Input-dependent Regularization of Conditional Density Models

Matthias Seeger SEEGER@DAI.ED.AC.UK
Institute for Adaptive and Neural Computation, 5 Forrest Hill, Edinburgh EH1 2QL, UK

Abstract

We emphasize the need for input-dependent regularization in the context of *conditional density models* (also: *discriminative models*) like Gaussian process predictors. This can be achieved by a simple modification of the standard Bayesian data generation model underlying these techniques. Specifically, we allow the latent target function to be *a-priori dependent* on the distribution of the input points. While the standard generation model results in robust predictors, data with missing labels is ignored, which can be wasteful if relevant prior knowledge is available. We show that discriminative models like *Fisher kernel* discriminants and *Co-Training* classifiers can be regarded as (approximate) Bayesian inference techniques under the modified generation model, and that the template Co-Training algorithm is related to a variant of the well-known *Expectation-Maximization (EM)* technique. We propose a template EM algorithm for the modified generation model which can be regarded as generalization of Co-Training.

1. Introduction

There are two basic paradigms for supervised classification: the *generative* and the *discriminative* one. Within the former, we try to model the generative process for input points *conditioned* on each of the classes. While this is a very powerful and flexible approach, it is often very difficult to model real-world *class-conditional* distributions, especially if the observed data is sparse and is represented in a high-dimensional space. In this paper, we are concerned with *discriminative* models which, instead of modelling class regions, try to model the boundaries between them. Applied to the same problem, discriminative methods often use many fewer parameters and behave more robustly than generative ones. A major drawback of discriminative methods is, however, that there is no natural way to deal with missing or uncertain information.

The structure of the paper is as follows. In section 2, we formalize our setting and introduce the standard Bayesian data generation model for discriminative methods. We show that under this generation model, data with missing class labels is useless for Bayesian inference. In section 3, we introduce and discuss a modification of this model which leads to input-dependent regularization. In section 4, we give some examples of input-dependent regularization in the literature. Section 5 shows how *Co-Training*(Blum & Mitchell, 1998) can be regarded as Bayesian inference, and how the basic Co-Training algorithm is related to *Expectation-Maximization (EM)*. We also propose a template EM algorithm for the modified generation model.

2. The standard Bayesian data generation model

Let $\mathcal{X}$ be the space of *input points* $\boldsymbol{x}$, $\mathcal{T} = \{1, \dots, c\}$ the set of *(class) labels* t. We are given a labeled sample $D_l = \{(\boldsymbol{x}_1, t_1), \dots, (\boldsymbol{x}_n, t_n)\}$ drawn independently and identically distributed (i.i.d.) from an unknown distribution $P(\boldsymbol{x}, t)$. Let $X_l = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}, T_l = \{t_1, \dots, t_n\}$. Furthermore we have access to an unlabeled sample $D_u = X_u = \{\boldsymbol{x}_{n+1}, \dots, \boldsymbol{x}_{n+m}\}$ drawn i.i.d. from $P(\boldsymbol{x}) = \sum_t P(\boldsymbol{x}, t)$. We can regard the missing labels $T_u = \{t_{n+1}, \dots, t_{n+m}\}$ as latent data. The goal is to predict the class label t on unseen examples $\boldsymbol{x}$ with small *generalization error* $e(\hat{g}) = Pr\{\hat{g}(\boldsymbol{x}) \neq t\}$, where the probability is over $P(\boldsymbol{x}, t)$.

The Bayesian approach to discrimination is to build a model of the data generation process, encode available prior knowledge in prior distributions and then turn the Bayesian handle to make inference. However, being within the discriminative paradigm, we are interested in modelling $P(t|\boldsymbol{x})$ rather than the class distributions $P(\boldsymbol{x}|t)$. Within the standard Bayesian generation model, we choose a model class $\{P(t|\boldsymbol{x}, \boldsymbol{\theta})\}$ and encode what we *believe* to know about the (unknown) $P(t|\boldsymbol{x})$ in the *prior distribution* $P(\boldsymbol{\theta})$. For example, $\boldsymbol{\theta}$ might be the weights of a multi-layer perceptron, for which the usage of a weight-decay prior $P(\boldsymbol{\theta})$ (being a zero-mean Gaussian) has become popular (e.g. MacKay, 1991). Or in the case of Gaussian process classification (e.g. Williams, 1997), $\boldsymbol{\theta}$ is a latent function, $P(\boldsymbol{\theta})$ a Gaussian process distribution, and $P(t|\boldsymbol{x}, \boldsymbol{\theta})$ are simply models for the noise. Even if we do not have strong prior knowledge about $P(t|\boldsymbol{x})$, we can use the principle of *Occam's razor* (e.g. MacKay, 1991) and penalize complicated models by assigning low prior probability to them.[1] This is known as *regularization*. Both weight-decay and Gaussian process priors can be seen as regularization.

In order to arrive at a complete generation model, we also have to specify a model class $\{P(\boldsymbol{x}|\boldsymbol{\mu})\}$ and a prior $P(\boldsymbol{\mu})$. The Bayesian approach to discrimination is to *assume* that these settings specify how the data has been generated. Namely, we first sample $\boldsymbol{\theta} \sim P(\boldsymbol{\theta})$ and $\boldsymbol{\mu} \sim P(\boldsymbol{\mu})$, then independently (conditioned on $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$) $\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\mu})$, $t_i \sim P(t|\boldsymbol{x}_i, \boldsymbol{\theta})$, $i = 1, \ldots, n$, and $\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\mu})$, $i = n+1, \ldots, n+m$. Under this assumption, consistent inference is done by conditioning on the data, i.e. computing the *posterior* $P(\boldsymbol{\theta}|D_l, D_u)$, and prediction uses this "updated" belief via $P(t|\boldsymbol{x}, D_l, D_u) = \int P(t|\boldsymbol{x}, \boldsymbol{\theta})P(\boldsymbol{\theta}|D_l, D_u)\, d\boldsymbol{\theta}$.[2] This data generation model is shown in figure 1.

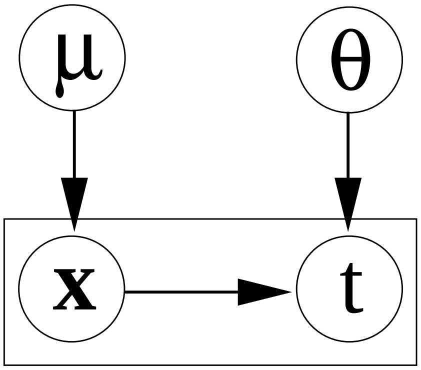

Figure 1. Standard data generation model

If this data generation assumption is correct, Bayesian prediction can be shown to be optimal, however, it remains a valid strategy even if the assumption is violated (or "partially correct", for example we could have $P(t|\boldsymbol{x}) = P(t|\boldsymbol{x}, \boldsymbol{\theta})$ for some $\boldsymbol{\theta}$ which has been sampled from a distribution different from the prior $P(\boldsymbol{\theta})$), and frequently outperforms other classification schemes on tasks where prior knowledge is available and can be encoded.

Under this model, $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are *a-priori independent*, i.e. $P(\boldsymbol{\theta}, \boldsymbol{\mu}) = P(\boldsymbol{\theta})P(\boldsymbol{\mu})$. The likelihood factors as

$$P(D_l, D_u|\boldsymbol{\theta}, \boldsymbol{\mu}) = P(T_l|X_l, \boldsymbol{\theta})P(X_l, D_u|\boldsymbol{\mu}),$$

which implies that $P(\boldsymbol{\theta}|D_l, D_u) \propto P(T_l|X_l, \boldsymbol{\theta})P(\boldsymbol{\theta})$, i.e. $P(\boldsymbol{\theta}|D_l, D_u) = P(\boldsymbol{\theta}|D_l)$, and $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are *a-posteriori independent*. Furthermore, $P(\boldsymbol{\theta}|D_l, \boldsymbol{\mu}) = P(\boldsymbol{\theta}|D_l)$. This means that neither knowledge of the unlabeled data D_u nor *any* knowledge of $\boldsymbol{\mu}$ changes the posterior belief $P(\boldsymbol{\theta}|D_l)$ of the labeled sample. Therefore, in the standard data generation model, unlabeled data cannot be used for Bayesian inference, and modelling the input distribution $P(\boldsymbol{x})$ is not necessary.

This fact is often seen as advantage of the standard model, since it implies that discrimination is *robust* w.r.t. assumptions of how the input data is distributed. However, it also means that we have to neglect unlabeled data D_u (even if available in great quantities) or available prior knowledge about $P(\boldsymbol{x})$, both of which might improve discrimination significantly (Blum & Mitchell, 1998; Nigam, McCallum, Thrun & Mitchell, 1998; Miller & Uyar, 1996). We also have to ask ourselves if we really *believe* in a prior independence of $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ for a given real-world task. Is it sensible to assume that knowledge about the input distribution does not influence the information we have (a-priori) about $P(t|\boldsymbol{x})$? As an example, suppose we want to regularize models $P(t|\boldsymbol{x}, \boldsymbol{\theta})$ according to their *smoothness* (e.g. the weight-decay prior). Is it sensible to enforce this requirement *globally*, i.e. to penalize a model for being rough in regions where examples $\boldsymbol{x}$ almost never fall into? We are on the safe side accepting this assumption, but also risk to ignore valueable information sources[3]. Furthermore, in certain experimental settings or learning tasks, this assumption is clearly violated (e.g. in the Co-Training setting, as discussed in section 5).

3. A modification to the standard data generation model

In section 2 we have motivated that treating $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$, i.e. the variables responsible for modelling $P(t|\boldsymbol{x})$ and $P(\boldsymbol{x})$, as a-priori independent might have drawbacks in many applications. The modification we suggest in this section is to simply drop this independence requirement. In other words, we construct a prior $P(\boldsymbol{\theta})$

[1] However, the notion of a "complicated model" frequently depends on what we know about the task.

[2] Note that the posterior $P(\boldsymbol{\mu}|D_l, D_u)$ is not required for prediction.

[3] The bare (empirical) fact that *unsupervised learning* techniques are often successful on real-world data indicates strongly that ignoring unlabeled data in classification might be suboptimal.

over $\boldsymbol{\theta}$ by choosing *conditional priors* $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ and then building the mixture

$$P(\boldsymbol{\theta}) = \int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})\, d\boldsymbol{\mu}. \qquad (1)$$

This way of construction $P(\boldsymbol{\theta})$ from distributions conditioned on the input distribution $\boldsymbol{\mu}$ is referred to as *input-dependent regularization.* The modified data generation model is shown in figure 2.

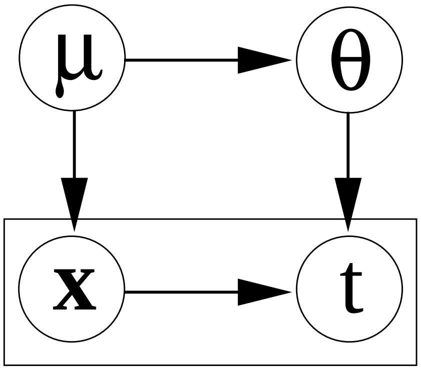

Figure 2. Modified data generation model in which $\boldsymbol{\theta}$ is allowed to depend on the input distribution $\boldsymbol{\mu}$.

The sampling process is modified in that we first sample $\boldsymbol{\mu} \sim P(\boldsymbol{\mu})$, then $\boldsymbol{\theta} \sim P(\boldsymbol{\theta}|\boldsymbol{\mu})$, i.e. conditioned on $\boldsymbol{\mu}$. Afterwards we sample independently (conditioned on $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$) $\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\mu}),\ t_i \sim P(t|\boldsymbol{x}_i, \boldsymbol{\theta}),\ i = 1, \dots, n,$ and $\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\mu}),\ i = n+1, \dots, n+m.$ It is obvious that (in general) under this generation model the posterior belief $P(\boldsymbol{\theta}|D_l, D_u)$ depends both on the unlabeled data D_u and on the prior $P(\boldsymbol{\mu})$. Note that the standard model of section 2 is a special case of the modified model.

Equation (1) can be seen as an instance of *hierarchical Bayesian design* (e.g. Berger, 1985). This technique allows us to create a prior which encodes complicated knowledge, by introducing new variables (called *hyperparameters* or *nuisance parameters*), then specifying prior distributions conditioned on the values of these parameters. Each of these conditional priors can be quite definitive, but if we place vague priors on the hyperparameters, the final marginal prior, obtained by integrating the hyperparameters out (as in (1)), will also be vague. Indeed we can regard $\boldsymbol{\mu}$ as nuisance parameter, since it is integrated out for prediction. However, direct evidence of $\boldsymbol{\mu}$ is available via D_u (and also X_l), while hyperparameters in hierarchical designs are usually buried high up in the hierarchy.

How can we (possibly) gain from information about $\boldsymbol{\mu}$, such as D_u? We have:

$$\begin{aligned} P(\boldsymbol{\theta}|D_l, D_u) &= \int P(\boldsymbol{\theta}, \boldsymbol{\mu}|D_l, D_u)\, d\boldsymbol{\mu} \\ &\propto \int P(T_l|X_l, \boldsymbol{\theta})P(X_l, D_u|\boldsymbol{\mu})P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})\, d\boldsymbol{\mu} \qquad (2) \\ &\propto P(T_l|X_l, \boldsymbol{\theta}) \int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu}|X_l, D_u)\, d\boldsymbol{\mu}. \end{aligned}$$

This should be compared to the posterior under the standard generation model, namely $P(\boldsymbol{\theta}|D_l, D_u) \propto P(T_l|X_l, \boldsymbol{\theta})P(\boldsymbol{\theta})$. If D_u is large, $P(\boldsymbol{\mu}|X_l, D_u)$ will be quite definitive (or peaked), i.e. the average over $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ in the last line of (2) will concentrate on a small region (for $\boldsymbol{\mu}$). Since the conditional $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ are usually more specific than the marginal $P(\boldsymbol{\theta})$, we see that the posterior belief in $\boldsymbol{\theta}$ should in general be narrower under the modified than under the standard model. An extreme case of this argument is analyzed in (Castelli & Cover, 1995). They make the strong assumptions that the input distribution is known completely *and* that all class-conditional distributions $P(\boldsymbol{x}|t)$ can be learned from unlabeled data only[4]. Thus, given an infinite amount of unlabeled data D_u, $P(\boldsymbol{\mu}|D_u)$ is a delta peak at $\hat{\boldsymbol{\mu}}$ (say), leading to $P(\boldsymbol{\theta}|D_l, D_u) \propto P(T_l|X_l, \boldsymbol{\theta})P(\boldsymbol{\theta}|\hat{\boldsymbol{\mu}})$. Now, only $c!$ values for $\boldsymbol{\theta}$ have non-zero probability under $P(\boldsymbol{\theta}|\hat{\boldsymbol{\mu}})$ (remember that the class-conditional distributions can be inferred exactly from $P(\boldsymbol{x}) = P(\boldsymbol{x}|\hat{\boldsymbol{\mu}})$, therefore only the assingment of these distributions to class labels remains to be done). In general, the gain on nontrivial tasks will be much less substantial, even if an unlimited number of unlabeled examples is available.

3.1 Why additional unlabeled data can hurt instead of help

It has been observed that using unlabeled data in addition to a set of labeled data occasionally *hurts* instead of being beneficial, w.r.t. generalization error (e.g. Zhang & Oles, 2000). In the context of this paper, we can motivate several possible reasons for such failures. First, the unlabeled data might have been used in an unfortunate way which is neither a Bayesian analysis nor a valid approximation to such, and therefore not in the scope of this paper. Second, established “black box algorithms” for supervised or unsupervised learning might have been used in a way which is not appropriate for the new “semi-supervised” problem. For example, the EM algorithm has frequently been used together with rather poor joint models for inputs and targets[5]. While such poor models are frequently

[4] The latter assumption is very strong, and we do not see a general way satisfy it in reality.

[5] A good example are *Naive Bayes* models.

well-suited and successful for classification based on labeled data, using them in an EM approach together with unlabeled data can be very problematic. A poor model, trained on a very small amount of labeled data, will usually confidently (but largely randomly!) label up the (potentially large) set of unlabeled data. In a few rounds, the EM algorithm will have converged into a poor local maximum of the joint likelihood which will often generalize worse than the initial model inferred from the labeled data only.

Third, the prior assumptions encoded via the structure of the model and the prior distributions might have been wrong for the problem at hand. This happens if the conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ enforce certain constraints very rigidly, and the true distribution $P(\boldsymbol{x}, t)$ happens to break some of them. In this case, the factor $\int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu}|X_l, D_u)\,d\boldsymbol{\mu}$ in (2) will assign very low probability to models $P(t|\boldsymbol{x}, \boldsymbol{\theta})$ close to the true $P(t|\boldsymbol{x})$, and if the labeled dataset D_l is rather small, the posterior $P(\boldsymbol{\theta}|D_l, D_u)$ will concentrate on a wrong region. This effect usually becomes stronger with growing D_u. In constrast to this, the standard model replaces this factor by $P(\boldsymbol{\theta})$ which is not affected by D_u. Since $P(\boldsymbol{\theta})$ is vague, but "on average" encodes a correct bias, as opposed to the systematically wrong one just described, predictions using the standard model can outperform input-dependent regularization.

Care must be taken towards these caveats when designing the conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$. While it is tempting (or maybe most feasible) to encode constraints rigidly, this should be done only if these are somewhat unquestionable. Since, via D_u, we obtain direct strong evidence about $\boldsymbol{\mu}$, we cannot rely on the fact that using a vague prior $P(\boldsymbol{\mu})$ results in a vague *marginal* $P(\boldsymbol{\theta})$. We also have to ensure that the *conditional* $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ are sufficiently vague w.r.t. unsure prior knowledge.

4. Examples and related work

In this section we argue that *Fisher kernel* discriminants (Jaakkola & Haussler, 1998) and *Co-Training* (Blum & Mitchell, 1998) can be seen as instances of input-dependent regularization.

Fisher kernels are covariance functions used in Gaussian process (or Support Vector machine) predictors, which are constructed based on a separate model $P(\boldsymbol{x}|\boldsymbol{\mu})$ of the input distribution $P(\boldsymbol{x})$, fitted to i.i.d. unlabeled data D_u. Specifying a covariance kernel is equivalent to specifying the geometry of the *feature space* in which kernel methods can be regarded as linear discriminants (however, the linear feature space induced by a kernel can be very complex, usually of very high or infinite dimension). Regularization of these machines works, in a nutshell, by penalizing discriminants by their squared norm in the feature space. Therefore, the Fisher kernel performs input-dependent regularization. More specifically, let K_μ be the Fisher kernel corresponding to the input distribution $\boldsymbol{\mu}$. Then, $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ is a zero-mean Gaussian process distribution with covariance function K_μ (recall that in Gaussian process classification, $\boldsymbol{\theta}$ is a function, and its prior is a distribution over functions). A full Bayesian analysis is intractable in this case, so that Fisher kernel discrimination usually works in two steps. First, we compute a model $\hat{\boldsymbol{\mu}}$ with maximum posterior probability $P(\boldsymbol{\mu}|D_u, X_l)$. We then approximate this posterior by the delta peak at $\hat{\boldsymbol{\mu}}$, which is reasonable if D_u is large. Using this approximation, the posterior in (2) becomes $\propto P(T_l|X_l, \boldsymbol{\theta})P(\boldsymbol{\theta}|\hat{\boldsymbol{\mu}})$. In a second step, we predict using this posterior, which usually involves further approximations[6].

The recently proposed *Co-Training* paradigm is an even more direct example of input-dependent regularization. In the original "noiseless" formulation, hard constraints on the target function are encoded in conditional priors, since these constraints depend on the input distribution. This view on Co-Training will be developed in section 5.

5. Co-Training as Bayesian inference

In this section, we show that *Co-Training* (Blum & Mitchell, 1998) can be seen as Bayesian inference under the modified generation model of section 3, using input-dependent regularization. The basic Co-Training algorithm is a "hard" variant of the *Expectation-Maximization (EM)* algorithm. We also propose a template EM algorithm for the modified generation model, which can be seen as generalization of Co-Training.

5.1 Co-Training and the notion of compatibility

For clarity, we will stick with the noiseless case discussed in (Blum & Mitchell, 1998). Let $\mathcal{X} = \mathcal{X}^{(1)} \times \mathcal{X}^{(2)}$ be the finite or countable input space. If $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$, the $\boldsymbol{x}^{(j)}$ should be regarded as different "views" on $\boldsymbol{x}$. For example, if $\boldsymbol{x}$ is a Web page, $\boldsymbol{x}^{(1)}$

[6] This is true for Gaussian process predictions using the Fisher kernel. Support Vector discrimination is a non-Bayesian technique which follows the paradigm of *Maximum Entropy discrimination* (Jaakkola, Meila & Jebara, 1999).

might be the text on the page, while $\boldsymbol{x}^{(2)}$ might be the text on hyperlinks referring to this page. We are also given spaces $\Theta^{(j)}$ of concepts $\boldsymbol{\theta}^{(j)}$, mapping $\mathcal{X}^{(j)}$ into $\{1,2\}$, $j=1,2,$[7] and set $\Theta = \Theta^{(1)} \times \Theta^{(2)}$. Elements $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) \in \Theta$ will be referred to as concepts over $\mathcal{X}$, although they are not in the strict sense, since usually $\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)}) \neq \boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})$ for some $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \mathcal{X}$. Whenever $\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)}) = \boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})$, we will write $\boldsymbol{\theta}(\boldsymbol{x}) = \boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)})$ for convenience. We assume that both classes $\Theta^{(j)}$ are learnable. If $A \subset \mathcal{X}$, we say that a concept $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})$ is *compatible* with A if $\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)}) = \boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})$ for all $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in A$. Denote by $\Theta(A)$ the space of all concepts compatible with A.[8] If $Q(\boldsymbol{x})$ is a distribution over $\mathcal{X}$ with *support* $S = \operatorname{supp} Q(\boldsymbol{x}) = \{\boldsymbol{x} | Q(\boldsymbol{x}) > 0\}$, we say that a concept $\boldsymbol{\theta}$ is *compatible* with the distribution Q if it is compatible with S.

In the Co-Training setting, there is an unknown input distribution $P(\boldsymbol{x})$. A *target concept* $\boldsymbol{\theta}$ is sampled from some unknown distribution over Θ, and the data distribution is $P(t|\boldsymbol{x}) = I_{\{\theta(\mathbf{x})=t\}}$ if $\boldsymbol{\theta} \in \Theta(\{\boldsymbol{x}\})$, $1/2$ otherwise[9]. However, the central assumption is that the target concept $\boldsymbol{\theta}$ is *compatible* with the input distribution $P(\boldsymbol{x})$. This implies that the target concept can be learned using only one of the views, i.e. from $D_l^{(j)} = \{(\boldsymbol{x}_i^{(j)}, t_i) | i = 1, \dots, n\}$ for one $j \in \{1,2\}$ only, if $\boldsymbol{x}_i = (\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)})$ and n is large enough. It also implies that we can make use of unlabeled data D_u, by observing that from $D_u \subset \operatorname{supp} P(\boldsymbol{x})$ it follows that the target concept must lie in $\Theta(\operatorname{supp} P(\boldsymbol{x})) \subset \Theta(D_u)$, which means that even *prior* to having seen any labeled data, we can shrink the effective concept space from Θ to $\Theta(D_u)$.

A simple sequential algorithm, described in subsection 5.3, can be used for the Co-Training setting. The basic idea is that we train two classifiers $\boldsymbol{\theta}^{(j)}$ in parallel, each of which only sees the $\mathcal{X}^{(j)}$ part of the input points. For each new unlabeled point, we produce a "pseudolabel" using *one* of the classifiers as predictor, then train the other one on the augmented dataset. Thus, the classifiers teach each other in turns, and this "switching" teacher-student relationship is backed by the compatibility assumption.

5.2 Co-Training as Bayesian inference

The compatibility assumption of subsection 5.1 is a prior assumption which can be encoded as follows. We model $P(\boldsymbol{x})$ by $\{P(\boldsymbol{x}|\boldsymbol{\mu})\}$ and a prior $P(\boldsymbol{\mu}) > 0$. For convenience, we introduce a further variable S which is deterministically related to $\boldsymbol{\mu}$ via $S = \operatorname{supp} P(\boldsymbol{x}|\boldsymbol{\mu})$, and we choose conditional priors $P(\boldsymbol{\theta}|S)$ as:

$$P(\boldsymbol{\theta}|S) = f_S(\boldsymbol{\theta}) I_{\{\theta \in \Theta(S)\}},\ S \subset \mathcal{X}, \tag{3}$$

where $f_S(\boldsymbol{\theta}) > 0$, and all $P(\boldsymbol{\theta}|S)$ are properly normalized. For example, if $\Theta(S)$ is finite, we can choose $f_S(\boldsymbol{\theta}) = |\Theta(S)|^{-1}$. As already mentioned above, we have a noiseless setting, i.e. $P(t|\boldsymbol{x}, \boldsymbol{\theta}) = (1/2)(I_{\{\theta^{(1)}(\mathbf{x}^{(1)})=t\}} + I_{\{\theta^{(2)}(\mathbf{x}^{(2)})=t\}})$. The data generation works as already described in section 3. First, we sample $\boldsymbol{\mu} \sim P(\boldsymbol{\mu})$ and set $S = \operatorname{supp} P(\boldsymbol{x}|\boldsymbol{\mu})$. Conditioned on S, we sample the target concept $\boldsymbol{\theta} \sim P(\boldsymbol{\theta}|S)$ (see (3)). Afterwards we sample independently (conditioned on $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$) $\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\mu})$, $t_i \sim P(t|\boldsymbol{x}_i, \boldsymbol{\theta})$, $i = 1, \dots, n$, and $\boldsymbol{x}_i \sim P(\boldsymbol{x}|\boldsymbol{\mu})$, $i = n+1, \dots, n+m$.[10]

From (2) we see that the posterior belief in $\boldsymbol{\theta}$ is

$$P(\boldsymbol{\theta}|D_l, D_u) \propto P(T_l|X_l, \boldsymbol{\theta}) \int P(\boldsymbol{\theta}|S) P(S|X_l, D_u)\, dS$$

$$= I_{\{\theta(\mathbf{x}_i)=t_i,\, i=1,\dots,n\}} \int P(\boldsymbol{\theta}|S) P(S|X_l, D_u)\, dS.$$

It is easy to see that $P(\boldsymbol{\theta}|D_l, D_u) \neq 0$ iff $\boldsymbol{\theta}$ is consistent with D_l and $\boldsymbol{\theta} \in \Theta(D_u \cup X_l)$. Namely, if $\boldsymbol{\theta} \notin \Theta(D_u \cup X_l)$, then $P(\boldsymbol{\theta}|S) = 0$ for all S which include D_u and X_l, and $P(S|D_u, X_l) = 0$ for all other S. On the other hand, if $\boldsymbol{\theta} \in \Theta(D_u \cup X_l)$, then we have $P(\boldsymbol{\theta}|\hat{S}) > 0$ and $P(\hat{S}|D_u, X_l) > 0$ at least for $\hat{S} = D_u \cup X_l$. Therefore, the set of all $\boldsymbol{\theta}$ for which $P(\boldsymbol{\theta}|D_l, D_u) > 0$ is identical to the remaining *version space*[11]. Co-Training, in the sense defined in (Blum & Mitchell, 1998), can therefore be seen as a way of updating a Bayesian posterior belief by conditioning on labeled and unlabeled data. Eventually, this procedure converges to a belief which allows only concepts which agree with the target concept on the support of $P(\boldsymbol{x})$, and Bayesian prediction on future $\boldsymbol{x} \sim P(\boldsymbol{x})$ coincides with the target concept.

If the data is not sufficient to pinpoint one concept, the concrete behaviour of Bayesian inference (with the generation model just described) depends on the bias induced by $f_S(\boldsymbol{\theta})$ in the conditional priors and

[7] For simplicity, we discuss two-class classification only.

[8] In order not to run into trivial problems, we assume that $\Theta(A)$ is never empty, which can be achieved by adding the constant concept 1 to both $\Theta^{(j)}$.

[9] Here, I_E is 1 if E is true, 0 otherwise. The scenario is called *noiseless* because the only source of randomness is the uncertainty in the target function.

[10] Note that $P(t|\boldsymbol{x}_i, \boldsymbol{\theta}) \neq 1/2$ since $\boldsymbol{\theta}$ is compatible with the support of $P(\boldsymbol{x})$ which includes $\boldsymbol{x}_i$.

[11] For a noiseless learning scenario, the *version space* is the set of all concepts which are consistent with all the observed data.

to some extent also on the prior $P(S)$, while a Co-Training algorithm depends on the biases used for learning in the spaces $\Theta^{(j)}$ (see subsection 5.3). Since both frameworks are quite general, it seems reasonable to state that (approximate) Bayesian inference with conditional priors based on the notion of *compatibility* between different *views* (representations) on examples and Co-Training are two sides of the same coin. This observation might have advantages for both fields. The idea of split representations, which has been proposed originally in the field of unsupervised learning (De Sa, 1993; Becker & Hinton, 1992) but has been transferred to the problem of "semi-supervised" learning in (Blum & Mitchell, 1998), might become a key technique for constructing conditional priors. On the other hand, the Bayesian view on Co-Training might help to deal with issues like noisy labels, learning biases and concept combination methods in a principled rather than heuristical way. For example, if we allow for noisy labels, the conditional priors based on the support (3) might be to rigid, in the sense discussed in subsection 3.1. More "careful" conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ could then be constructed as monotonic increasing functions of $Pr_{P(\mathbf{x}|\mu)}\{\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)}) = \boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})\}$, as mentioned in (Blum & Mitchell, 1998). If we do conditional density estimation rather than classification in the spaces $\Theta^{(j)}$, i.e. fit models $P(t|\boldsymbol{x}^{(j)}, \boldsymbol{\theta}^{(j)})$, even more interesting scores like $E_{P(\mathbf{x}|\mu)}[I(t^{(1)}, t^{(2)}|\boldsymbol{x})]$, $t^{(j)} \sim P(t|\boldsymbol{x}^{(j)}, \boldsymbol{\theta}^{(j)})$, $j = 1, 2$, could be used to construct $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ (see also Becker, 1992). Here, I denotes *mutual information*.

5.3 Co-Training as Expectation Maximization

In this subsection, we describe the Co-Training algorithm and show how it can be related to the powerful *Expectation-Maximization (EM)* algorithm (Dempster, Laird & Rubin, 1977).

The basic Co-Training algorithm proposed in (Blum & Mitchell, 1998) works as follows. Initialize the pair $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) \in \Theta$ by training on the labeled data D_l only. A growing working set is initialized with D_l. The algorithm is incremental, adding unlabeled points one by one. Each time a new point is added, one of the $\boldsymbol{\theta}^{(j)}$ is picked at random, the label of the new point is predicted using this concept, and the point together with the pseudolabel is added to the working set. Finally, both $\boldsymbol{\theta}^{(j)}$ are updated by retraining on the new working set[12]. This basic scheme is quite flexible, for example unlabeled points can be added in small batches rather than sequentially, or the order in which the points are added can be determined using heuristics. Furthermore, once all points of D_u have been added, $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ might not agree on points $\boldsymbol{x}$ in the test set, and other heuristics have to be used to combine them.

The EM algorithm is a general method for finding *Maximum Likelihood (ML)* or *Maximum A-Posteriori (MAP)* estimates in the presence of missing data. In the generalized formulation of (Hinton & Neal, 1997), we maintain a current estimate and a (completely factorized) distribution Q over the missing data, and iterate *E steps* (in which Q is updated) and *M steps* (in which we update the estimate). In a generally inferior *stochastic EM* algorithm (Celeux & Diebolt, 1985), instead of maintaining and propagating the Q distribution, we compute Q in each iteration, sample the hidden data from Q and use it to update the estimate. This variant cannot be considered to be an EM technique, since it does not come with the same convergence guarantees, however it is obviously related.

We are after a MAP estimate of $(\boldsymbol{\theta}, S)$ (recall that $S = \operatorname{supp} P(\boldsymbol{x}|\boldsymbol{\mu})$), and the missing data are the missing labels T_u. We choose a sequential variant of EM, in which the estimate is initialized by training on D_l only (S is initialized with X_l), and new points from D_u are added one at a time. This resembles Co-Training and seems reasonable in the light of subsection 3.1. In order not to get buried by notation, we will denote the unlabeled points *currently* used by the algorithm by X_u, i.e. $X_u = \{\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+s}\}$ in iteration s. Also $T_u = \{t_{n+1}, \ldots, t_{n+s}\}$, and t_{n+s} is added in iteration s.

At the beginning of iteration s, we add a new unlabeled point, which enlarges the "effective" missing data T_u. Next, we perform a *partial* E step to update the Q distribution, which amounts to computing[13] $Q(t_{n+s}) = P(t_{n+s}|D_l, X_u, \boldsymbol{\theta}) = P(t_{n+s}|\boldsymbol{x}_{n+s}, \boldsymbol{\theta})$ and sampling $t_{n+s} \sim Q(t_{n+s}) = (1/2)(I_{\{\theta^{(1)}(\mathbf{x}^{(1)}_{n+s})=t_{n+s}\}} + I_{\{\theta^{(2)}(\mathbf{x}^{(2)}_{n+s})=t_{n+s}\}})$. Note that Q remains the same on the other missing labels, and their "pseudolabel" values in T_u (sampled in earlier iterations) remain unchanged. This is equivalent to choosing one of the $\boldsymbol{\theta}^{(j)}$ at random and setting $t_{n+s} = \boldsymbol{\theta}^{(j)}(\boldsymbol{x}^{(j)}_{n+s})$. t_{n+s} is added to T_u. In the M step we update $(\boldsymbol{\theta}, S)$ so as to increase the posterior on the data (D_l, X_u, T_u). We

[12] A variant only retrains the one $\boldsymbol{\theta}^{(j)}$ which has *not* been used to label the new point. These two variants do not show significantly different behaviour.

[13] In the standard (stochastic) EM algorithm, we would have to update Q over *all* missing variables T_u, namely set it to $P(T_u|X_u, \boldsymbol{\theta})$. However, in the formulation of (Hinton & Neal, 1997) partial E *and* M steps are also allowed, possibly resulting in slower convergence.

do this by first setting[14] $S = X_u$, then retraining *both* $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ on the augmented data. It is obvious that this algorithm is equivalent to the basic Co-Training scheme described above.

A major benefit of this view on Co-Training is that we can easily generalize the basic scheme to more realistic settings, such as label noise or less rigid priors (see subsection 3.1), while remaining in the established frameworks of (approximate) Bayesian inference and Expectation-Maximization with their strong probabilistic, non-heuristic foundations. A step in this direction is done in the next subsection.

5.4 A Bayesian generalization of Co-Training

By generalizing the "hard" EM view on Co-Training, given in subsection 5.3, we can derive an EM algorithm to obtain a MAP estimate of $\boldsymbol{\theta}$ for the general data generation model of section 3. We are dealing with noisy labels, i.e. noise models $P(t|\boldsymbol{x}^{(j)}, \boldsymbol{\theta}^{(j)})$. These can be combined in an intuitive way as $P(t|\boldsymbol{x}, \boldsymbol{\theta}) \propto P(t|\boldsymbol{x}^{(1)}, \boldsymbol{\theta}^{(1)}) P(t|\boldsymbol{x}^{(2)}, \boldsymbol{\theta}^{(2)})$, which amounts to simply adding the log odds. For details on the following derivation, see (Jordan, Gharamani, Jaakkola & Saul, 1999). Using Jensen's inequality, we derive a variational lower bound of the log joint[15] w.r.t. T_u *and* $\boldsymbol{\mu}$ as follows:

$$\log P(D_l, X_u, \boldsymbol{\theta}) = \log \sum_{T_u} \int P(T_l, T_u, X_l, X_u, \boldsymbol{\theta}, \boldsymbol{\mu})\, d\boldsymbol{\mu}$$
$$\geq \sum_{T_u} \int Q(T_u, \boldsymbol{\mu}) \log \frac{P(T_l, T_u, X_l, X_u, \boldsymbol{\theta}, \boldsymbol{\mu})}{Q(T_u, \boldsymbol{\mu})}\, d\boldsymbol{\mu}. \tag{4}$$

It is easy to see that the distribution which maximizes the bound is given by $Q(T_u, \boldsymbol{\mu}) = Q(T_u) Q(\boldsymbol{\mu})$, $Q(T_u) = P(T_u|X_u, \boldsymbol{\theta})$, $Q(\boldsymbol{\mu}) \propto P(\boldsymbol{\theta}|\boldsymbol{\mu}) P(X_l, X_u|\boldsymbol{\mu}) P(\boldsymbol{\mu})$. The maximizer for $Q(\boldsymbol{\mu})$ is intractable in general, but we can choose the best variational distribution from a tractable family (e.g. the Gaussian family), by maximizing the relevant part of the lower bound,

$$\int Q(\boldsymbol{\mu}) \log \frac{P(\boldsymbol{\theta}|\boldsymbol{\mu}) P(X_l, X_u|\boldsymbol{\mu}) P(\boldsymbol{\mu})}{Q(\boldsymbol{\mu})}\, d\boldsymbol{\mu}, \tag{5}$$

w.r.t. $Q(\boldsymbol{\mu})$ from this family only. We follow the scheme and the notations of subsection 5.3 and choose a sequential variant of EM with partial Q updates in the E steps (note that here, the Q distribution over the missing variables consists of the product of $Q(T_u)$ *and* $Q(\boldsymbol{\mu})$). Again, the estimate $\boldsymbol{\theta}$ is initialized by training on D_l only. An important difference to the algorithm in subsection 5.3 is that here the variables in T_u *remain* hidden, with our uncertainty about them encoded in $Q(T_u)$, they are *not* fixed to "pseudolabel" values. Iteration s of our algorithm consists of:

1. Add $\boldsymbol{x}_{n+s}$ to X_u, t_{n+s} to T_u. Update $Q(T_u)$ partially by setting $Q(t_{n+s}) = P(t_{n+s}|\boldsymbol{x}_{n+s}, \boldsymbol{\theta})$ and leaving it unchanged on the other variables.
2. Update $Q(\boldsymbol{\mu})$ by maximizing (5) w.r.t. $Q(\boldsymbol{\mu})$ within a fixed family.
3. Update the estimate $\boldsymbol{\theta}$ by maximizing the lower bound (4) for fixed Q. This is tractable since the family for $Q(\boldsymbol{\mu})$ has been chosen accordingly.

Even if applied to the Co-Training setting, there are several differences between this EM algorithm and the basic Co-Training procedure. First, Co-Training resembles stochastic EM by simply sampling and filling in missing labels, while EM maintains and propagates (via Q) a "soft" distribution over these labels. Second, in EM *both* $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ are combined in the E step to update the missing label uncertainties $Q(t_{n+s})$, while Co-Training chooses one of them at random, which then determines $Q(t_{n+s})$ alone. As for the EM algorithm, the high flexibility of the probabilistic setting immediately suggests a host of variations. For example, we could update larger parts of (or the complete) $Q(T_u)$ in the E steps, thus possibly refining incorrect earlier uncertainty estimates. The order in which new points are added could be determined using greedy selection with probabilistic criteria such as the entropy of $Q(t_{n+i}) = P(t_{n+i}|\boldsymbol{x}_{n+i}, \boldsymbol{\theta})$ for candidate points $\boldsymbol{x}_{n+i}$ not yet included in X_u. Finally, we could even try to obtain a more accurate approximation to $P(\boldsymbol{\theta}|D_l, D_u)$ than a MAP one, by employing Variational Bayesian techniques (Attias, 1999). The validity of such approaches will have to be tested carefully in comparisons on real-world tasks, since there is always the possibility that greater flexibility and power comes with a lack of robustness (see subsection 3.1).

6. Conclusions

We have given a detailed discussion of the standard Bayesian data generation model for discriminative architectures and shown that unlabeled data or side information about the input distribution cannot be used for inference. A simple modification of the generation model was proposed which gives us the neces-

[14] This choice surely increases the posterior. The old value for S does not include $\boldsymbol{x}_{n+s}$ and therefore gives rise to posterior probability 0 once $\boldsymbol{x}_{n+s}$ is added.

[15] Maximizing the log posterior w.r.t. $\boldsymbol{\theta}$ is equivalent to maximizing the log joint.

sary flexibility to explore "semi-supervised" learning in the Bayesian discriminative context. By constructing conditional priors which perform input-dependent regularization, information about the input distribution can be used to guide inference and prediction. A particularly clear instance of this, namely Co-Training, has been discussed in detail. Finally, we have proposed a template EM algorithm for MAP estimation within the modified generation model, a special case of which can be regarded as a generalization of Co-Training.

This paper provides a clarifying overview and gives theoretical and algorithmic ideas, however, in the present form, lacks backing by experimental results. To round it up by providing such is a pressing issue (note that many of the methods tested in (Nigam & Ghani, 2000) are also cases of the framework developped here). Furthermore it will be interesting to compare this framework to other general methods for the "semi-supervised" problem. In the long term, finding general methods to construct meaningful, yet tractable conditional priors (such as the method of multiple views and compatibility, as exploited by Co-Training) and developping algorithms for approximate Bayesian inference using these priors, are important topics for future work.

Acknowledgements

We thank Chris Williams, Ralf Herbrich and Hugo Zaragoza (the "Eagles") for helpful discussions. The author gratefully acknowledges support through a research studentship from *Microsoft Research Ltd.*

References

Attias, H. (1999). A Variational Bayesian framework for graphical models. *Advances in NIPS 12*. MIT Press.

Becker, S., & Hinton, G. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *355*, 161–163.

Berger, J. (1985). Statistical decision theory and Bayesian analysis. 2nd edition. Springer.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with Co-Training. *Proceedings of COLT.*

Castelli, V., & Cover, T. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters*, *16*, 105–111.

Celeux, G., & Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quaterly*, *2*, 73–82.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*, 1–38.

De Sa, V. (1993). Learning classification with unlabeled data. *Advances in NIPS 6*. Morgan Kaufmann.

Hinton, G., & Neal, R. (1997). A new view on the EM algorithm that justifies incremental and other variants. In M. Jordan (Ed.), *Learning in Graphical Models*. Kluwer.

Jaakkola, T., Meila, M., & Jebara, T. (1999). Maximum Entropy Discrimination. *Advances in NIPS 12*. MIT Press.

Jordan, M., Gharamani, Z., Jaakkola, T., & Saul, L. (1999). An introduction to variational methods for graphical models. *Machine Learning*, *37*, 193–233.

MacKay, D. (1991). Bayesian Methods for Adaptive Models. PhD thesis, California Institute of Technology.

Miller, D., & Uyar, H. (1996). A Mixture of Experts classifier with learning based on both labelled and unlabelled data. *Advances in NIPS 9*, 571–577. MIT Press.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Text classification from labeled and unlabeled documents using EM. *Proceedings of AAAI.*

Nigam, K., & Ghani, R. (2000). Understanding the behaviour of Co-Training. *KDD Workshop on Text Mining.*

Williams, C. K. I. (1997). Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. Jordan (Ed.), *Learning in Graphical Models*. Kluwer.

Zhang, T., & Oles, F. (2000). A probability analysis on the value of unlabeled data for classification problems. *Proceedings of ICML.*