not be physically close, which requires long range high-order terms.

As in our previous paper [Baqué et al., 2017b], we assume that we observe D data point $(\mathbf{X}^0,\mathbf{I}^0),\ldots,(\mathbf{X}^D,\mathbf{I}^D)$ at training time, where $\mathbf{I}^d$ represents a multi-view image and $\mathbf{X}^d$ the corresponding ground truth presences. The purpose of training is then to optimize the CRF parameters θ to maximize $\sum\limits_{d\leq D} \log P(\mathbf{X}^d;\mathbf{I}^d)$.

As discussed in section 5.3, this can be tackled by end-to-end CRF learning techniques. In Baqué et al. [2017b], the back mean-field method is used. It improves slightly the results but at the price of a careful initialization, and careful tuning of the MF step sizes, mean-field temperature and learning rate.

Here, we compare this method to the ground truth clamping of section 5.4, which is inspired by the Multi-Modal Back Mean-Field. We observe that it is much more versatile, robust and easier to train compared to the back mean-field one which, barely works.

As in [Baqué et al., 2017b], we consider our **Wildtrack Dataset** of Chavdarova et al. [2018] to which we contributed. In this dataset, we provide a large-scale HD multi-camera pedestrian dataset. The seven-static-camera set-up captures realistic and challenging scenarios of walking people. Notably, its camera calibration, with joint high-precision projections, widens the range of algorithms which may make use of this dataset. It aims to help accelerate the research on automatic camera calibration, such annotations also accompany this dataset.

For evaluation, we use the **Multiple Object Detection Accuracy (MODA)** metric [Kasturi et al., 2009] which we will plot as a function of the distance parameter r, which measures the distance maximal from the ground truth for a detection to be considered as valid.

Fig. 5.3 shows the MODA score for the test set for several training methods and the dependency in r.

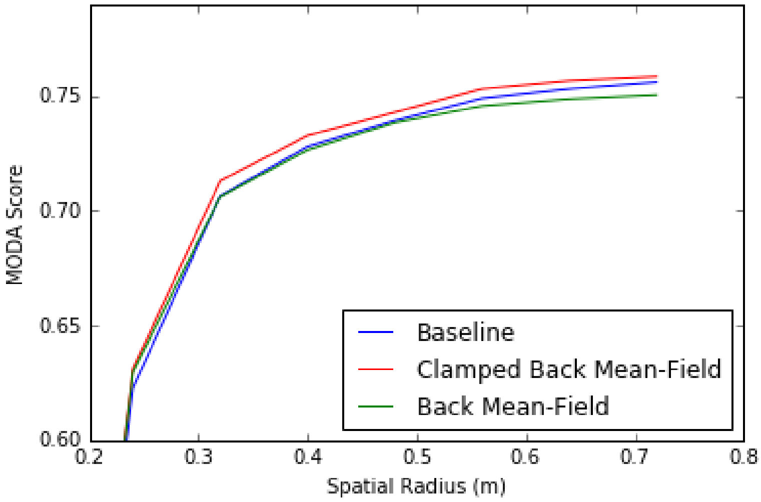


Figure 5.3 – MODA score after or before end-to-end fine-tuning of the CRF

5.5.2 Learning Surface Reconstruction with Multi-Modal Mean-Fields

We now apply our generic structured learning CNN to a monocular surface reconstruction task. More precisely, from a single RGB image of a blank slate which is deformed under the effect of the wind, our goal is to reconstruct its 3D shape. The shape is parametrized by the coordinates of the vertices of a mesh which is initialized at regular intervals on the original slate before deformation. Because the reconstructed surface is completely blank, without structure, its shape is inherently ambiguous, or, in other terms, multi-modal.

Architecture Similarly to the approach used in previous works in the domain of CNN-based surface reconstruction, we use a Convolutional Neural Network inspired by the architecture of the ResNet-51 one [He et al., 2016], followed by three fully-connected layers to predict a vector of size $363 = 3 \times 11 \times 11$, corresponding to the three coordinates of the displacement of control points on the mesh.

Whereas in standard benchmark implementations, the predictions would directly be taken to be the output of the network, we add several clamped MF layers as the one depicted in Figure 5.2. Instead of using a Cross-Entropy loss, we use a standard *L2-Distance*. The pairwise potentials take the form of a 4-connected grid – meaning that each vertex is connected only to its neighbors – and are predicted by a network that shares weights with the ResNet-51 used for the unaries, but has also three separate fully-connected layers.

Note that, in our implementation, the unary potentials are pretrained using the benchmark architecture described above.

Results We compare our method to the benchmark one, where only a standard neural network is used and to one where we train the same architecture as ours – with mean-field layers –, but without using the ground-truth clamping approach.

Since that the problem we are dealing with is ambiguous and that several reconstructions are possible for a given RGB image, a perfectly chosen and trained Neural Network could, in theory, minimize the vertex-wise Root-Mean Square Error (RMSE), while not caring about the structure of the problem. Nevertheless, we expect our approach to provide more naturally looking reconstructions and more reconstruction with small RMSE, while potentially making bigger mistakes on some other, at the benefit of inter-vertices coherence.

As explained above, there is no fundamental reason why our model should be better than the standard approach with respect to the RMSE loss. Nevertheless, and quite strikingly, our approach outperforms it, as shown in table 5.4.

Method	**Test RMSE**
Baseline (CNN)	17.6 %
Back-MF	15.9 %
Clamped Back-MF	15.6 %

Figure 5.4 – Regression results.

In order to get a more fine grained understanding of the benefits brought by our method, and see if it really improves the results in the way it is meant to, we plot the cumulative histogram of RMSE errors in Figure 5.5. It clearly shows that the Clamped-MF approach outperforms the standard CNN one. Interestingly, and as expected, we can see on the top right of the plot that, for very large errors, the standard CNN makes slightly smaller mistakes than ours.

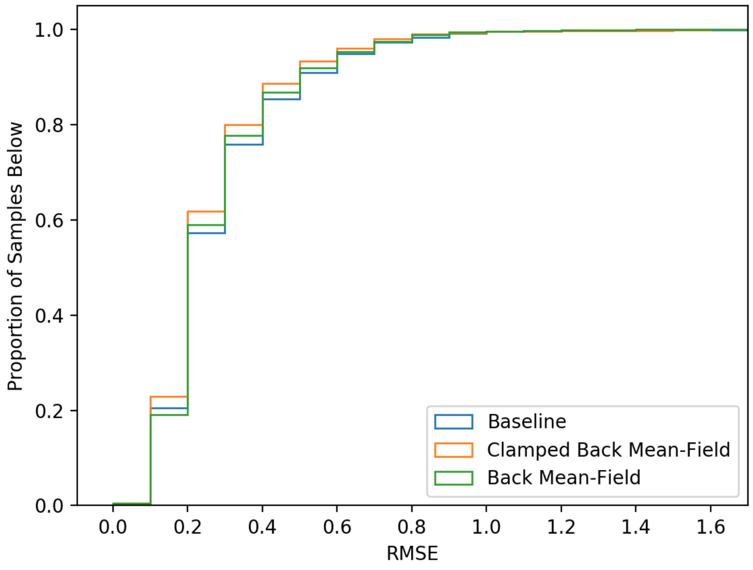


Figure 5.5 – Cumulative histogram of achieved RMSE, in percentage of the dataset.

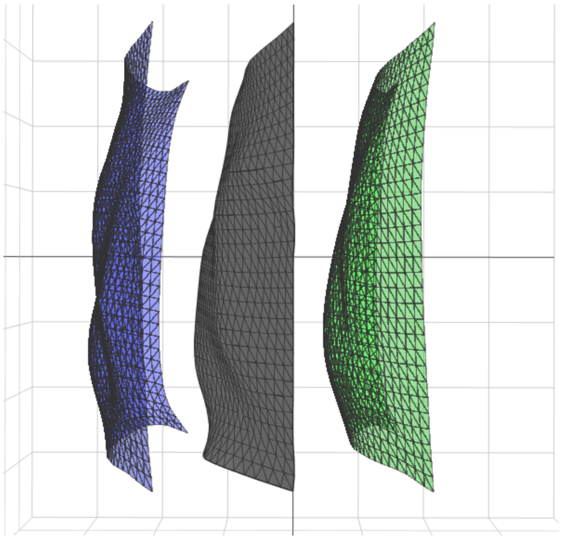


Figure 5.6 – Surface reconstruction. Left: **Baseline**. Center: **Ground-Truth**. Right: **Clamped Back MF**.

Finally, we provide a qualitative comparison of the obtained results in Figure 5.6, which shows that our approach makes more naturally looking reconstructions and, unlike benchmark approaches, does not tend to make over-smooth reconstructions.

5.6 Chapter Conclusion

We presented a new structured learning method based on a conjunction between Back Mean-Field and Multi-Modal Mean-Field. We showed how other standard CRF training methods can be interpreted as specific instantiation of this new Multi-Modal Back Mean-Field approach.

We proposed a simplification of the model, which makes it seamlessly integrable in standard CNN architectures, thus proposing a dedicated, principled structured learning architecture, at no additional computational cost.

Future works should explore more practical applications of our approach for a wide range of structured learning problems.

6 Application: Multi-Camera People Detection

6.1 Introduction

Multi-Camera Multi-Target Tracking (MCMT) algorithms have long been effective at tracking people in complex environments. Before the emergence of Deep Learning, some of the most effective methods relied on simple background subtraction, geometric and sparsity constraints, and occlusion reasoning [Fleuret et al., 2008, Berclaz et al., 2011, Alahi et al., 2011]. Given the limited discriminative power of background subtraction, they work surprisingly well as long as there are not too many people in the scene. However, their performance degrades as people density increases, making the background subtraction used as input less and less informative.

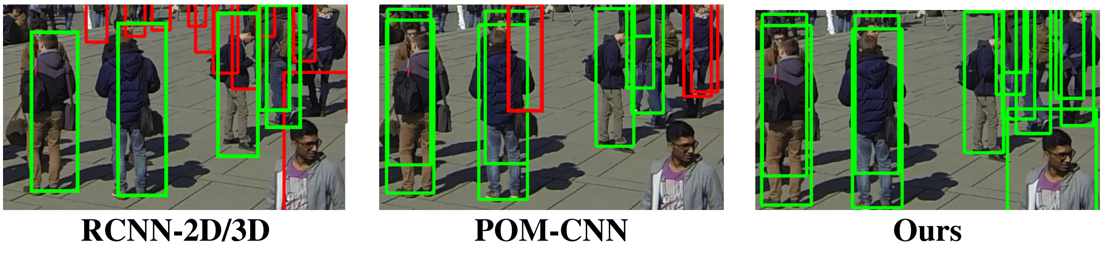


Figure 6.1 – Multi-camera detection in a crowded scene. Even though there are 7 cameras with overlapping fields of view, baselines inspired by earlier approaches—- **RCNN-2D/3D** by Xu et al. [2016] and **POM-CNN** by Fleuret et al. [2008], as described in Section 6.7.2—both generate false positives denoted by red rectangles and miss or misplace a number of people, whereas ours does not. This example is representative of the algorithm's behavior and is best viewed in color.

Since then, Deep Learning based people detection algorithms in single images [Ren et al., 2015] have become among the most effective. However, their power has only rarely been leveraged for MCMT purposes. Some recent algorithms, such as the one of Xu et al. [2016], attempt to do so by first detecting people in single images, projecting the detections into a common reference-frame, and finally putting them into correspondence to achieve 3D localization and eliminate false positives. As shown in Fig. 6.1, this is prone to errors for two reasons. First, projection in the reference frame is inaccurate, especially when the 2D detector has not been specifically trained for that purpose. Second, the projection is usually preceded by Non Maximum Suppression (NMS) on the output of the 2D detector, which does not take into account the multi-camera geometry to resolve ambiguities.

Ideally, the power of Deep Learning should be combined with occlusion reasoning much earlier in the detection process than is normally done. To this end, we designed a joint CNN/CRF model whose posterior distribution can be approximated by Mean-Field inference using standard differentiable operations. Our model is trainable end-to-end and can be used in both supervised and unsupervised scenarios.

More specifically, we reason on a discretized ground plane in which detections are represented by boolean variables. The CRF is defined as a sum of innovative high-order terms whose values are computed by measuring the discrepancy between the predictions of a generative model that accounts for occlusions and those of a CNN that can infer that certain image patches look like specific body parts. To these terms, we add unary and pairwise ones to increase robustness and model physical repulsion constraints.

To summarize, our contribution is a joint CNN/CRF pipeline that performs detection for MCMT purposes in such a way that NMS is not required. Because it explicitly models occlusions, our algorithm operates robustly even in crowded scenes. Furthermore, it outputs probabilities of presence on the ground plane, as opposed to binary detections, which can then be linked into full trajectories using a simple flow-based approach [Berclaz et al., 2011].

6.2 Related Work

In this section, we first discuss briefly recent Deep Learning approaches to people detection in single images. We then move on to multi-image algorithms and techniques for combining CNNs and CRFs.

6.2.1 Deep Monocular Detection

As in many other domains, CNN-based algorithms [Ren et al., 2015, Redmon et al., 2016] have become for very good for people detection in single images and achieve state-of-the-art performance [Zhang et al., 2016]. Algorithms in this class usually first propose potential candidate bounding boxes with scores assigned to them. They then perform Non-Maximum Suppression (NMS) and return a final set of candidates. The very popular method of Ren et al. [2015] performs both steps in a single CNN pass through the image. It returns a feature map in which a feature vector of constant dimension is associated to each image pixel. For any 2D bounding-box of any size in that image, a feature vector of any arbitrary dimension can then be computed using Region Of Interest (ROI) pooling and fed to a classifier to assess whether the bounding box does indeed correspond to a true detection.

While this algorithm has demonstrated its worth on many benchmarks, it can fail in crowded scenes such as the one of Figure 6.1. This is perennial problem of single-image detectors when people occlude each other severely. One solution to this problem is to rely on cameras with overlapping fields of view, as discussed below.

6.2.2 Multi-Camera Pedestrian Detection

Here, we distinguish between recent algorithms that rely on Deep Learning but do not explicitly account for occlusions and older ones that model occlusions and geometry but appeared before the Deep Learning became popular. Our approach can be understood as a way to bring together their respective strengths.

The recent algorithm of Xu et al. [2016] runs a monocular detector similar to the one of Ren et al. [2015] on multiple views and infers people ground locations from the resulting detections. However, this method is prone to errors both because the 2D detections are performed independently of each other and because combining them by projecting them onto the ground plane involves reprojection errors and ignores occlusions. Yet, it is representative of the current MCMT state-of-the-art and is benchmarked against much older algorithms [Fleuret et al., 2008, Berclaz et al., 2011] that rely on background subtraction instead of a Deep Learning approach.

These older algorithms use multiple cameras with overlapping fields of view to leverage geometrical or appearance consistency across views to resolve the ambiguities that arise in crowded scenes and obtain accurate 3D localisation [Fleuret et al., 2008, Alahi et al., 2011, Peng et al., 2015]. They rely on Bayesian inference and graphical models to enforce detection sparsity. For example, the Probabilisitic Occupancy Map (POM) approach Fleuret et al. [2008] takes background subtraction images as input and relies on Mean Field inference to compute probabilities of presence in the ground plane. More specifically, given several cameras with overlapping fields of view of a discretized ground plane, POM first performs background subtraction. It then uses a generative model that represents humans as simple rectangles in order to create synthetic ideal images that would be observed if people were at given locations. Under this model of the image given the true occupancy, it approximates the probabilities of occupancy at every location using Mean Field inference. Because the generative model explicitly accounts for occlusions, POM is robust and often performs well. But it relies on background subtraction results as its only input, which is not discriminative enough when the people density increases, as shown in Figure 6.1. The algorithm of Alahi et al. [2011] operates on similar principles as POM but introduces more sophisticated human templates. Since it also relies on background subtraction, it is subject to the same limitations when the people density increases. And so is the algorithm of Peng et al. [2015] that introduces a more complex Bayesian model to enhance the results of Alahi et al. [2011].

6.2.3 Combining CNNs and CRFs

Using a CNN to compute potentials for a Conditional Random Field (CRF) and training them jointly for structured prediction purposes has received much attention in recent years [LeCun et al., 2006, Do and Artieres, 2010, Domke, 2013, Zheng et al., 2015, Arnab et al., 2015, Kirillov et al., 2015, Larsson et al., 2017, Bagautdinov et al., 2017].

However, properly training the CRFs remains difficult because many interesting models yield intractable inference problems. A popular workaround is to optimize the CRF potentials so as to minimize a loss defined on the output of an inference algorithm. Back Mean-Field [Domke, 2013, Zheng et al., 2015, Arnab et al., 2015, Larsson et al., 2017] has emerged as a promising way to do this. It relies on the fact that the updates steps during Mean-Field inference are continuous and parallelizable Baqué et al. [2016]. It is therefore possible to represent these operations as additional layers in a Neural Network and back-propagate through it. So far, this method has mostly been demonstrated either for toy problems or for semantic segmentation with attractive potentials, whereas our approach also requires repulsive potentials.

6.3 Modeling Occlusions in a CNN Framework

The core motivation behind our approach is to properly handle occlusions, while still leveraging the power of CNNs. To do so, we must model the interactions between multiple people who occlude each other but may not be physically close to each other. Our solution is to introduce an *observation space*; a generative model for observations given where people are located in the ground plane; and a discriminative model that predicts expected observations from the images. We then define a loss function that measures how different the CNN predictions are from those generated by the model. Finally, we use a Mean-Field approach with respect to probabilities of presence in the ground plane to minimize this loss. We cast this computation in terms of minimizing the energy of a Conditional Random Field in which the interactions between nodes are non-local because the people who occlude each other may not be physically close, which requires long range high-order terms.

In the remainder of this section, we first introduce the required notations to formalize our model. We then define a CRF that only involves high-order interaction potentials. Finally, we describe a more complete one that also relies on unary and pairwise terms.

6.3.1 Notations

We discretrize the ground plane in grid cells and introduce Boolean variables that denote the presence or absence of someone in the cell. Let us therefore consider a discretized ground plane containing N locations. Let X_i be the boolean variable that denotes the presence of someone at location i. Let us assume we are given C RGB images $\mathbf{I}^c$ of size $H^c \times W^c$ from multiple views $1 \leq c \leq C$ and $\mathbf{I} = \{\mathbf{I}^1, \ldots, \mathbf{I}^C\}$. For each ground plane location i and camera c, let the smallest rectangular zone containing the 2D projection of a human-sized 3D cylinder located at i be defined by its top-left and bottom-right coordinates T_i^c and B_i^c. For a pixel $k \in \{1, \ldots, H^c\} \times \{1, \ldots, W^c\}$, let L_k^c be the set of such projections that contain k.

We also introduce a CNN that defines an operator $\mathscr{F}(\cdot; \theta_F)$, which takes as input the RGB image of camera c and outputs a feature map $\mathscr{F}^c = \mathscr{F}(\mathbf{I}^c; \theta_F)$, where θ_F denotes the network's parameters. It contains a d-dimensional vector $\mathscr{F}_k^c$ for each pixel k.

6.3.2 High-Order CRF

We take the energy of our CRF to be a sum of High-Order potentials $\phi_{\mathrm{h}}^{c,k}$, one for each pixel. They handle jointly detection, and occlusion reasoning while removing the need for Non-Maximum Suppression. Each of these potentials use Probability Product Kernels to represent the agreement between a generative model and a discriminative model over the *observation space*, at a given pixel, as depicted in Fig. 6.2. We therefore write

$$
\begin{aligned}
P(\mathbf{X} \mid \mathbf{I}) &= \frac{1}{Z} \exp -E_{\mathrm{h}}(\mathbf{X} \mid \mathscr{F}(I; \theta_F)) \,, \\
E_{\mathrm{h}}(\mathbf{X}; \mathscr{F}) &= \sum_{1 \leq c \leq C, k \in \{1,\ldots,H^c\} \times \{1,\ldots,W^c\}} \phi_{\mathrm{h}}^{c,k}(\mathbf{X} \mid \mathscr{F}_k^c) \,.
\end{aligned}
\tag{6.1}
$$

Assuming we know the values of the occupancy variables $\mathbf{X}$, the generative model computes distributions over the set of *observations*. For each pixel in each image, it considers the corresponding line of sight and computes a distribution of vectors depending on the probability that it actually belongs to the successive people it traverses. This results in images whose pixels are vectors representing a distribution of 2D vectors,

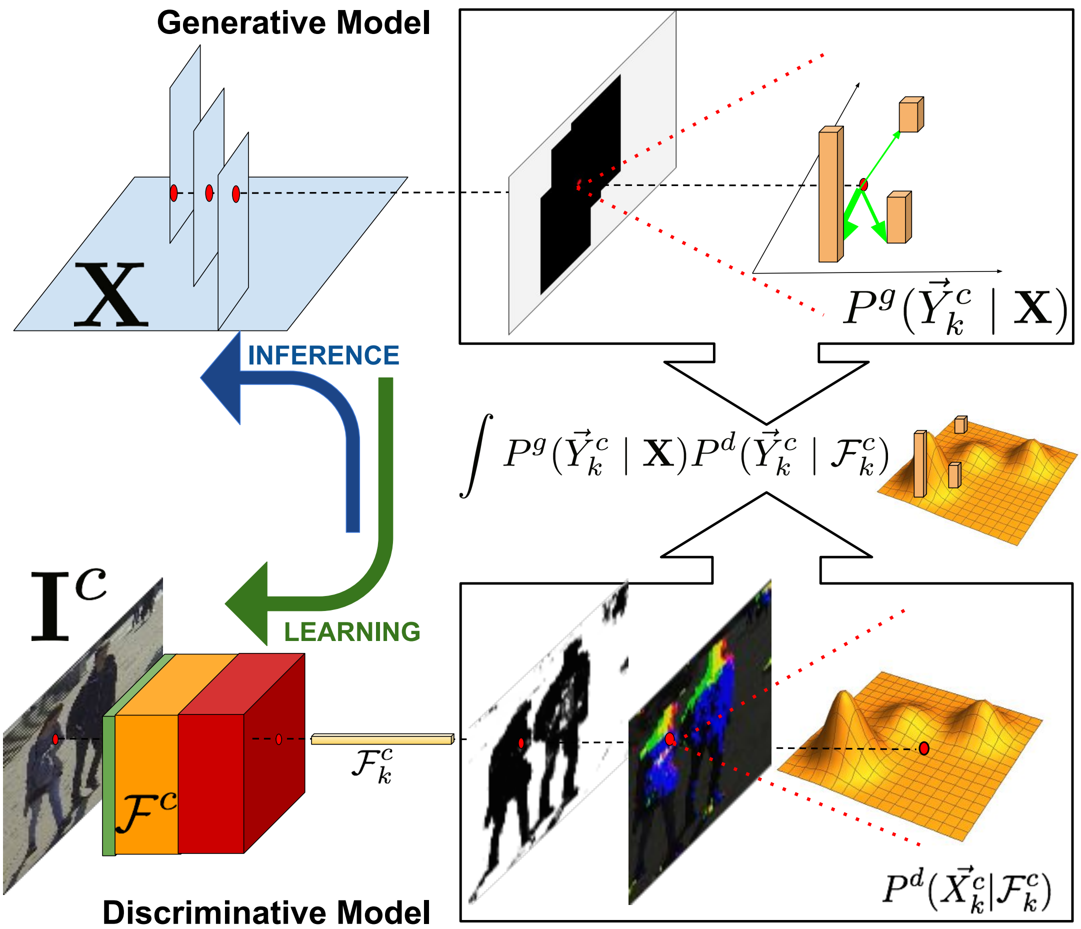

Generative Model
$\mathbf{X}$
$P^g(\vec{Y}_k^c \mid \mathbf{X})$
INFERENCE
LEARNING
$\int P^g(\vec{Y}_k^c \mid \mathbf{X}) P^d(\vec{Y}_k^c \mid \mathcal{F}_k^c)$
$\mathbf{I}^c$
$\mathcal{F}^c$
$\mathcal{F}_k^c$
$P^d(\vec{X}_k^c|\mathcal{F}_k^c)$
Discriminative Model

Figure 6.2 – Schematic representation of our High-Order potentials as described in Section 6.3.2.

the observations, as depicted in the top row of Fig 6.2. Our discriminative model relies on a CNN which tries to predict similar distributions of 2D vectors, directly by looking at the image. For ease of understanding, we first present in more details a simple version of our High-Order potentials $\phi_{\mathrm{h}}^{c,k}$. It assumes that our observations are zeros and ones at every pixel. The discriminative model therefore acts much as the background subtraction algorithms used in Fleuret et al. [2008] did. We then extend them to take into account the 2D vector output of our discriminative model.

Simple Generative Model

We first introduce a binary *observation* variable $Y_k^c \in \{0,1\}$ over which we define two distributions P^g and P^d produced by the generative and discriminative model respectively.

We take the distribution P^g to be

$$\begin{aligned} &P^g(Y_k^c = 1|\mathbf{X}) = 0, \text{ if } X_i = 0 \,\forall\, i \in L_k^c\,, \\ &P^g(Y_k^c = 1|\mathbf{X}) = 1 \text{ otherwise,} \end{aligned} \tag{6.2}$$

and the discriminative one P^d to be $P^d(Y_k^c|\mathscr{F}_k^c) = f_\mathrm{b}(\mathscr{F}_k^c;\theta_b)$, where F_k^c is the d-dimensional feature vector associated to pixel k introduced above and f_b is a Multi-Layer Perceptron (MLP) with weights θ_b. In other words, f_b plays the role of a CNN-based semantic segmentor or background-subtraction.

For each pixel, we then take the high-order potential to be the dot product between the distributions

$$\phi_\mathrm{h}^{c,k}(Z;\mathscr{F}_k^c) = -\mu_\mathrm{h} \log \int_{Y_k^c \in \{0,1\}} P^g(Y_k^c|\{Z_i\}_{i\in L_k^c}) P^d(Y_k^c|\mathscr{F}_k^c)\,, \tag{6.3}$$

as in the probability product kernel method of Jebara et al. [2004]. Intuitively, $\phi_\mathrm{h}^{c,k}$ is high when the segmentation produced by the network matches the projection of the detections in each camera plane using the simple generative model of Equation 6.2. μ_h is an energy scaling parameter.

Full Generative Model

The above model correctly accounts for occlusions and geometry but ignores much image information by focusing on background / foreground decisions. To refine it, we model the part of the bounding-box a pixel belongs to rather than just the fact that it belongs to a bounding-box. To this end, we redefine the Boolean auxiliary variable Y_k^c as

$$\vec{Y}_k^c \in \{0\} \cup \mathbb{R}^2\,, \tag{6.4}$$

where the label 0 represents background as before, and a label in $\mathbb{R}^2$ denotes the displacement with respect to the center of the body of the visible person at this pixel location.

To extend the simple model and account for what part of a bounding-box pixel k belongs to if it does, we sample from the distribution $P^g(\vec{Y}_k^c|\{X_i\}_{i\in L_k^c})$. To this end, let us assume without loss of generality that the L^c_k are ordered by increasing distance to the camera, as shown in the top left corner of Fig. 6.2. We initialize the variables $\vec{Y}^c_k$ to 0. Then, for each i in L^c_k such that $X_i = 1$, we draw a boolean random variable O_i with fixed expectancy o. If $O_i = 1$, then

$$\begin{aligned} \vec{Y}_k &= \vec{y}^i_k\,, \\ &= \left(\frac{k_x - 0.5(T^c_{i\,x} + B^c_{i\,x})}{B^c_{i\,x} - T^c_{i\,x}} ; \frac{k_y - 0.5(T^c_{i\,y} + B^c_{i\,y})}{B^c_{i\,y} - T^c_{i\,y}} \right), \end{aligned} \tag{6.5}$$

that is, the relative location of pixel k with respect to the projection of detection i in camera c, as depicted in the upper right corner of Fig. 6.2.

We define the distribution $P^d(\vec{Y}_k|\mathscr{F}^c_k)$ as an M-Modal Gaussian Mixture

$$\begin{aligned} &P^d(\vec{Y}_k = 0) = f_{\mathrm{b}}(\mathscr{F}^c_k;\theta_b)\,, \\ &P^d(\vec{Y}_k|\vec{X}_k \neq 0) = \sum_{1\leq m\leq M} f_{\mathrm{h}}(\mathscr{F}^c_k;\theta_{\mathrm{h}})_m \mathcal{N}(\vec{Y}_k - \alpha_m;\sigma_m)\,, \end{aligned} \tag{6.6}$$

as depicted in the bottom right corner of Fig. 6.2. As a result, $P^d(\vec{Y}_k = 0)$ is the same as in the simple model but $P^d(\vec{Y}_k|\vec{Y}_k \neq 0)$ encodes more information. (α_m, σ_m) are Gaussian parameters learned for each mode m. f_{h} is a MLP parametrized by θ_{h} that outputs M normalized real probabilities where M is a meta-parameter of our model. Similarly, $f_{\mathrm{b}}(\mathscr{F}^c_k;\theta_b)$ is a background probability.

Finally, as in Equation 6.3, we take our complete potential to be

$$\phi^{c,k}_{\mathrm{h}}(\mathbf{X}\,|\,\mathscr{F}^c_k) = -\mu_{\mathrm{h}} \log \int_{\vec{X}_k\in\{0\}\cup\mathbb{R}^2} P^g(\vec{Y}_k|\{X_i\}_{i\in L^c_k}) P^d(\vec{Y}_k|\mathscr{F}^c_k)\,. \tag{6.7}$$

6.3.3 Complete CRF

To increase the robustness of our CRF, we have found it effective to add, to the high-order potentials of Equation 6.1, unary and pairwise ones to exploit additional image information. We therefore write our complete CRF model as

$$\begin{aligned} P(\mathbf{X} \mid \mathbf{I}) &= \frac{1}{Z} \exp\left[-E(\mathbf{X} \mid \mathscr{F})\right] , \\ E(\mathbf{X} \mid \mathscr{F}) &= E_{\mathrm{h}}(\mathbf{X} \mid \mathscr{F}) + \sum_{i \le N} \phi_{\mathrm{u}}^{i}(X_i \mid \mathscr{F}) + \sum_{i \le N, j \le N} \phi_{\mathrm{p}}(X_i, X_j), \end{aligned} \tag{6.8}$$

where ϕ_{h} is the high-order CRF of Equation 6.1, the ϕ_{u}^{i} are unary potentials, and ϕ_{p} pairwise ones, which we describe below.

Unaries

The purpose of our unary potentials is to provide a prior probability of presence at a given location on the ground, before considering the occlusion effect and non maximum suppression. For each location i and camera c, we use a CNN $f_{\mathrm{u}}(T_i^c, B_i^c, \mathscr{F}^c)$, which outputs a probability of presence of a person at location i. f_{u} works by extracting a fixed size feature vector from the rectangular region defined by T_i^c, B_i^c in $\mathscr{F}^c$, using an ROI pooling layer Ren et al. [2015]. A detection probability is finally estimated using an MLP. Estimates from the multiple cameras are pooled through a max operation

$$\phi_{\mathrm{u}}^{i}(X_i \mid \mathscr{F}) = -\mu_{\mathrm{u}} Z_i \max_c \log \frac{f_{\mathrm{u}}(T_i^c, B_i^c, \mathscr{F}_c)}{1 - f_{\mathrm{u}}(T_i^c, B_i^c, \mathscr{F}_c)} , \tag{6.9}$$

where μ_{u} is a scalar that controls the importance of unary terms compared to others.

Pairwise

The purpose of our pairwise potentials is to represent the fact that two people are unlikely to stand too close to each others.

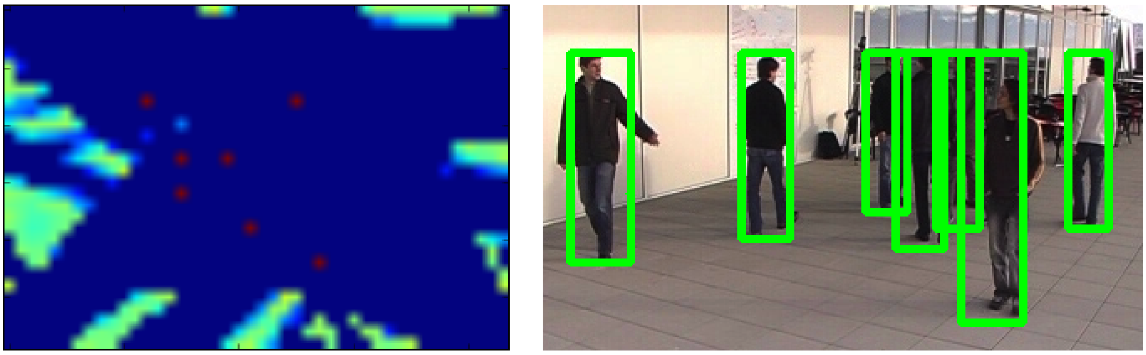
For all pairs of locations (i, j), let $E_{\mathrm{p}}^{i,j} = E_{\mathrm{p}}[|x_i - x_j|; |y_i - y_j|]$, where E_{p} is a 2D kernel function of of predefined size. We write

$$\phi_{\mathrm{p}}(X_i, X_j) = E_{\mathrm{p}}^{i,j} X_i X_j \tag{6.10}$$

for locations that are closer to each other than a predefined distance and 0 otherwise.

6.4 Inference and Derivation

Given the CRF of Equation 6.8 and assuming all parameters known, finding out where people are in the ground plane amounts to minimizing ϕ with respect to $\mathbf{X}$, the vector of binary variables that indicates which ground locations contain someone, which amounts to computing a Maximum-a-Posteriori of the posterior P. Instead of doing so directly, which would be intractable, we use Mean-Field inference Wainwright and Jordan [2008] to approximate of P by a fully-factorised distribution Q. As in Fleuret et al. [2008], this produces a Probability Occupancy Map, that is, a probability of presence $Q(X_i = 1)$, at each location, such as the one depicted by Fig. 6.3.



(a) (b)

Figure 6.3 – Output. (a) Given a set of images of the same scene, ours algorithm produces a Probabilistic Occupancy Map, that is, a probability of presence at each location of the ground plane. Red values indicate probabilities close to 1 and blue ones values close to zero. (b) Because the probabilities are very peaked, they can easily be thresholded to produce detections whose projections are the green boxes in the original image(s).

To perform this minimization, we rely on the natural-gradient descent scheme of Baqué et al. [2016]. It involves taking gradient steps that are proportional to

$$\nabla_{\eta_i} = \mathbb{E}_Q\left[(\phi(\mathbf{X}\,|\,\mathscr{F}))\,|X_i = 1\right] - \mathbb{E}_Q\left[(\phi(\mathbf{X},\mathscr{F}))\,|X_i = 0\right], \tag{6.11}$$

for each location i. The contribution to ∇_{η_i} of the unaries derives straightforwardly from Equation 6.9. Similarly, the one of the pairwise potentials of Equation 6.10 is

$$\begin{aligned}(\nabla_{\eta_i})_{\text{p}} &= -\sum_j E_{\text{p}}^{i,j} Q_j(X_j = 1)\,, \qquad (6.12)\\ &= -\sum_j E_{\text{p}}[|x_i - x_j|, |y_i - y_j|] Q_j(X_j = 1)\,,\end{aligned}$$

which can be implemented as a convolution over the current estimate of the probabilistic occupancy map Q with the two dimensional kernel $E_{\mathrm{p}}[.,.]$. This makes it easy to unroll the inference steps using a Deep-Learning framework.

Formulating the contributions of the higher-order terms of Equation 6.7 is more involved and requires simplifications. We first approximate the Gaussians used in Equation 6.6 by a function whose value is 1 in B_m and ϵ elsewhere, where B_m is the rectangle of center α_m and half-size $3\sigma_m$. Note that this approximation is only used for inference purposes, and that during training, it keeps its original Gaussian form. We then threshold the Gaussian weights f_{h} resulting in the binary approximation $\tilde{f}_{\mathrm{h}}$. This yields a binary approximation $\widetilde{P}^{\mathrm{d}}(\vec{Y}_k)$ of $P^{\mathrm{d}}(\vec{Y}_k)$. Note that the corresponding approximate potential $\widetilde{\phi}_{\mathrm{h}}^{c,k}(Z,\mathscr{F}_k^c)$ can be either $O(\log\epsilon)$, if $P^{\mathrm{d}}(\vec{Y}_k, b_k = 1; Z) = 0$ for all $\vec{Y}_k$ such that $P^{\mathrm{d}}(\vec{Y}_k) > \epsilon$ or $O(\log(1))$. Hence, the configurations where $\phi_{\mathrm{h}}^{c,k}(\mathbf{X},\mathscr{F}_k^c) = O(\log\epsilon)$ will dominate the others when computing the expectancies. This yields the approximation of Equation 6.11,

$$\widetilde{\nabla\eta_i} = -C(\mathbb{E}_Q\left[\Delta(\mathbf{X})|X_i = 1\right] - \mathbb{E}_Q\left[\Delta(\mathbf{X})|X_i = 0\right])\,, \tag{6.13}$$

where $C = -log\epsilon$ is a constant and $\Delta(\mathbf{X})$ is a binary random variable, which takes value 1 if $\widetilde{\phi}_{\mathrm{h}}^{c,k}(\mathbf{X}\,|\,\mathscr{F}_k^c) = 0$, and 0 otherwise. Note that $\phi_{\mathrm{h}}^{c,k}(\mathbf{X}\,|\,\mathscr{F}_k^c) = O(\log(1))$ iff

$$\exists i \leq N, m \leq M \text{ s.t } \tilde{f}_{\mathrm{h}}(\mathscr{F}_k^c;\theta_{\mathrm{h}})_m = 1 \text{ and } \vec{Y}_k^i \in B_m\,. \tag{6.14}$$

This means that for each pixel k, given a thresholded output from the network $\tilde{f}_{\mathrm{h}}(\mathscr{F}_k^c;\theta_{\mathrm{h}})$, we obtain a list of *compatible explanations* $\mathscr{C}_k \subset \{1,\ldots,N\}$ such that pixel k defines a very simple pattern-based potential of the form 1 if $X_i = 0\ \forall\, i \in \mathscr{C}_k$, 0 otherwise, which is similar to the potentials used in the Mean-Fields algorithms of Vineet et al. [2014], Fleuret et al. [2008], Kohli and Rother [2012], Arnab et al. [2015].

6.5 Training

We now show how our model can be trained first in a supervised manner and then in an unsupervised one.

6.5.1 Supervised Training

Let us first assume that we observe D data point $(\mathbf{X}^0,\mathbf{I}^0),\ldots,(\mathbf{X}^D,\mathbf{I}^D)$, where $\mathbf{I}^d$ represents a multi-view image and $\mathbf{X}^d$ the corresponding ground truth presences. The purpose of training is then to optimize the network parameters $\theta_F,\theta_\mathrm{u},\theta_\mathrm{h}$ defined in Sections 6.3.1, 6.3.3 and 6.3.2 respectively, the gaussian parameters α,σ of Equation 6.6 and the energy-scaling meta-parameters $\mu_\mathrm{u},\mu_\mathrm{h}$ of Eqs. 6.9 and 6.3 to maximize $\sum\limits_{d\leq D}\log P(\mathbf{X}^d|\mathbf{I}^d)$. It cannot be done directly using Equation 6.8 because computing the partition function Z is intractable.

Back Mean-Field An increasingly popular work-around is to optimize the above-mentioned parameters to ensure that the output of the Mean-Field inference fits the ground truth. In other terms, let $Q_{\theta_F,\theta_\mathrm{u},\theta_\mathrm{h},\alpha,\sigma}(\mathbf{X}\mid\mathbf{I})$ be the distribution obtained after inference . We look for

$$\underset{\theta_F,\theta_\mathrm{u},\theta_\mathrm{h},\alpha,\sigma}{\operatorname{argmax}}\sum_{(\mathbf{X}^d|\mathbf{I}^d)}\log Q_{\theta_F,\theta_\mathrm{u},\theta_\mathrm{h},\alpha,\sigma}(\mathbf{X}=\mathbf{x}^d\mid\mathbf{I}^d)\,. \tag{6.15}$$

Since $Q_{\theta_F,\theta_\mathrm{u},\theta_\mathrm{h},\alpha,\sigma}(\mathbf{X}=\mathbf{x}^d\mid\mathbf{I}^d)$ is computed via a sequence of operations which are all differentiable with respect to the parameters $\theta_F,\theta_\mathrm{u}$, and θ_h, it is therefore possible to solve Equation 6.15 by stochastic gradient descent Domke [2013], Zheng et al. [2015].

Pre-training However, it still remains difficult to optimize the whole model from scratch. We therefore pre-train our potentials separately before end-to-end fine-tuning. More precisely, the CNN f_u that appears in the unary terms of Equation 6.9 is trained as a standard classifier that gives the probability of presence at a given location, given the projection of the corresponding bounding-box in each camera view. For each data point, this leaves the high-order terms for which we need to optimize

$$\sum_{c}\sum_{k\in\mathscr{P}_c}\log(\phi_\mathrm{h}^{c,k}(\mathbf{X}^d\mid\mathscr{F}_k^c))\,, \tag{6.16}$$

with respect to the parameters of the Gaussian Mixture network θ_h,α, and σ. We use Jensen's inequality to take our generative distribution P^g out of the integral in Equation 6.7 and approximate it by random sampling procedure described in Section 6.3.2. We rewrite the set of samples for $\vec{Y}_k^c$ from all the pixels from all the cameras from all the data-points as $S(\mathbf{X}^0,\ldots,\mathbf{X}^D)$. The optimization objective of Equation 6.16 can then be rewritten as

$$\sum_{\vec{y}_s \in S(\mathbf{X}^0,\ldots,\mathbf{X}^D)} \log(P^d(\vec{y}_s | \mathscr{F}_k^c, \theta_\mathrm{h}, \alpha, \sigma)) \,, \tag{6.17}$$

which is optimized by alternating a standard stochastic gradient descent for the θ_h parameters and a closed form batch optimization for α,σ. This procedure is similar to one often used to fit a Mixture of Gaussians, except that, during the E-Step, instead of computing the class probabilities directly to increase the likelihood, we optimise the parameters of the network through gradient descent.

This pre-training strategy creates potentials which are reasonable but not designed to be commensurate with each others. We therefore need to choose the two energy parameters scalars μ_u, and μ_h, via grid-search in order to optimize the relative weights of Unary and High-Order potentials before using the Back-Mean field method.

6.5.2 Unsupervised Training

In the absence of annotated training data, inter-view consistency and translation invariance still provide precious a-priori information, which can be leveraged to train our model in an unsupervised way.

Let us assume that the background-subtracting part of the network, which computes f_b, the MLP introduced in Section 6.3.2, is reasonably initialized. In practice, it is easy to do either by training it on a segmentation dataset or by relying on simple background subtraction to compute f_b. Then, starting from initial values of the parameters θ, we first compute the Mean-Field approximation of $P(\mathbf{X} \,|\, \mathbf{I}_0, \theta)$, which gives us a first lower bound of the partition function. We then sample $\mathbf{X}$ from Q and use that to train our potentials separately as if these samples were ground truth-data, using the supervised procedure of Section 6.5.1. We then iterate this procedure, that is, Mean-Field inference, sampling from $\mathbf{X}$, and optimizing the potentials sequentially. This can be interpreted as an Expectation-Maximization (EM) Blei et al. [2016] procedure to optimize an Expected

Lower Bound (ELB) to the partition function Z of Equation 6.8.

6.6 Implementation Details

Our implementation uses a single VGGNet-16 Network with pre-trained weights. It computes features that will then be used to estimate both unary and pairwise potentials. The features map $\mathscr{F}^c = \mathscr{F}(\mathbf{I}^c;\theta_F)$ is obtained by upsampling of the convolutional layers.

Similarly to the classification step in Ren et al. [2015], we restrict the Region-Of-Interest pooling layer (ROI) to the features from the last convolutional layer of VGGNet. The output of the ROI is a 3x3x1024 tensor, which is flattened and input to a two layers MLP with ReLU non-linearities. In a similar way as in previous works on segmentation Zheng et al. [2015], we use a two layers MLP to classify each hyper-column of our dense features map $\mathscr{F}^c = \mathscr{F}(\mathbf{I}^c;\theta_F)$ to produce segmentation f_b and Gaussian Class f_h probabilities.

We use $M = 8$ modes for Multi-Modal Gaussian distribution of Eq. 6.6 for all our experiments and we have not assessed the impact of this choice on the performance. Besides, our kernel defining the pairwise potentials of Eq. 6.10 takes an arbitrary uniform constant value. For unsupervised training, we use a fixed number of 6 EM iterations, which we empirically found to be enough.

Finally, all our pipeline is implemented end-to-end using standard differentiable operations from the Theano Deep-Learning library Theano Development Team [2016]. For Mean-Field inference, we use a fixed number of iterations (30) and step size (0.01).

6.7 Evaluation

6.7.1 Datasets, Metrics, and Baselines

We introduce here the datasets we used for our experiments, the metrics we relied on to evaluate performance, and the baselines to which we compared our approach.

Datasets.

- **ETHZ**. It was acquired using 7 cameras to film the dense flow of students in front of the ETHZ main building in Zürich for two hours. It comprises 250 annotated temporal 7-image frames in which up to 30 people can be present at a time. We used 200 of these frames for training and validation and 50 for evaluation. See the image of Figure 6.1 for a visualization.

- **EPFL**. The images were acquired at 25 fps on the terrace of an EPFL building in Lausanne using 4 DV cameras. The image of Figure 6.3 is one of them. Up to 7 people walk around for about 3 1/2 minutes. As there are only 80 annotated frames, we used them all for evaluation purposes and relied either on pre-trained models or unsupervised training.

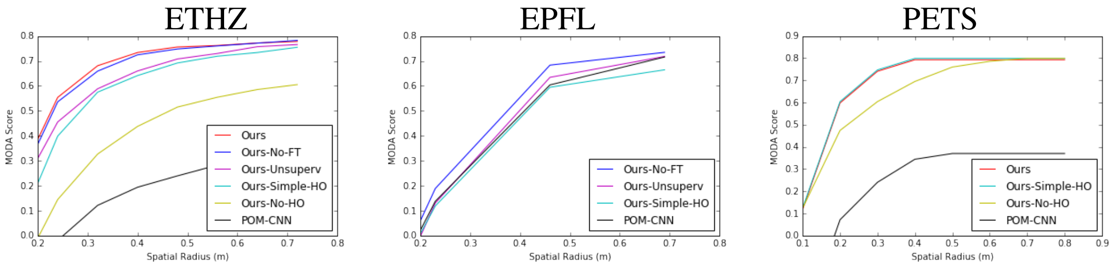
- **PETS**. The standard **PETS 2009** (PETS S2L1) is widely used for monocular and multi-camera detection. It contains 750 annotated images and was acquired from 7 cameras. It is a simple dataset in the sense that it is not very crowded, but the calibration is inaccurate and the image quality low.

Metrics. Recall from Section 6.4, that our algorithms produces Probabilistic Occupancy Maps, such as the ones of Figure 6.3. They are probabilities of presence of people at ground locations and are very peaky. We therefore simply label locations where the probability of presence is greater than 0.5 as being occupied and will refer to these as *detections*, without any need for Non-Maximum suppression. We compute false positive (FP), false negative (FN) and true positives (TP) by assigning detections to ground truth using Hungarian matching. Since we operate in the ground plane, we impose that a detection can be assigned to a ground truth annotation only if they are less than a distance r away. Given FP, FN and TP, we can evaluate:

- **Multiple Object Detection Accuracy (MODA)** which we will plot as a function of r, and the **Multiple Object Detection Precision (MODP)** Kasturi et al. [2009].

- **Precision-Recall**. Precision and Recall are taken to be TP/(TP + FN) and TP/(TP+FP) respectively.

We will report MODP, Precision, and Recall for $r = 0.5$, which roughly corresponds to the width of a human body. Note that these metrics are unforgiving of projection errors because we measure distances in the ground plane, which would not be the case if we evaluated overlap in the image plane as is often done in the monocular case. Nevertheless, we believe them to be the metrics for a multi-camera system that computes the 3D location of people.



	ETHZ		EPFL		PETS	
Method	**Precision / Recall**	**MODP**	**Precision / Recall**	**MODP**	**Precision / Recall**	**MODP**
Ours	**95 / 80%**	**53.8%**	-	-	-	-
Ours-No-FT	93 / 80%	53.4%	**88 / 82%**	**48.3%**	**93 / 87%**	**60.4%**
Ours-Unsuperv	86 / 80%	49.8%	80 / 85%	47.5%	-	-
Ours-Simple-HO	87 / 70%	47.5%	85 / 75%	43.2%	93 / 87%	60.4%
Ours-No-HO	84 / 55%	34.4%	37 / 68%	23.3%	93 / 81%	55.2%
POM-CNN	75 / 55%	30.5%	80 / 78%	45.9%	90 / 86%	42.9%
RCNN-2D/3D	68 / 43%	18.4%	39 / 50%	21.6%	50 / 63%	27.6%

Figure 6.4 – Results on our three test datasets. **Top row.** MODA scores for the different methods as function of the radius r used to compute it, as discussed in Section 6.7.1. **Bottom row.** Precision/Recall and MODP for the different methods for $r = 0.5$. Some of the values are absent either due to the bad calibration of the data-set, or missing ground-truth, as explained in Sections 6.7.1 and 6.7.2. The numbers we report for the **RCNN-2D/3D** baseline are much lower than those reported in Xu et al. [2016] for the method that inspired it, in large part because we evaluate our metrics in the ground plane instead of the image plane and because Xu et al. [2016] uses a temporal consistency to improve detections.

Baselines and Variants of our Method. We implemented the following two baselines.

- **POM-CNN**. The multi-camera detector Fleuret et al. [2008] described in Section 6.2.2 takes background subtraction images as its input. In its original implementation, they were obtained using traditional algorithms Ziliani and Cavallaro [1999], Oliver et al. [2000]. For a fair comparison reflecting the progress that has occurred since then, we use the same CNN-based segmentor as the one use to segment the background, that is $f_{\mathrm{b}}(\mathscr{F}^c_k;\theta_b)_0$ from Equation 6.6.

- **RCNN-2D/3D**. The recent work of Xu et al. [2016] proposes a MCMT tracking framework that relies on a powerful CNN for detection purposes Ren et al. [2015], as discussed in Section 6.2.2. Since the code of Xu et al. [2016] is not publicly available, we reimplemented their detection methodology as faithfully as possible but *without* the tracking component for a fair comparison with our approach that operates on images acquired at the same time. Specifically, we run the 2D detector Ren et al. [2015] on each image. We then project the bottom of the 2D bounding box onto the ground reference frame as in Xu et al. [2016] to get 3D ground coordinates. Finally, we cluster all the detections from all the cameras using 3D proximity to produce the final set of detections.

To gauge the influence of the different components or our approach, we compared these baselines against the following variants of our method.

- **Ours**. Our method with all three terms in the CRF model turned on, as described in Section 6.3.3, and fine tuned end-to-end through back Mean-Field, as described in Section 6.5.1.
- **Ours-No-FT**. **Ours** without the final fine-tuning.
- **Ours-Unsuperv**. Same as **Ours-No-FT** but the training is done without ground truth annotations, as described in Section 6.5.2.
- **Ours-Simple-HO** : We replace the full High-Order term of Section 6.3.2 with the simplified one that approximates the one of Fleuret et al. [2008], as described at the beginning of that section.
- **Ours-No-HO**. We remove the High-Order term of Section 6.3.2 altogether.

6.7.2 Results

We report our results on our three test datasets in Figure 6.4.

ETHZ. **Ours** and **Ours-No-FT** clearly dominate the **RCNN-2D/3D** and **POM-CNN** baselines, with **Ours** slightly outperforming **Ours-No-FT** because of the fine-tuning.

Simplifying the high-order term, as in **Ours-Simple-HO**, degrades performance and removing it, as in **Ours-No-HO**, degrades it even more. The methods discussed above rely on supervised training, whereas **Ours-Unsuperv** does not but still outperforms the baselines.

EPFL. Because the images have different statistics than those of **ETHZ**, the unary terms as well as the people detector **RCNN-2D/3D** relies on are affected. And since there is no annotated data for retraining, as discussed above, the performance of **Ours-No-HO** and **RCNN-2D/3D** drop very significantly with respect to those obtained on **ETHZ**. By contrast, the high order terms are immune to this, and both **Ours-No-FT** and **Ours-Unsuperv** hold their performances.

PETS. The ranking of the methods is the same as before except for the fact that **Ours-Simple-HO** does as well as **Ours-No-FT**. This is because the **PETS** dataset is poorly calibrated, which results in inaccurate estimates of the displacement vectors in the generative model of Section 6.3.2. As a result, it does not deliver much of a performance boost and we therefore did not find it meaningful to report results for unsupervised training and fine-tuning of these High-Order potentials.

From Detections to Trajectories. Since our method produces a Probability Occupancy Map for every temporal frame in our image sequences, we can take advantage of a simple-flow based method Berclaz et al. [2011] to enforce temporal consistency and produce complete trajectories. As shown in Figure 6.5 this leads to further improvements for all three datasets.

Method	**ETHZ**	**EPFL**	**PETS**
Ours	74.1%	68.2%	79.8%
Ours + Berclaz et al. [2011]	75.2%	76.9%	83.4%

Figure 6.5 – MODA scores for $r = 0.5$ before and after enforcing temporal consistency.

6.8 Discussion

We introduced a new CNN/CRF pipeline that outperforms the state-of-the art for multi-camera people localization in crowded scenes. It handles occlusion while taking full advantage of the power of a modern CNN and can be trained either in a supervised or unsupervised manner.

A limitation, however, is that the CNN used to compute our unary potentials still operates in each image independently as opposed to pooling very early the information from multiple images and then leveraging the expected appearance consistency across views. In future work, we will therefore investigate training such a CNN for people detection on multiple images simultaneously, jointly with our CRF.

7 Concluding Remarks

In this thesis, we studied and proposed improvements of the variational mean-field methods in Computer Vision. We applied this new tool to structured parameter learning in Conditional Random Fields. Our methodology was to find generic algorithmic solutions to fundamental issues and then show how these new tools could be used to solve practical problems better than previous approaches.

7.1 Summary and contributions

In Chapter 3, we proposed a new approach to mean-field inference, which is more efficient, better understood and brings better results than previous methods. The performance of this method has been acknowledged in other works, which use it for many different tasks, that we did not envision initially, such as cancer detection, ultrasound processing or image attribute prediction. Furthermore, our work sheds light on several ad-hoc heuristics that were used for parallel mean-field inference in conditional random fields, and we provide convergence guarantees for such methods.

In Chapter 4, we moved to the challenge of adding structure to the *naive* mean-field method. Since it looks for a fully factorized approximation to a complex posterior distribution by minimizing the KL-divergence to it, the standard mean-field approach is often too simplistic. Sometimes, this rough approximation is sufficient to extract the information of interest, and sometimes, the posterior is so complex that finding a good variational approximation to it, is almost impossible. However, is many cases of practical interest, the posterior has a clear multi-modal structure, with a limited number of modes. In order to handle such situations, we designed an efficient Multi-Modal Mean-Field approximation method. We showed, that, in practice, it can be used to propose multiple solutions to a CRF inference problem, which brings improved performance for several segmentation and tracking algorithms.

In Chapter 5, we used the tools developed in th two previous ones to design a novel parameter learning algorithm for Conditional Random Fields. This approach is based on the Multi-Modal Mean-Field method and we showed that several classical parameter learning algorithms for CRFs, can be interpreted as specific instantiations of ours. However, in the general case, our approach can be computationally costly, making it unappealing compared to fast Neural Network based learning, combined with the back mean-field method. We therefore proposed a simplification of our approach, which, while being much more computationally simple and as easily scalable as other popular methods, retains good performances.

Finally, in Chapter 6, we proposed to use Conditional Random Fields and our new approach to mean-field inference for people detection in a crowded scene from multiple views. We presented a new detection model, which extends the popular POM algorithm of [Fleuret et al., 2008] using CNN-based potentials. Our method performs better and is more robust to occlusions, illumination changes or other perturbations than previous approaches. In order to evaluate our algorithm in a truly challenging setting, we recorded and annotated a new multi-person, multi-camera tracking dataset, which has been made public and is known as the Wildtrack dataset [Chavdarova et al., 2018].

7.2 Limitations and future work

7.2.1 New Applications of Structured Learning for CRFs

In this thesis, we proposed a novel approach to structured parameters learning for Conditional Random Fields. We demonstrated performance gains on a limited number of practical applications , which is not yet enough to make Multi-Modal Back Mean-Field a widely used practical tool. Therefore, we hope that future work will aim at using one or another instantiation of our method to demonstrate its applicability on a wider range of CRF problems.

We envision applications in several domains of computer vision where the learning problems are inherently structured and where CRFs have long been used. First, we think that our method could be used for curvilinear structure delineation in medical images. Indeed, such problems are highly structured, because we want to reconstruct consistent paths and they are multi-modal since medical images are often noisy and ambiguous.

Multiple people 3D pose estimation is another area of computer vision where CRFs and structured learning have traditionally been used. Future research will aim at combining our Multi-Modal Back Mean-Field with state of the art Deep-Learning tools in this domain.

However, researchers will hopefully find many other tasks of interest where our algorithms can be put to use.

Because of the recent improvements of CNN-based semantic segmentation methods and of the introduction of large scale datasets such as the Cityscape one, very little ambiguities remain for the semantic segmentation of these images. Therefore, we think that standard semantic segmentation tasks would benefit only very marginally from our approach. However, it should be used in low data contexts, such as for medical imaging or potentially for multiple-instance semantic segmentation.

7.2.2 Multiple-people multi-camera tracking with Deep-Occlusion reasoning

The multi-camera setting is a popular and affordable solution to people tracking in densely crowded scenes. We have demonstrated that mean-field inference in Conditional Random Fields, can be combined with deep Convolutional Neural Networks to obtain state of the art results in this task. However, several challenges regarding this framework are still open.

In our work, tracking cannot be performed in real time, because of the computational cost of the inference model. Future work should look at more efficient implementations of out approach, potentially leveraging on recent progress of Deep-Learning libraries.

The inter-dataset transferability of our method should be improved in the future. Indeed, a CNN that was trained with a given camera and background setting will not perform as well on very different scenes. We are exploring solutions where the potentials are trained using large scale 2D datasets, such as the MSCOCO one, to improve the robustness of our algorithm.

Our framework is currently being used for behavioral analysis and social scene understanding tasks in videos. It will be extended, for instance to detect groups of peoples who interacting with each other. This new feature is being developed in collaboration with a retail company to analyze representative-customer interactions in their shops.

Bibliography

A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity Driven People Localization with a Heterogeneous Network of Cameras. *Journal of Mathematical Imaging and Vision*, 2011.

S-.I. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10 (2):251–276, 1998.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Gan. *arXiv preprint arXiv:1701.07875*, 2017.

A. Arnab, S. Zheng, S. Jayasumana, B. Romera-Paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, and P. H. S. Torr. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 35(1):37–52, Jan 2018.

Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip H. S. Torr. Higher Order Potentials in End-To-End Trainable Conditional Random Fields. *CoRR*, abs/1511.08119, 2015.

T. Bagautdinov, P. Fua, and F. Fleuret. Probability Occupancy Maps for Occluded Depth Images. In *Conference on Computer Vision and Pattern Recognition*, 2015.

T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese. Social Scene Understanding: End-To-End Multi-Person Action Localization and Collective Activity Recognition. In *Conference on Computer Vision and Pattern Recognition*, 2017.

T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling Facial Geometry Using Compositional VAEs. In *Conference on Computer Vision and Pattern Recognition*, 2018.

G.H. Bakir, T. Hofmann, B. Schölkopf, A.J. Smola, B. Taskar, and S.V.N. Vishwanathan. *Predicting Structured Data*. MIT Press, 2007.

P. Baqué, J. Hours, F. Fleuret, and P. Fua. A Provably Convergent Alternating Minimization Method for Mean Field Inference. *CoRR*, abs/1502.05832, 2015.

P. Baqué, T. Bagautdinov, F. Fleuret, and P. Fua. Principled Parallel Mean-Field Inference for Discrete Random Fields. In *Conference on Computer Vision and Pattern Recognition*, 2016.

P. Baqué, F. Fleuret, and P. Fua. Deep Occlusion Reasoning for Multi-Camera Multi-Target Detection. In *International Conference on Computer Vision*, 2017a.

P. Baqué, F. Fleuret, and P. Fua. Shape Optimization of Technical Devices via Gradient Descent using Convolutional Neural Network Proxies, September 2017b. PCT patent application (PCT/EP2017/072550).

P. Baqué, E. Remelli, F. Fleuret, and P. Fua. Geodesic convolutional shape optimisation. *CoRR*, abs/1802.04016, 2018. URL https://arxiv.org/pdf/1802.04016.pdf.

O. Barinova, V. Lempitsky, and P. Kohli. On Detection of Multiple Object Instances Using Hough Transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9), 2012.

D. Batra, P. Yadollahpour, A. Guzman-rivera, and G. Shakhnarovich. Diverse M-Best Solutions in Markov Random Fields. In *European Conference on Computer Vision*, pages 1–16, 2012.

H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1614–1627, 2014.

J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):1806–1819, 2011.

K. Bernardin and R. Stiefelhagen. Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *CoRR*, abs/1601.00670, 2016.

L. Bottou and O. Bousquet. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems*, 2008.

A. Bouchard-Côté and M.I. Jordan. Optimization of Structured Mean Field Objectives. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 67–74, 2009.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11): 1222–1239, 2001.

L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967. ISSN 0041-5553.

T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret. The Wildtrack Multi-Camera Person Dataset. In *Conference on Computer Vision and Pattern Recognition*, 2018.

L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR*, abs/1706.05587, 2017.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *International Conference for Learning Representations*, 2015.

W. Cho, S. Kim, S. Park, and J. Park. Mean Field Annealing EM for Image Segmentation. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, pages 568–5713, 2000.

T.M.T Do and T. Artieres. Neural Conditional Random Fields. In *International Conference on Artificial Intelligence and Statistics*, 5 2010.

J. Domke. Learning Graphical Model Parameters with Approximate Marginal Inference. *arXiv Preprint*, 2013.

T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo. A Semi-Automatic System for Ground Truth Generation of Soccer Video Sequences. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 559–564, 2009.

J.C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite Objective Mirror Descent. In *COLT*, pages 14–26, 2010.

F. Eaton and Z. Ghahrmani. Choosing a Variable to Clamp: Approximate Inference Using Conditioned Belief Propagation. In *International Conference on Artificial Intelligence and Statistics*, 2009.

D. Eigen, C. Puhrsch, and R. Fergus. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012.

Reza Zanjirani Farahani, Nasrin Asgari, Nooshin Heidari, Mahtab Hosseininia, and Mark Goh. Survey: Covering problems in facility location: A review. *Comput. Ind. Eng.*, 62 (1):368–407, February 2012.

F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008.

R. Fransens, C. Strecha, and L. Van Gool. A Mean Field EM-Algorithm for Coherent Occlusion Handling in Map-Estimation Prob. In *Conference on Computer Vision and Pattern Recognition*, 2006.

R. Frostig, S. Wang, P.S. Liang, and C.D. Manning. Simple MAP Inference via Low-Rank Relaxations. In *Advances in Neural Information Processing Systems*, pages 3077–3085, 2014.

F. Galasso, N.S. Nagaraja, T.J. Cardenas, T. Brox, and B.Schiele. A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis. In *International Conference on Computer Vision*, December 2013.

A.E. Gelfand and A.F.M Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

I. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, D. Warde-farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

L. Gorelick, Y. Boykov, O. Veksler, A.I. Ben, and A. Delong. Submodularization for Binary Pairwise Energies. In *Conference on Computer Vision and Pattern Recognition*, pages 1154–1161, 2014.

Inc. Gurobi Optimization. Gurobi Optimizer Reference Manual, 2016. URL http://www.gurobi.com.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

G.E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14:1771–1800, 2002.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9 (8):1735–1780, 1997.

M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14(1):1303–1347, May 2013.

J. Hur and S. Roth. Joint Optical Flow and Temporally Consistent Semantic Segmentation. *CoRR*, abs/1607.07716, 2016.

Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844, December 2004. ISSN 1532-4435.

L.P. Kadanoff. More is the Same; Phase Transitions and Mean Field Theories. *Journal of Statistical Physics*, 137(5):777, 2009.

J. Kappes, B. Andres, F.A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. Kausler, T. Kröger, J. Lellmann, et al. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *International Journal of Computer Vision*, 115(2):155–184, 2015.

R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. Zhang. Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009.

M. E. Khan, P. Baqué, F. Fleuret, and P. Fua. Kullback-Leibler Proximal Variational Inference. In *Advances in Neural Information Processing Systems*, 2015.

D P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv Preprint*, 2014.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *International Conference for Learning Representations*, 2014.

T.N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*, 2016.

A. Kirillov, D. Schlesinger, S. Zheng, B. Savchynskyy, P.H.S. Torr, and C. Rother. Joint Training of Generic CNN-CRF Models with Stochastic Optimization. *arXiv Preprint*, 2015.

P. Kohli and C. Rother. Higher-Order Models in Computer Vision. In Olivier Lezoray and Leo Grady, editors, *Image Processing and Analysis with Graphs*, pages 65–100. CRC Press, 2012.

D. Koller and N. Friedman. *Probabilistc Graphical Models*. The MIT Press, 2009.

V. Kolmogorov. A New Look at Reweighted Message Passing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5):919–930, 2015.

P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems*, 2011.

P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected Crfs with Gaussian Edge Potentials. In *arXiv Preprint*, 2012.

P. Krähenbühl and V. Koltun. Parameter Learning and Convergent Inference for Dense Random Fields. In *International Conference on Machine Learning*, pages 513–521, 2013.

A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

A. Kundu, V. Vineet, and V. Koltun. Feature Space Optimization for Semantic Video Segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2016.

M. Larsson, F. Kahl, S. Zheng, A. Arnab, P. Torr, and R. Hartley. Learning Arbitrary Potentials in CRFs with Gradient Descent. *arXiv Preprint*, 2017.

Y. LeCun, S. Chopra, and R. Hadsell. A Tutorial on Energy-Based Learning. *Predicting Structured Data*, 2006.

Y. Li and R. S. Zemel. Mean Field Networks. In *International Conference on Machine Learning*, 2014.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.-Y. Fu, and A.C. Berg. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision*, 2016.

J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *CoRR*, abs/1611.08408, 2016. URL http://arxiv.org/abs/1611.08408.

R. Mandeljc, S. Kovačičand M. Kristan, and J. Perš. Tracking by Identification Using Computer Vision and Radio. *Sensors*, 2012.

Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., New York, NY, USA, 2009. ISBN 019857083X, 9780198570837.

T.P. Minka. Expectation Propagation for Approximate Bayesian Inference. In *Conference on Uncertainty in Artificial Intelligence*, pages 362–369, 2001.

K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Onference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kholi. Decision Tree Fields. In *International Conference on Computer Vision*, November 2011.

N.M. Oliver, B. Rosario, and A.P. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang. Robust Multiple Cameras Pedestrian Detection with Multi-View Bayesian Network. *Pattern Recognition*, 48(5):1760–1772, 2015.

L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2016.

B.T. Polyak. Some Methods of Speeding Up the Convergence of Iteration Methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

E. Premachandran, D. Tarlow, and D. Batra. Empirical Minimum Bayes Risk Prediction: How to Extract an Extra Few % Performance from Vision Models with Just Three More Parameters. In *Conference on Computer Vision and Pattern Recognition*, June 2014.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., 1990. ISBN 1-55860-124-4.

V. Ramakrishna and D. Batra. Mode-Marginals: Expressing Uncertainty via Diverse M-Best Solutions. *Advances in Neural Information Processing Systems*, 2012.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2016.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, 2015.

J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical Deformable Dense Matching. *International Journal of Computer Vision*, 120(3): 300–323, 2016.

M. Saito, T. Okatani, and K. Deguchi. Application of the Mean Field Methods to MRF Optimization in Computer Vision. In *Conference on Computer Vision and Pattern Recognition*, June 2012.

Lawrence Saul and Michael I. Jordan. Exploiting Tractable Substructures in Intractable Networks. In *Advances in Neural Information Processing Systems*, pages 486–492, 1995.

M. Simonovsky and N. Komodakis. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. In *Conference on Computer Vision and Pattern Recognition*, pages 29–38, July 2017.

X. Sun, M. Christoudias, V. Lepetit, and P. Fua. Real-Time Landing Place Assessment in Man-Made Environments. *Machine Vision and Applications*, 2013.

J. W. Suurballe. Disjoint Paths in a Network. *Networks*, 4:125–145, 1974.

B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning Structured Prediction Models: A Large Margin Approach. In *International Conference on Machine Learning*, pages 896–903, 2005.

M. Teboulle. Entropic Proximal Mappings with Applications to Nonlinear Programming. *Mathematics of Operations Research*, 17(3):670–690, 1992.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

Cédric Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer, 2009 edition, September 2008. ISBN 3540710493.

V. Vineet, G. Sheasby, J. Warrell, and P.H.S. Torr. Posefield: An Efficient Mean-Field Based Method for Joint Estimation of Human Pose, Segmentation, and Depth. In *Conference on Computer Vision and Pattern Recognition*, pages 180–194, 2013.

V. Vineet, J. Warrell, and P.H.S. Torr. Filter-Based Mean-Field Inference for Random Fields with Higher-Order Terms and Product Label-Spaces. *International Journal of Computer Vision*, 110(3):290–307, 2014.

M.J. Wainwright and M.I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, January 2008.

B. Walsh. Markov chain monte carlo and gibbs sampling, 2004.

C. Wang, N. Komodakis, and N. Paragios. Markov Random Field Modeling, Inference & Learning in Computer Vision & Image Understanding: A Survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

A. Weller and J. Domke. Clamping Improves TRW and Mean Field Approximations. In *Advances in Neural Information Processing Systems*, 2015.

A. Weller and T. Jebara. Approximating the Bethe Partition Function. In *Uncertainty in Artificial Intelligence*, 2014.

J.M. Winn and C.M. Bishop. Variational Message Passing. In *Journal of Machine Learning Research*, pages 661–694, 2005.

Y. Xu, X. Liu, Y. Liu, and S.C. Zhu. Multi-View People Tracking via Hierarchical Trajectory Composition. In *Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

P. Yadollahpour, , D. Batra, and B. Shakhnarovich. Discriminative Re-Ranking of Diverse Segmentations. In *Conference on Computer Vision and Pattern Recognition*, 2013.

Jonathan S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, July 2001.

F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016.

S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. How Far Are We from Solving Pedestrian Detection? In *Conference on Computer Vision and Pattern Recognition*, pages 1259–1267, 2016.

Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016.

S. Zheng, S. Jayasumana, B. Romera-paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional Random Fields as Recurrent Neural Networks. In *International Conference on Computer Vision*, 2015.

F. Ziliani and A. Cavallaro. Image Analysis for Video Surveillance Based on Spatial Regularization of a Statistical Model-Based Change Detection. In *International Conference on Image Analysis and Processing*, 1999.

Pierre BAQUE

pierre.baque@epfl.ch – French Nationality

www.pierrebaque.com

EDUCATION:

2014-2018 ***PhD candidate - CVLab, EPFL*** (Lausanne, Switzerland) – Currently 4th year PhD candidate at the computer vision laboratory in EPFL. Working on Variational Inference and structured/deep learning for graphical models, applied to tracking and to multiple instance segmentation.
Initiated a project on 3D shape optimization for fluid dynamics using Deep Neural Networks which now involves 6 students and researchers. Applied for a patent under the name ''Shape optimization of technical devices via gradient descent using Convolutional Neural Network proxies''.

2012-2013 ***Master Parisien de Recherche Opérationnelle - Polytechnique/ENSTA*** (Paris) –
MSc in Operations Research, Computer-Science and Applied Mathematics. Top level courses in Graphs theory, Combinatorial Optimization and Optimal Stochastic Command. Average mark 17.1/20

2009 – 2012 ***Ecole Polytechnique*** (Paris) – France's top ranking University for science and engineering. Intensive and competitive courses in mathematics and physics. Specialization (MSc) in Applied Mathematics, Statistics and Probabilities. Got maximal mark "A" in all scientific courses in the three years. Ranked 28th out of 480 students.

2007 – 2009 ***Lycée Pierre de Fermat*** (Toulouse, France) – Preparatory program leading to nationwide competitive entrance examinations to the French Grandes Ecoles. Ranked 8th at the nationwide competitive entrance examinations to Polytechnique. Independently, Ranked 4th at the competitive entrance examinations to Centrale Paris.

WORK EXPERIENCE:

2016 **Sonalytic, London.** Artificial Intelligence Research Consultant.
Worked as a consultant on the development of a new generation of robust music identification software. Trying to push the state of the art toward a flexible and efficient usage of Deep Learning methods in this domain.

2012- 2014 **Credit-Suisse, London.** Exotic Equity Derivatives.
Platform and analytics development. Invented and leaded a dividend prediction tool which is now used by more than 20 traders. Created an optimal index replication software which was put in production for hedging multi-billion USD books.
Day to day trading. Was in charge of a USD 1.3 billion Delta-One book. Back-up trader on the main structured single-stock book in Credit-Suisse London.

2012-2013 **Thales, Paris.** 8-Months consulting. While studying in Masters, created my consulting structure to work with Thales DIS in order to explore new applications to their Linear Accelerator.

2011 **GoGorilla Media, New-York.** Internship in this marketing company. Created an Android and Ipad application which clients can rent and make their own. The sales team present it to clients as an add-on to their campaigns.

SELECTED PUBLICATIONS:

Arxiv **Geodesic Convolutional Shape Optimization.** *Pierre Baqué* · Edoardo Remelli · Francois Fleuret ·Pascal Fua

CVPR 2018 **WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection.** Tatjana Chavdarova · *Pierre Baqué* · Andrii Maksai · Stéphane Bouquet · Cijo Jose · Louis Lettry · Francois Fleuret · Pascal Fua · Luc Van Gool

ICCV 2017 **Deep-Occlusion reasoning for Multi-Camera Multi-People Tracking.** *Pierre Baqué* · Francois Fleuret ·Pascal Fua

CVPR 2017 **Multi-Modal Mean-Fields inference via cardinality based clamping.** *Pierre Baqué* · Francois Fleuret ·Pascal Fua

CVPR 2016 **Principled Parallel Mean-Field inference for discrete random fields.** *Pierre Baqué* · Timur Bagautdinov · Francois Fleuret ·Pascal Fua

NIPS 2015 **Kullback-Leibler Proximal Variational Inference.** M.E Khan · *Pierre Baqué* · Francois Fleuret ·Pascal Fua

GRANTS AND AWARDS:

2018 **Innogrant innovation fellowship (100 kCHF)**

2017 **Bridge Proof of Concept research funding (130 kCHF)**

OTHER PROJECTS:

2016 **EPFL Pedestrian annotation tool, Project Leader**. Managed a project involving four persons that aims at annotating large-scale pedestrian tracking datasets. We deployed a web-based tool and used Amazon Mechanical Turk to get human labeling. (http://pedestriantag.epfl.ch)

2015 **CarmenV2, co-Creator**. Developed a ropeways-transportation engineering software. Invented and implemented the first automatic ropeways implantation algorithm. The software was sold and is used by more than 10 engineering and constructing firms.

2012-2013 **CrowdGuess, Founding member**. Launched a Bitcoin predictive market online platform.

2012-2013 **So What Project, Founding member**. Market making and algorithmic trading on the electronic betting exchange through a JAVA API.

2010 **Image processing and statistics project**. Aimed at counting people in a demonstration with the help of a video camera. Achieved satisfactory results (sampling error + or -5%). Selected as the best project of the year by the computer science department of Polytechnique.

SKILLS:

Coding skills: Very strong knowledge of Python and strong background in Java, C++ and R.
Expertise with Theano, TensorFlow and CUDA GPU programming.
Working knowledge in web development .

Sports: Tae Kwon Do: Won French Championships 2003 and 2007. Junior category.
Running : Personal Best 32’04” on 10km and 1h10’30” on Half-Marathon.
1 month cycling trip across Kirghizstan and Ouzbekistan.

Languages: **French**, native **- English**, fluent;
Spanish, working knowledge **- Chinese**, basics