A Taxonomy for Semi-Supervised Learning Methods

Matthias Seeger
Department of EECS
University of California at Berkeley
485 Soda Hall, Berkeley CA
mseeger@cs.berkeley.edu

September 20, 2009

Abstract

We propose a simple taxonomy of probabilistic graphical models for the semi-supervised learning problem. We give some broad classes of algorithms for each of the families and point to specific realizations in the literature. Finally, we shed more detailed light on the family of methods using input-dependent regularization (or conditional prior distributions) and show parallels to the Co-training paradigm.

1 The Semi-Supervised Learning Problem

The *semi-supervised learning (SSL)* problem has recently drawn large attention in the machine learning community, mainly due to its significant importance in practical applications. In this Section we define the problem and introduce the notation to be used in the rest of this Chapter.

In statistical machine learning, we distinguish between *unsupervised* and *supervised learning*. In the former scenario we are given a sample $\{x_i\}$ of patterns in $\mathcal{X}$ drawn *independently and identically distributed (i.i.d.)* from some unknown data distribution with density $P(x)$. Our goal is to estimate the density or a (known) functional thereof. Supervised learning consists of estimating a functional relationship $x \to y$ between a covariate $x \in \mathcal{X}$ and a class variable[1] $y \in \{1,\ldots,M\}$, with the goal of minimizing a functional of the (joint) data distribution $P(x,y)$ such as the probability of classification error. The marginal data distribution $P(x)$ is referred to as input distribution. Classification can be treated as a special case of estimating the joint density $P(x,y)$, but this is wasteful since x will always be given at prediction time, so there is no need to estimate the input distribution.

The terminology "unsupervised learning" is a bit unfortunate, the term *density estimation* should probably be preferred. Traditionally, many techniques for density estimation propose a latent (unobserved) class variable y and estimate $P(x)$ as *mixture distribution* $\sum_{y=1}^{M} P(x|y)P(y)$. Note that y has a fundamentally different role than in classification, in that its existence and range c is a modeling choice rather than observable reality. However, in other density estimation techniques, such as nonlinear dimensionality reduction, the term "unsupervised" does not make sense.

[1] We restrict ourselves to classification scenarios in this Chapter.

The semi-supervised learning problem belongs to the supervised category, since the goal is to minimize the classification error, and an estimate of $P(x)$ is not sought after.[2] The difference to a standard classification setting is that along with a *labeled sample* $D_l = \{(x_i, y_i) \mid i = 1, \ldots, n\}$ drawn i.i.d. from $P(x,y)$ we also have access to an additional *unlabeled sample* $D_u = \{x_{n+j} \mid j = 1, \ldots, m\}$ from the marginal $P(x)$. We are especially interested in cases where $m \gg n$ which may arise in situations where obtaining an unlabeled sample is cheap and easy, while labeling the sample is expensive or difficult. We denote $X_l = (x_1, \ldots, x_n), Y_l = (y_1, \ldots, y_n)$ and $X_u = (x_{n+1}, \ldots, x_{n+m})$. The unobserved labels are denoted $Y_u = (y_{n+1}, \ldots, y_{n+m})$. In a straightforward generalization of SSL (not discussed here) uncertain information about Y_u is available.

There are two obvious baseline methods for SSL. We can treat it as a supervised classification problem by ignoring D_u, or we can treat y as latent class variable in a mixture estimate of $P(x)$ which is fitted using an unsupervised method, then associate latent groups with observed classes using D_l (see Section 3.1 for more details). One would agree that any valid SSL technique should outperform both baseline methods significantly in a range of practically relevant situations. If this sounds rather vague, note that in general for a *fixed* SSL method it should be easy to construct data distributions for which either of the baseline methods does better.[3] In our view, SSL is much more a practical than a theoretical problem. A useful SSL technique should be configurable to specifics of the task in a similar way than Bayesian learning, through the choice of prior and model. While some theoretical work has been done for SSL, the bulk of relevant work so far has tackled real-world applications.

2 Paradigms for Semi-Supervised Learning

Since SSL methods are supervised learning techniques, they can be classified according to the standard taxonomy into *generative* and *diagnostic* paradigms. In this Section we present these paradigms and highlight their differences in the case of SSL. We also note that this taxonomy which originated for purely supervised methods, can be ambiguous when applied to SSL, and we suggest how the borderline can be drawn exactly.

In the figures of this Section, we employ a convenient graphical notation frequently used in statistics and machine learning [17, 16]. These so-called directed *graphical models* (or indepence diagrams) have the following intuitive semantics. Nodes represent random variables. The parents of a node i are the nodes j for which a directed edge $j \to i$ exists.[4] It is possible to sample the value of a node once the values of all its parents are known. Thus, a graphical model is a simple way of representing the sampling mechanism from a distribution over several variables. As such, the graphical model encodes conditional independency constraints that have to hold for the distribution. In order to sample from the distribution, we start with nodes without parents and work in the directions of the edges.

[2] While this statement is probably open to debate, it is in fact agreed upon in statistics. In our opinion, methods should be classified foremost according to the problem they try to solve, *not* by which sources of data they make use of. On the other hand, there are problems in which density estimation is the goal and labeled data is treated as auxiliary source. However, these fall into a category with very different characteristics and are not in the scope of this Chapter. In our opinion, it would be very confusing to lump them together with methods we classify as SSL here. A label like "semi-unsupervised learning" would be more appropriate.

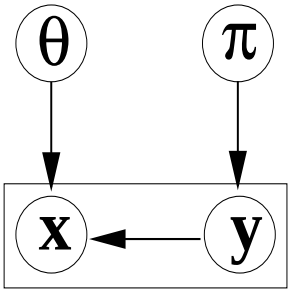
[3] This is a "no free lunch" statement for SSL, but in practice it seems to be a more serious problem than in the purely supervised context (where a "no free lunch" statement holds as well). See Chap.??? in this volume for some examples.

[4] Directed cycles are not allowed. In other words, it must be impossible to return to a node by moving along edges and respecting their direction.

We also make use of *plates* which are rectangular boxes grouping a set of nodes. This means that the group is sampled repeatedly and independently from the same distribution (i.i.d.) conditioned on all nodes which are parents of any plate member. For example, the figure of Section 2.1 means that we first sample θ and π independently (neither has parents), then draw a sample $\{(x_i, y_i)\}$ i.i.d. conditioned on θ, π (which are parents of the plate).

Note that we describe the generative and diagnostic paradigm from an explicitly Bayesian viewpoint. This is somewhat a matter of personal choice here, and certainly one could sketch these classes without ever mentioning concepts like prior distributions. On the other hand, the Bayesian view avoids many unnecessary complications, in that all variables are random, no difference has to be made between functional and probabilistic independence, *etc*, so we do not think our presentation lacks clarity or generality because of this choice.

2.1 The Generative Paradigm



We refer to architectures following the *generative paradigm* as *generative methods*. Within such, we model the class distributions $P(x|y)$ using model families $\{P(x|y,\theta)\}$, furthermore the class priors $P(y)$ by $\pi_y = P(y|\pi)$, $\pi = (\pi_y)_y$. We refer to an architecture of this type as a *joint density model*, since we are modeling the full joint density $P(x,y)$ by $\pi_y P(x|y,\theta)$. For any fixed $\hat{\theta}, \hat{\pi}$, an estimate of $P(y|x)$ can then be computed by Bayes' formula:

$$P(y|x,\hat{\theta},\hat{\pi}) = \frac{\hat{\pi}_y P(x|y,\hat{\theta})}{\sum_{y'=1}^{M} \hat{\pi}_{y'} P(x|y',\hat{\theta})}.$$

This is sometimes referred to as *plug-in estimate*. Alternatively, one can obtain the Bayesian predictive distribution $P(y|x,D_l)$ by averaging $P(y|x,\theta,\pi)$ over the posterior $P(\theta,\pi|D_l)$.[5] Within the generative paradigm, a model for the marginal $P(x)$ emerges naturally as

$$P(x|\theta,\pi) = \sum_{y=1}^{M} \pi_y P(x|y,\theta).$$

If labeled and unlabeled data are available, a natural criterion emerges as the *joint log likelihood* of both D_l *and* D_u,

$$\sum_{i=1}^{n} \log \pi_{y_i} P(x_i|y_i,\theta) + \sum_{i=n+1}^{n+m} \log \sum_{y=1}^{M} \pi_y P(x_i|y,\theta), \tag{1}$$

or alternatively the posterior $P(\theta,\pi|D_l,D_u)$.[6] This is essentially an issue of maximum likelihood in the presence of missing data (treating y as latent variable), which can in principle be attacked by the expectation-maximization (EM) algorithm (see Section 3.1) or by direct gradient descent.

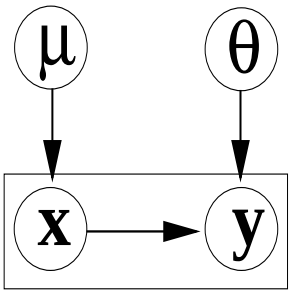
Some researchers have been quick in hailing this strategy as an obvious solution to the SSL problem, but this is not the case, in about the same sense as generative methods often do not provide

[5] In a sense, the predictive distribution is a Bayesian's best estimate of the underlying true data distribution $P(y|x)$. It is however obtained as posterior expectation, not by maximizing some criterion.

[6] To predict, we average $P(y|x,\theta,\pi)$ over the posterior. If we know that x is drawn from $P(x)$ and independent from D, we should rather employ the posterior $P(\theta,\pi|D_l,D_u,x)$. However, in this case the test set usually forms a part of D_u, and the two posteriors are the same.

good solutions to classification problems. Generative techniques provide an estimate of $P(x)$ along the way, although this is not required for classification, and in general this proves wasteful given limited data. For example, maximizing the joint likelihood of a finite sample need not lead to small classification error, because depending on the model it may be possible to increase the likelihood more by improving the fit of $P(x)$ than the fit of $P(y|x)$. This is an instance of the general problem of balancing the impact of D_l and D_u on the final predictions, especially in the case $m \gg n$. This issue is discussed in Section 3.1. Furthermore, in the SSL setting y is a latent variable which has to be summed out on D_u, leading to highly multimodal posteriors, so that likelihood or posterior maximization techniques are plagued by the presence of very many (local) minima.

2.2 The Diagnostic Paradigm



In *diagnostic methods*, we model the conditional distribution $P(y|x)$ directly using the family $\{P(y|x,\theta)\}$. To arrive at a complete sampling model for the data, we also have to model $P(x)$ by a family $P(x|\mu)$, however if we are only interested in updating our belief in θ or in predicting y on unseen points, this is not necessary, as we will see next. Under this model, θ and μ are *a priori independent*, i.e. $P(\theta,\mu) = P(\theta)P(\mu)$.

The likelihood factors as

$$P(D_l, D_u|\theta,\mu) = P(Y_l|X_l,\theta)P(X_l,D_u|\mu),$$

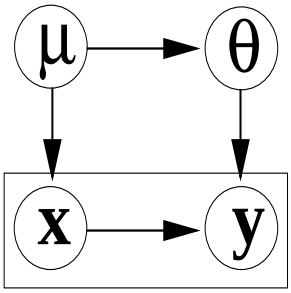
which implies that $P(\theta|D_l,D_u) \propto P(Y_l|X_l,\theta)P(\theta)$, i.e. $P(\theta|D_l,D_u) = P(\theta|D_l)$, and θ and μ are *a-posteriori independent*. Furthermore, $P(\theta|D_l,\mu) = P(\theta|D_l)$. This means that neither knowledge of the unlabeled data D_u nor *any* knowledge of μ changes the posterior belief $P(\theta|D_l)$ of the labeled sample. Therefore, in the standard data generation model for diagnostic methods, unlabeled data D_u cannot be used for Bayesian inference, and modelling the input distribution $P(x)$ is not necessary. There are non-Bayesian diagnostic techniques in which we can make use of D_u (see Section 3.2), but the impact of doing so (as opposed to ignoring D_u) is usually very limited. In order to make significant use of unlabeled data in diagnostic methods, the data generation model discussed above has to be modified as discussed in the following Section.

2.3 Regularization depending on the Input Distribution

When learning from a sample D_l of limited size, typically very many associations $x \to y$ are consistent with the data. The idea of *regularization* is to bias our choice of classifier towards "simpler" hypotheses, by adding a regularization functional to the criterion to be minimized which grows with complexity. Here, the notion of simplicity depends on the task and the model setup. For example, for a linear model it is customary to penalize a norm of the weight vector, and for some commonly used regularization functionals this can be shown to be equivalent to placing a zero-mean prior distribution on the weight vector. From now on we will only be interested in regularization by priors and will use the terms interchangeably.

We have seen in Section 2.2 that with straight diagnostic Bayesian methods for classification, we cannot make use of additional unlabeled data D_u, because θ (parameterizing $P(y|x)$) and μ (parameterizing $P(x)$) are *a priori* independent. In other words, the model family $\{P(y|x,\theta)\}$ is regularized *independently* of the input distribution.

If we allow prior dependencies between θ and μ, *e.g.* $P(\theta,\mu) = P(\theta|\mu)P(\mu)$ and $P(\theta) = \int P(\theta|\mu)P(\mu)\,d\mu$ (as shown in the independence diagram to the right), the situation is different. The conditional prior $P(\theta|\mu)$ in principle allows information about μ to be transferred to θ. In general, θ and D_u will be dependent given the labeled data D_l, therefore unlabeled data can change our posterior belief in θ.



We conclude that to make use of additional unlabeled data within the context of diagnostic Bayesian supervised techniques, we have to allow an *a priori* dependence between the latent function representing the conditional probability and the input probability itself. In other words, we have to use a *regularization of the latent function which depends on the input distribution*. The potential gain can be demonstrated by the following argument. Note that conditional priors imply a marginal prior $P(\theta)$ which is a mixture distribution: $P(\theta) = \int P(\theta|\mu)P(\mu)\,d\mu$. By conditioning on the unlabeled data, this is replaced by $P(\theta|D_u) = \int P(\theta|\mu)P(\mu|D_u)\,d\mu$ which can have a much smaller entropy than $P(\theta)$, implying that the posterior belief $P(\theta|D_l,D_u)$ can be much narrower than $P(\theta|D_l)$. On the other hand, the same argument can be used to demonstrate that using additional unlabeled data D_u can hurt instead of help. Namely, if the priors $P(\theta|\mu)$ enforce certain constraints very rigidly, but these happen to be violated in the true distribution $P(x,y)$, the conditional "prior" $P(\theta|D_u)$ will assign much lower probability than $P(\theta)$ to models $P(y|x,\theta)$ close to the truth, and the posterior $P(\theta|D_l,D_u)$ can be concentrated around suboptimal models. While it is certainly easy to construct artificial situations where additional unlabeled data hurts, it is worrying that such failures do happen quite unexpectedly in practically relevant settings as well. For a more thorough analysis of this problem, see Cozman and Cohen (Chap.??? in this volume).

We note that while the modification to the standard data generation model for diagnostic methods suggested here is straightforward, choosing appropriate conditional priors $P(\theta|\mu)$ suitable for a task at hand can be challenging. However, several general techniques for SSL can actually be seen as realizing input-dependent regularization, as is demonstrated in Section 3.3.

The reader may feel uneasy at this point. If we use *a priori* dependent θ and μ, the final predictive distribution *depends* on the prior $P(\mu)$ over the input distribution. This forces us to model the input distribution itself, in contrast to the situation for standard diagnostic methods. In this case, will our method still be a diagnostic one? Is it not the case that any method which models $P(x)$ in some way, must automatically be generative? Diagnostic methods can be much more parsimonious simply because $P(x)$ need not be estimated. In order to implement input-dependent regularization, do we have to use a generative model with the drawbacks discussed in Section 2.1? There is indeed some ambiguity here, but we will try to clarify this point in Section 2.4. Under this general viewpoint, input-dependent regularization is indeed a diagnostic SSL technique.

In the diagnostic paradigm for purely supervised tasks, θ and μ are treated as *a priori* independent, leading to the fact that no aspects of $P(x)$ have to be estimated. While this is convenient, it is not clear whether we should really believe in such independence for a real-world task. For example, suppose that $P(\theta)$ enforces smoothness of the relationship $P(y|x,\theta)$. Is it sensible to enforce smoothness of $x \to y$ around all x, or should we not rather penalize rough behaviour only where $P(x)$ has significant volume? The former is more conservative and possibly more robust, but also risks ignoring valuable information sources (see Section 3.3.1 for an example).

2.4 The Borderline between the Paradigms

While the borderline between supervised and unsupervised methods is clearly drawn, the distinction between generative and diagnostic techniques can be ambiguous, especially if we apply this taxonomy to SSL. In this Section we give two criteria for a clear discrimination: a simple and a more elaborate one. In a sense they are both based on the same issue, namely the *role* that the $P(x)$ estimate plays for the prediction.

Recall that we restrict ourselves to methods whose ultimate goal it is to estimate $P(y|x)$. Traditionally, generative methods achieve this by modelling the joint distribution $P(y,x)$ and fit this model to data by capturing characteristics of the true joint data distribution. An estimate of $P(x)$ can always be obtained by marginalizing the joint estimate. In contrast, diagnostic methods concentrate on modelling the conditional distribution $P(y|x)$ only, and an estimate of $P(x)$ cannot be extracted. However, in the SSL case we do have to model $P(x)$ in order to profit from D_u. So are all SSL methods generative? We argue against this viewpoint and try to classify SSL techniques according to the role which the $P(x)$ estimate actually plays.

While it is true that any SSL method has to model $P(x)$ in some way, in a generative technique we model the class-conditional distributions $P(x|y)$ explicitly, so that the model for $P(x)$ is a mixture of those. From these estimates (and the estimates of $P(y)$) we obtain an estimate of $P(y|x)$ using Bayes' formula. Characteristics of the predictive estimate (such as the function class in a parametric situation) depend entirely on the class-conditional models. For example, if the latter are Gaussian with the same covariance matrix, the predictive estimates will be based on linear functions. In a nutshell, we specify the $P(x|y)$ using our modelling toolbox, which implies the form of our $P(y|x)$ and $P(x)$ estimates (the latter is a mixture of the $P(x|y)$). The only way to encode *specific* properties for the latter estimates is to find $P(x|y)$ candidates which are both tractable to work with and imply the desired properties of $P(y|x)$ and $P(x)$. In contrast to that, in a diagnostic method we model $P(y|x)$ directly, and also typically have considerable freedom in modelling $P(x)$. In SSL we regularize the $P(y|x)$ estimates using information from $P(x)$, but we do not have to specify the class-conditional distributions explicitly.[7] While this definition is workable for the SSL methods mentioned here, it may be too restrictive on the generative side. For example, the "many-centers-per-class" model of Section 3.1 is clearly generative, but works with a mixture model for $P(x)$ which has several components for each class y, and $P(x|y)$ is modelled indirectly via $P(x|y) = \sum_k \pi_y \beta_{y,k} P(x|k)$, *i.e.* as a mixture itself. In the following paragraph we suggest an alternative view which leaves more freedom for generative techiques.

The practical success of SSL has shown that unlabeled data, *i.e.* knowledge about $P(x)$, can be useful for supervised tasks, but it is not necessarily the same *type* of knowledge that would lead to a good estimate of $P(x)$ according to common performance criteria for density estimation. In fact, it is actually a few general characteristics of $P(x)$ which seem to help classification (see for example Section 3.3.1). For example, if we convert a purely diagnostic technique such as SVM or logistic regression into an SSL technique by employing a regularizer penalizing $P(y|x)$ estimates which violate *certain aspects* of $P(x)$ such as the cluster assumption (see Section 3.3.1), the influence of $P(x)$ on the final $P(y|x)$ estimate is restricted to just these aspects that we hope are important for better classification. These restrictions are engineered by us because we want to make best use of D_u *in order to predict* $P(y|x)$. In contrast if we perform SSL by maximizing a suitably reweighted

[7] There are of course class-conditional distributions which are *implied* by the models of $P(y|x)$ and $P(x)$ (use Bayes' formula), but importantly we do not have to work with them directly, so that their form is not restricted by tractability requirements.

version of the joint log likelihood (Eq. 1) of a mixture model (see Section 3.1), such a restriction to classification-relevant aspects is not given or at least not directly planned. In fact the joint model is designed in much the same way as we would do for density estimation.

For example, consider the framework of conditional priors of Section 2.3. While it is essential to learn about $P(x)$ in SSL, the impact of an oversimple model for $P(x)$ on the final prediction is much less severe than in density estimation. This is because a suitable regularization will only depend on certain aspects of $P(x)$ (for example on the coarse locations of high density regions under the cluster assumption, see Section 3.3.1), and our model for the x distribution only has to be able to capture those accurately.

3 Examples

In this Section we provide examples of SSL methods falling in each of the categories introduced in the previous Section. We do not try to provide a comprehensive literature review here (see [22] for review of work up to about 2001), but are selective in order to point out characteristics of and differences between the categories. Note that in this context (and also in [22]) some methods are classified as "baseline methods". This does not constitute a devaluation, and in fact some of these methods belong to the top-performers on some tasks. Furthermore, we think that theoretical analyses of such methods are of great value, not least because many practitioners use them. Our label applies to methods which can be derived fairly straightforwardly from standard unsupervised or supervised methods, and we hope that truely novel proposals are in fact compared against the most closely related baseline methods.

3.1 Generative Techniques

Recall from Section 2.1 that generative techniques use a model family $\{P(x,y|\theta,\pi)\}$ in order to model the joint data distribution $P(x,y)$. The simplest idea is to run a mixture density estimation method for $P(x)$ on $X_l \cup X_u$, treating y as a latent class variable, then using the labeled sample D_l in order to associate latent classes with actual ones. An obvious problem with this approach is that the labeling provided by the unsupervised method may be inconsistent with D_l, in which case the clustering should be modified to achieve consistency with D_l. Castelli and Cover [6] provide a simple analysis of this baseline method under fairly unrealistic identifiability conditions. Namely, they assume that the data distribution is exactly identifiable by the unsupervised method at hand, which employs a mixture model with one component for each class. It is not clear how to achieve this in practice, even if $P(x)$ is exactly known.[8] In the large-sample limit, all class distributions can be learned perfectly, but the assignment of classes to label names obviously remains completely open. However, only a few additional labeled points are required in order to learn this assignment. In fact, it is easy to see that the error rate converges to the Bayes error exponentially fast (in the number of labeled examples drawn from $P(x,y)$).

Another baseline method consists of maximizing the joint likelihood of Eq. 1. For $m > 0$, the criterion to be minimized is not convex and typically multimodal, so we have to contend ourselves with

[8] It is not unrealistic to assume that $P(x)$ is exactly known, or that $m \to \infty$. The problem is that they assume that if $P(x)$ is viewed as mixture distribution, then the model can fit the class distributions $P(x|y)$ exactly. This is not realistic for real-world problems, especially if the quantities of interest are simply good estimates of $P(y|x)$ or a small generalization error of the resulting classifier.

finding a local maximum. This can be done by direct gradient descent or more conveniently by applying the *expectation maximization (EM)* algorithm [11]. The latter is an iterative procedure which is guaranteed to converge to a local maximum of the likelihood. If all data in Eq. 1 were labeled, a local maximum would be found by a single optimization over θ. In fact, if the class-conditional distributions $P(x|y,\theta)$ are from an exponential family, the global maximum can be found analytically. EM works by assigning label distributions $q(y|x_i)$ to all points x_i. For a labeled point, the label is represented in that $q(y|x_i) = \delta_{y,y_i}$. If x_i is unlabeled, we use the conditional posterior (for the current θ), *i.e.* $q(y|x_i) \propto \pi_y P(x_i|y,\theta)$. Intuitively, this choice reflects our best current point estimate for the label of x_i. The E step in EM consists of computing $q(y|x_i)$ for all points. In the M step, the parameters θ, π are updated by maximizing the expected log likelihood under the q distributions:

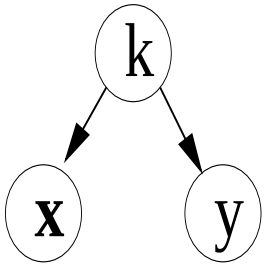
$$\phi(\theta', \pi') = \sum_{i=1}^{n+m} \sum_{y=1}^{M} q(y|x_i) \log \pi'_y P(x_i|y,\theta').$$

E and M steps are iterated until convergence. It is easy to show that ϕ is a lower bound on the joint log likelihood Eq. 1 for *any* choice of q on the unlabeled points. The bound becomes an equality if the q are chosen as posteriors and the parameters θ, π are not changed. Furthermore, under this choice the gradient of lower bound and joint log likelihood are the same at θ, π, so that if EM converges we have found a local maximum of Eq. 1.

The idea of using EM on a joint generative model to train on labeled and unlabeled data is almost as old as EM itself. Titterington *et.al.* ([27], Sect. 5.7) review early theoretical work on the problem of discriminant analysis in the presence of additional unlabeled data. The most common assumption is that the data has been generated from a mixture of two Gaussians with equal covariance matrices, in which case the Bayes discriminant is linear. They analyze the "plug-in" method from the generative paradigm (see Section 2.1) in which the parameters of the class distributions are estimated by maximum likelihood. If the two Gaussians are somewhat well-separated, the asymptotic gain of using unlabeled samples is very significant. For details, see [21, 13, 14]. McLachlan [18] gives a practical algorithm for this case which is essentially a "hard" version of EM, *i.e.* in every E step the unlabeled points are allocated to one of the populations, using the discriminant derived from the mixture parameters of the previous step (note that the general EM algorithm had not been proposed at that time). He proves that for "moderate-sized" training sets from each population and for a pool D_u of points sampled from the mixture, if the algorithm is initialized with the ML solution based on the labeled data, the solutions computed by the method converge almost surely against the true mixture distribution with $|D_u| = m \to \infty$. These early papers provide some important insight into properties of the semi-supervised problem, but their strict assumptions limit the conclusions that can be drawn for large real-world problems.

The EM algorithm has been applied to text classification by Nigam *et.al.* (see [20], or Chap.??? in this volume). From Eq. 1 we see that in the joint log likelihood, labeled and unlabeled data are weighted at the ratio n to m. This "natural" weighting makes sense if the likelihood is taken at face value, *i.e.* as a correct description of the sampling mechanism for the data, but it is somewhat irrelevant for the problem of SSL where a strong sampling bias is present whose exact size is usually unknown. In other words, unlabeled data is often available in huge quantities simply because it can be obtained much cheaper than labeled data. If we use the natural weighting in the interesting case $m \gg n$, the labeled data D_l are effectively ignored. Nigam *et.al.* suggest to reweight the terms in Eq. 1 by $(1-\lambda)/n$ and λ/m respectively (the natural weighting is given by $\lambda = m/(m+n)$) and to adjust λ by standard techniques such as cross-validation on D_l.

Note that y is treated as the latent class variable as far as the estimation of $P(x)$ from D_u is concerned, and we can just as well allow for more mixture components than classes. Namely, we can introduce an additional *separator variable* k such that under the model x and y are independent given k. This means that all the information x contains about its class y is already captured in k. This fact is illustrated in the independence model on the right.



The reweighted joint log likelihood is

$$\frac{1-\lambda}{n}\sum_{i=1}^{n}\log\sum_{k}\beta_{y_i,k}\pi_k P(x_i|k,\theta)+\frac{\lambda}{m}\sum_{i=n+1}^{n+m}\log\sum_{k}\pi_k P(x_i|k,\theta),$$

where $\pi_k = P(k|\theta)$ and $\beta_{y,k} = P(y|k,\theta)$. It is straightforward to maximize this criterion using EM. Miller and Uyar [19] present some results using this model together with Gaussian components $P(x|k,\theta)$. The "many-centers-per-class" case in [20] is equivalent to this method.

Some drawbacks of this simple generative mixture model approach have already been mentioned in Section 2.1. First, the weighting λ between the labeled and unlabeled data sources has to be chosen carefully, for example the natural weighting is usually not appropriate. A selection of λ by cross-validation on D_l is robust in principle, but bound to fail if n is very small. Second, for λ not close to 0 the joint log likelihood has many (local) maxima, and for $\lambda \to 1$ consistency with D_l is less and less enforced. Both problems are adressed in a principal manner by Corduneanu and Jaakkola [9]. Under suitable identifiability conditions[9] on $P(x|y,\theta)$ the maximum point for $\lambda = 0$ (labeled data only) is unique, while for $\lambda = 1$ (unlabeled data only) there are many equivalent maximum points at least due to label permutation symmetry. Therefore, as we trace the maximum point for growing λ starting from 0, the path must split at a first critical $\lambda^* > 0$. The authors argue that the maximum point of the log likelihood at this λ^* provides a promising solution to the SSL problem (in this generative setting) in that it still fully incorporates the label information. Also, the path up to λ^* is unique, while it splits for larger λ, and the decision of which one to follow is independent of the label information. They show how to employ homotopy continuation (path following) methods in order to trace the solution path up to λ^* fairly efficiently. By restricting themselves to $\lambda \leq \lambda^*$ they circumvent the many (local) maxima problem, and their choice of $\lambda = \lambda^*$ is well-motivated.

Murray and Titterington (see [27], Ex. 4.3.11) suggest to use D_l for each class to obtain kernel-based estimates of the densities $P(x|t)$. They fix these estimates and use EM in order to maximize the joint likelihood of D_l, D_u w.r.t. the mixing coefficients π_t only.[10] This procedure is robust, but does not make a lot of use of the unlabeled data. If D_l is small, the kernel-based estimates of the $P(x|t)$ will be poor, and even if D_u can be used to obtain better values for the mixing coefficients, this is not likely to rescue the final discrimination. Furthermore, the procedure has been suggested for situations where the natural weighting between D_l, D_u is appropriate, which is typically not the case for SSL.

Shahshahani and Landgrebe [25] provide an analysis aimed towards the general question whether unlabeled data can help in classification, based on methods originating in asymptotic maximum-likelihood theory. Their argumentation is somewhat unclear and has been criticized by various other authors (*e.g.*, [20, 29]). They do not define model classes and seem to confuse asymptotic and

[9] These are not very restrictive, for example they hold for all (regular) exponential families.

[10] EM w.r.t. the mixing coefficients only always converges to a unique *global* optimum. It is essentially a variant of the *Blahut-Arimoto algorithm* to compute the *rate distortion function* which is important for quantization, see [10].

finite-sample terms. After all, their claim seems to be that unlabeled data can reduce the asymptotic variance of an estimator, but they do not worry about the fact that such modifications could actually introduce new bias, especially in the interesting case where $m \gg n$. On the practical side, the algorithm they suggest is the joint EM scheme discussed above.

Another analysis of SSL which also employs Fisher information, is given by Zhang and Oles [29]. The authors show that for purely diagnostic models, unlabeled data cannot help (this fact is of course known since a long time, see also Section 2.2). In the generative setup, they show that D_u can only help. While this is true under their assumptions, it draws on asymptotic concepts and may not be relevant in practical situations. The Fisher information characterizes the minimal *asymptotic* variance of an unbiased estimator only, and the maximum-likelihood estimator is typically only *asymptotically* unbiased. Applying such concepts to the case where D_l is small cannot lead to strong conclusions, and the question of (even asymptotic) bias remains in the case where m grows much faster than n. On the practical side, some empirical evidence is presented on a text categorization task which shows that unlabeled data can lead to instabilities in common transduction algorithms and therefore "hurt" (see comments in Section 2.3).

3.2 Diagnostic Techniques

We noted in Section 2.2 that unlabeled data cannot be used in Bayesian diagnostic methods if θ and μ are *a priori* independent, so in order to make use of D_u we have to employ conditional priors $P(\theta|\mu)$. Unlabeled data may still be useful in non-Bayesian settings. An example has been given by Tong and Koller [28] under the name of *restriced Bayes optimal classification (RBOC)*. Consider a diagnostic method in which the sum of an empirical loss term and a regularization functional is minimized. The empirical loss term is the expectation w.r.t. the labeled sample D_l of a loss function relevant for the problem (for example, the zero-one loss $L(x,y,h) = \mathrm{I}_{\{t \neq h(x)\}}$). The authors suggest incorporating unlabeled data D_u by estimating $P(x,y)$ from $D_l \cup D_u$, then replacing the empirical loss term by the expectation of the loss under this estimate. The regularization term is not changed. We can compare this method directly with input-dependent regularization (see Section 2.3). In the former, the empirical loss part (the negative log likelihood for a probabilistic model) is modified based on D_u, in the latter it is the regularization term. We would not expect RBOC to produce very different results from the corresponding diagnostic technique, especially if n is rather small (which is the interesting case in practice). This is somewhat confirmed by the weak results in [28]. A very similar idea is proposed in [7] in order to modify the diagnostic SVM framework.

Anderson [1] suggested an interesting modification of logistic regression in which unlabeled data can be used. In binary logistic regression, the log odds are modelled as linear function, which gives $P(x|1) = \exp(\beta^T x)P(x|2)$ and $P(x) = (\pi_1 \exp(\beta^T x) + 1 - \pi_1)P(x|2)$, where $\pi_1 = P\{t = 1\}$. Anderson now chooses the parameters β, π_1 and $P(x|2)$ in order to maximize the likelihood of both D_l and D_u, subject to the constraints that $P(x|1)$ and $P(x|2)$ are normalized. For finite $\mathcal{X}$, this problem can be transformed into an unconstrained optimization w.r.t. the parameters β, π_1. For a continuous input variable x, Anderson advocates using the form of $P(x|2)$ derived for the "finite $\mathcal{X}$" case, although this is not a smooth function. Unfortunately, it is not clear how to generalize this idea to more realistic models, for example how to "kernelize" it, and the form of $P(x|2)$ is inadequate for many problems with infinite $\mathcal{X}$.

3.3 Input-Dependent Regularization

We discussed in Section 2.3 that unlabeled data D_u can be useful within a diagnostic technique if θ and μ are dependent *a priori*. In order to implement this idea, we have to specify conditional priors $P(\theta|\mu)$ encoding our belief in how characteristics of $x \rightarrow y$ depend on knowledge about $P(x)$.

3.3.1 The Cluster Assumption

It is not hard to construct "malicious" examples of $P(x,y)$ which defy any given dependence assumption on θ, μ. However, in practice it is often the case that cluster structure in the data for x indeed is mostly consistent with the labeling. It is not very fruitful to speculate about why this is the case, although certainly there is a selection bias towards features (*i.e.* components in x) which are *relevant* w.r.t. the labeling process, which means they should group in the same way (w.r.t. a simple distance) as labelings. The *cluster assumption (CA)* (*e.g.*, [22]) provides a general way of exploiting this observation for SSL. It postulates that two points x', x'' should have the same label y with high probability if there is a "path" between them in $\mathcal{X}$ which moves through regions of significant density $P(x)$ only. In other words, a discrimination function between the classes should be smooth within connected high-density regions of $P(x)$. Thus, the CA can be compared directly with *global* smoothness assumptions requiring the discriminant to change smoothly everywhere, independent of $P(x)$. While the latter penalize sharp changes also in regions which will be sparsely populated by training and test data, the CA remains indifferent there.

The CA is implemented (to different extent) in a host of methods proposed for SSL. Most prominent are probably *label propagation* methods [26, 4, 30]. The rough idea is to construct a graph with vertices from $X_l \cup X_u$ which contains the test set to be labeled and all of X_l. Nearest neighbors are joined by edges with a weight proportional to local correlation strength. We then initialize the nodes corresponding to X_l with the labels Y_l and propagate label distributions over the remaining nodes in the manner of a Markov chain on the graph [26]. It is also possible to view the setup as a Gaussian field with the graph and edge weights specifying the inverse covariance matrix [30]. Label propagation techniques implement the CA relative to unsupervised spectral clustering [4]. The CA has been implemented for kernel machines by way of the cluster kernel [8]. Furthermore, the generative SSL techniques of Section 3.1 can be seen as implementing the CA relative to a mixture model clustering.

A generalization of the CA has been given by Corduneanu and Jaakkola (see Chap.??? in this volume) who show how to obtain a regularizer for the conditional distribution $P(y|x)$ from information-theoretic arguments.

3.3.2 The Fisher Kernel

The *Fisher kernel* was proposed in [15] in order to exploit additional unlabeled data within a kernel-based *support vector machine (SVM)* framework for detecting remote protein homologies. The idea is to fit a generative model $P(x|\mu)$ to D_u by maximum likelihood (resulting in $\hat{\mu}$, say). If x are DNA sequences, a hidden Markov model (HMM) can be employed. $P(x|\hat{\mu})$ represents the knowledge extracted from D_u, and the Fisher kernel is a general way of constructing a covariance kernel $K_{\hat{\mu}}$ which depends on this knowledge. We can then fit an SVM or a Gaussian process (GP) classifier to D_u using the kernel $K_{\hat{\mu}}$. Identifying this setup as an instance of input-dependent regularization is easiest in the GP context. Here, θ is a process representing the discriminant function (we assume

$c = 2$ for simplicity), and $P(\theta|\mu)$ is a GP distribution with zero mean function and covariance kernel K_μ. In the ML context, $P(\mu|D_u)$ is approximated by the Delta distribution $\delta_{\hat{\mu}}$.

Define the Fisher score to be $F_{\hat{\mu}}(x) = \nabla_{\hat{\mu}} \log P(x|\mu)$ (the gradient w.r.t. μ is evaluated at $\hat{\mu}$). The Fisher information matrix is $F = \mathrm{E}_{P(\cdot|\hat{\mu})}[F_{\hat{\mu}}(x)F_{\hat{\mu}}(x)^T]$. The naive Fisher kernel is $K_{\hat{\mu}}(x,x') = F_{\hat{\mu}}(x)^T F^{-1} F_{\hat{\mu}}(x')$. In a variant, F is replaced by αI for a scale parameter α. Other variants of the Fisher kernel are obtained by using the Fisher score $F_{\hat{\mu}}(x)$ as feature vector for x and plug these into a standard kernel such as the Gaussian (RBF) one. The latter "embeddings" seem to be more useful in practice. The Fisher kernel can be motivated from various angles (see [15]), for example as first-order-approximation to a sample mutual information between x, x' [23].

3.3.3 Co-training

Co-training was introduced by Blum and Mitchell [5] and is related to earlier work on unsupervised learning [2]. The idea is to make use of different "views" on the objects to be classified (we restrict ourselves to binary classification, $c = 2$, and to two views). For example, a Web page can be represented by the text on the page, but also by the text of hyperlinks referring to the page. We can train classifiers separately which are specialized to each of the views, but in this context unlabeled data D_u can be helpful in that although the true label is missing, it must be the same for all the views. It turns out that Co-training can be seen as a special case of Bayesian inference using conditional priors (see Section 2.3), as is demonstrated below in this Section.

Let $\mathcal{X} = \mathcal{X}^{(1)} \times \mathcal{X}^{(2)}$ be a finite or countable input space. If $x = (x^{(1)}, x^{(2)})$, the $x^{(j)}$ are different "views" on x. We are also given spaces $\Theta^{(j)}$ of concepts (binary classifiers) $\theta^{(j)}$. Elements $\theta = (\theta^{(1)}, \theta^{(2)}) \in \Theta = \Theta^{(1)} \times \Theta^{(2)}$ are called concepts over $\mathcal{X}$, although we may have $\theta^{(1)}(x^{(1)}) \neq \theta^{(2)}(x^{(2)})$ for some $x = (x^{(1)}, x^{(2)}) \in \mathcal{X}$. Whenever the $\theta^{(j)}$ agree, we write $\theta(x) = \theta^{(1)}(x^{(1)})$. If $A \subset \mathcal{X}$, we say that a concept $\theta = (\theta^{(1)}, \theta^{(2)})$ is *compatible* with A if $\theta^{(1)}(x^{(1)}) = \theta^{(2)}(x^{(2)})$ for all $x = (x^{(1)}, x^{(2)}) \in A$. Denote by $\Theta(A)$ the space of all concepts compatible with A.[11] If $Q(x)$ is a distribution over $\mathcal{X}$ with *support* $S = \operatorname{supp} Q(x) = \{x|Q(x) > 0\}$, we say that a concept θ is *compatible* with the distribution Q if it is compatible with S.

In the Co-training setting, there is an unknown input distribution $P(x)$. A *target concept* θ is sampled from some unknown distribution over Θ, and the data distribution is $P(y|x) = I_{\{\theta(\mathbf{x})=t\}}$ if $\theta \in \Theta(\{x\})$, $1/2$ otherwise[12]. However, the central assumption is that the target concept θ is *compatible* with the input distribution $P(x)$. More specifically, the support of the concept distribution must be contained in $\Theta(\operatorname{supp} P(x))$. Therefore, unlabeled data D_u can be used by observing that $\Theta(\operatorname{supp} P(x)) \subset \Theta(D_u \cup X_l)$, so the effective concept space can be shrunk from Θ to $\Theta(D_u \cup X_l)$.

We demonstrate that Co-training can be understood as Bayesian inference with conditional priors encoding the compatibility assumption. We model $P(x)$ by $\{P(x|\mu)\}$ and introduce the variable $S = \operatorname{supp} P(x|\mu)$ for convenience, then define $P(\theta|\mu) = P(\theta|S)$ as

$$P(\theta|S) = f_S(\theta) \mathrm{I}_{\{\theta \in \Theta(S)\}},\ S \subset \mathcal{X},$$

where $f_S(\theta) > 0$, and all $P(\theta|S)$ are properly normalized. For example, if $\Theta(S)$ is finite, we can choose $f_S(\theta) = |\Theta(S)|^{-1}$. The likelihood is given by $P(y|x,\theta) = (1/2)(\mathrm{I}_{\{\theta^{(1)}(x^{(1)})=t\}} + \mathrm{I}_{\{\theta^{(2)}(x^{(2)})=t\}})$

[11] 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 $\Theta^{(j)}$.

[12] 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.

(noiseless case). Since $P(\theta|S) = 0$ for $\theta \notin \Theta(S)$, the conditional prior encodes the compatibility assumption. The posterior belief about θ is given by

$$P(\theta|D_l, D_u) \propto \mathrm{I}_{\{\theta(x_i)=y_i, i=1,\ldots,n\}} \int P(\theta|S)P(S|X_l, D_u)\, dS,$$

so that $P(\theta|D_l, D_u) \neq 0$ iff θ is consistent with the labeled data D_l and $\theta \in \Theta(D_u \cup X_l)$. Namely, if $\theta \notin \Theta(D_u \cup X_l)$, then $P(\theta|S) = 0$ for all S which contain $D_u \cup X_l$, and $P(S|D_u, X_l) = 0$ for all other S. On the other hand, if $\theta \in \Theta(D_u \cup X_l)$, then we have $P(\theta|\hat{S}) > 0$ and $P(\hat{S}|D_u, X_l) > 0$ at least for $\hat{S} = D_u \cup X_l$. In the terminology of Blum and Mitchell, $\mathrm{supp}\, P(\theta|D_l, D_u)$ is equal to the "version space" given all the data. The biases for the learning methods on $\Theta^{(j)}$ may be encoded in the potentials $f_S(\theta)$.

Once Co-training is understood within a Bayesian framework with conditional priors, one can employ standard techniques in order to perform inference. In fact, we showed in [24] that the Co-training algorithm suggested by Blum and Mitchell can be seen as a variant of (sequential) EM on the probabilistic model sketched above. This viewpoint allows us to generalize Co-training along various dimensions, *e.g.* allowing for noise, smoother prior distributions, using batch rather than online training, uncertain rather than fixed labels on the test points, *etc*. We refer to [24] for details.

4 Conclusions

In this Chapter we have described a simple taxonomy of methods for semi-supervised learning and given many examples of SSL methods for each of the categories. Advantages and potential pitfalls of each group have been discussed. We have underlined the importance of using conditional priors in diagnostic Bayesian SSL techniques and have given several examples of methods proposed in the literature which fall into this category.

References

[1] J. A. Anderson. Multivariate logistic compounds. *Biometrika*, 66:17–26, 1979.

[2] S. Becker and G.E. Hinton. A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.

[3] S. Becker, S. Thrun, and K. Obermayer, editors. *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[4] M. Belkin and P. Niyogi. Using manifold structure for partially labeled classification. In Becker et al. [3].

[5] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with Co-Training. In *Conference on Computational Learning Theory 11*, 1998.

[6] V. Castelli and T. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16:105–111, 1995.

[7] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[8] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In Becker et al. [3].

[9] A. Corduneanu and T. Jaakkola. Continuation methods for mixing heterogeneous sources. In A. Darwiche and N. Friedman, editors, *Uncertainty in Artificial Intelligence 18*. Morgan Kaufmann, 2002.

[10] Thomas Cover and Joy Thomas. *Elements of Information Theory*. Series in Telecommunications. John Wiley & Sons, 1st edition, 1991.

[11] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Roy. Stat. Soc. B*, 39:1–38, 1977.

[12] T. Dietterich, S. Becker, and Z. Ghahramani, editors. *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[13] S. Ganesalingam and G. McLachlan. The efficiency of a linear discriminant function based on unclassified initial samples. *Biometrika*, 65:658–662, 1978.

[14] S. Ganesalingam and G. McLachlan. Small sample results for a linear discriminant function estimated from a mixture of normal populations. *Journal of Statistical Computation and Simulation*, 9:151–158, 1979.

[15] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1999.

[16] M. I. Jordan, editor. *Learning in Graphical Models*. Kluwer, 1997.

[17] S. Lauritzen. *Graphical Models*. Oxford Statistical Sciences. Clarendon Press, 1996.

[18] G. McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70:365–369, 1975.

[19] David Miller and Hasan Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in Neural Information Processing Systems 9*, pages 571–577. MIT Press, 1997.

[20] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 1998.

[21] T. J. O'Neill. Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73:821–826, 1978.

[22] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for ANC, Edinburgh, UK, 2000. See `www.kyb.tuebingen.mpg.de/bs/people/seeger`.

[23] M. Seeger. Covariance kernels from Bayesian generative models. In Dietterich et al. [12], pages 905–912.

[24] Matthias Seeger. Input-dependent regularization of conditional density models. Technical report, Institute for ANC, Edinburgh, UK, 2000. See `www.kyb.tuebingen.mpg.de/bs/people/seeger`.

[25] B.M. Shahshahani and D.A. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.

[26] Martin Szummer and Tommi Jaakkola. Partially labeled classification with Markov random walks. In Dietterich et al. [12], pages 945–952.

[27] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1st edition, 1985.

[28] S. Tong and D. Koller. Restricted Bayes optimal classifiers. In *Proceedings of AAAI*, pages 658–664, 2000.

[29] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In P. Langley, editor, *International Conference on Machine Learning 17*. Morgan Kaufmann, 2000.

[30] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In T. Fawcett and N. Mishra, editors, *International Conference on Machine Learning 20*. Morgan Kaufmann, 2003.