A Taxonomy for Semi-Supervised Learning Methods

Seeger, Matthias
Max Planck Institute for Biological Cybernetics
P.O. Box 21 69, 72012 Tuebingen, Germany
E-mail: seeger@tuebingen.mpg.de

1 Introduction

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 paper, we review some existing methods, thereby isolating important general concepts of this new paradigm, and relating it with more traditional ones.

In statistical machine learning, methods are divided in *unsupervised* and *supervised* ones. Unsupervised learning is known as density estimation in statistics: characteristics of the distribution of a variable $\boldsymbol{x} \in \mathcal{X}$ are to be estimated from a sample $\{\boldsymbol{x}_i\}$, assumed to be drawn *independently and identically distributed* (i.i.d.) from an unknown $P(\boldsymbol{x})$. Versatile models go beyond simple families, trying to represent $P(\boldsymbol{x})$ via latent factors. For example, *mixture models* have the form $\sum_{y=1}^{C} P(\boldsymbol{x}|y)P(y)$, with a class variable $y \in \{1, \ldots, C\}$ which is not observed (unsupervised). In supervised learning, we estimate the relationship $\boldsymbol{x} \to y$, based on a sample $\{(\boldsymbol{x}_i, y_i)\}$, *i.e.* y is observed. We can of course always estimate the joint density $P(\boldsymbol{x}, y)$, but this is wasteful since $\boldsymbol{x}$ is always given at prediction time, so there is no need to estimate its marginal distribution. It is important to note that the role of y is quite different in (supervised) classification versus mixture modelling. In the former, y is a query variable, while in the second, it is merely used as nuisance variable to obtain a more versatile $P(\boldsymbol{x})$ model. In the sequel, $\boldsymbol{x}$ is referred to as input, y as target.

The SSL problem is a supervised one, since the goal is to predict y for given $\boldsymbol{x}$. The difference to standard classification lies in the available data: a small *labeled sample* $D_l = \{(\boldsymbol{x}_i, y_i) \,|\, i = 1, \ldots, n\}$ is extended by a larger *unlabeled sample* $D_u = \{\boldsymbol{x}_{n+j} \,|\, j = 1, \ldots, m\}$ drawn from the marginal $P(\boldsymbol{x})$. In many machine learning applications, unlabeled data D_u can easily be obtained in large quantities, while labeling such data is expensive and time-consuming. We denote $\boldsymbol{X}_l = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n), Y_l = (y_1, \ldots, y_n)$ and $\boldsymbol{X}_u = (\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+m})$. The unobserved labels are denoted $Y_u = (y_{n+1}, \ldots, y_{n+m})$.

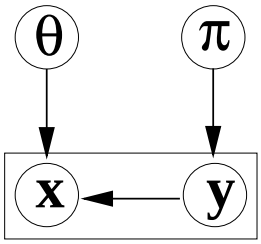
There are two obvious baseline methods for SSL. First, we can ignore D_u and treat it as a supervised problem using D_l only. Second, we can treat y as latent class variable in a mixture model, converting it into an unsupervised problem with partial supervision on the data in D_l. Of course, any valid SSL technique should outperform both baseline methods significantly on practically relevant situations it is designed for.

In our view, SSL is much more a practical than a theoretical problem. A useful general SSL technique should be configurable to specifics of the task just like Bayesian methods, through the choice of prior and model.

2 Paradigms for Semi-Supervised Learning

Being supervised techniques, SSL methods can be classified into *generative* and *diagnostic* paradigms, although the borderline is somewhat ambiguous. We review these paradigms and introduce a third one which is very relevant in the SSL context.

2.1 The Generative Paradigm



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

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

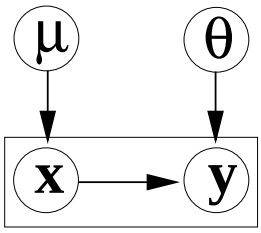
We can also obtain the Bayesian predictive distribution $P(y|\boldsymbol{x}, D_l)$ by averaging $P(y|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\pi})$ over the posterior $P(\boldsymbol{\theta},\boldsymbol{\pi}|D_l)$. Within the generative paradigm, a model for the marginal $P(\boldsymbol{x})$ emerges naturally as $P(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\pi}) = \sum_{y=1}^{M} \pi_y P(\boldsymbol{x}|y,\boldsymbol{\theta})$. If labeled and unlabeled data are available, a natural criterion emerges as the *joint log likelihood* of both D_l *and* D_u,

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

or alternatively the log posterior $\log P(\boldsymbol{\theta},\boldsymbol{\pi}|D_l, D_u)$. This is an issue of maximum likelihood (ML) for partly missing data, and expectation-maximization techniques can be used (see Section 3.1).

This generative baseline technique seems like an obvious "best solution" to SSL, but it is not in general. SSL is fundamentally a classification problem, where the goal is to estimate $P(y|\boldsymbol{x})$, while generative technique focus on modelling the joint $P(\boldsymbol{x}, y)$. In many situations, $\boldsymbol{x}$ has a very complicated structure, and a generative technique will spend much effort to capture it, in fact using y as a nuisance grouping variable to this end, rather than heeding its role as primary query variable. The problem is closely related to the well-known fact that generative models often do not work well for classification in a purely supervised setting. Classical generative techniques can be no more than a baseline for SSL, and specific SSL methods will have to outperform them.

2.2 The Diagnostic Paradigm



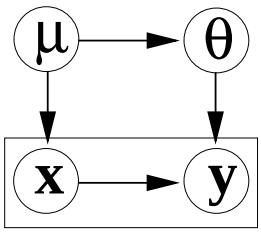
In *diagnostic methods*, we model the conditional distribution $P(y|\boldsymbol{x})$ directly using the family $\{P(y|\boldsymbol{x},\boldsymbol{\theta})\}$. To arrive at a complete sampling model for the data, we also have to model $P(\boldsymbol{x})$ by a family $P(\boldsymbol{x}|\boldsymbol{\mu})$,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

$$P(D_l, D_u|\boldsymbol{\theta},\boldsymbol{\mu}) = P(Y_l|\boldsymbol{X}_l,\boldsymbol{\theta})P(\boldsymbol{X}_l, D_u|\boldsymbol{\mu}),$$

which implies that $P(\boldsymbol{\theta}|D_l, D_u) \propto P(Y_l|\boldsymbol{X}_l,\boldsymbol{\theta})P(\boldsymbol{\theta})$, 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 about the unlabeled data D_u nor *any* knowledge about $\boldsymbol{\mu}$ modifies the posterior belief, which is $P(\boldsymbol{\theta}|D_l)$ based on D_l alone. Unlabeled data is of no use in a standard diagnostic model. We have to modify the data generation model in order to make it applicable to SSL.

2.3 Regularization depending on the Input Distribution

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 $\boldsymbol{\theta}$ (parameterizing $P(y|\boldsymbol{x})$) and $\boldsymbol{\mu}$ (parameterizing $P(\boldsymbol{x})$) are *a priori* independent. In other words, the model family $\{P(y|\boldsymbol{x}, \boldsymbol{\theta})\}$ is regularized *independently* of the input distribution. If we allow prior dependencies between $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$, *i.e.* $P(\boldsymbol{\theta}, \boldsymbol{\mu}) = P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})$ and $P(\boldsymbol{\theta}) = \int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})\, d\boldsymbol{\mu}$, the situation is different. The conditional prior $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ in principle allows information about $\boldsymbol{\mu}$ to be transferred to $\boldsymbol{\theta}$, allowing for example a regularization of $\boldsymbol{\theta}$ which depends on what is known about $P(\boldsymbol{x})$



In order to make use of unlabeled data in diagnostic Bayesian models, we have to introduce *a priori* dependence between the conditional probability $P(y|\boldsymbol{x})$ and the marginal $P(\boldsymbol{x})$. If $P(y|\boldsymbol{x})$ is represented via functions, their *regularization has to depend on the input distribution.* Conditional priors imply a mixture distribution as marginal prior $P(\boldsymbol{\theta}) = \int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})\, d\boldsymbol{\mu}$. By conditioning on the unlabeled data, this is replaced by $P(\boldsymbol{\theta}|D_u) = \int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu}|D_u)\, d\boldsymbol{\mu}$, which can be much narrower than $P(\boldsymbol{\theta})$, so that the posterior belief $P(\boldsymbol{\theta}|D_l, D_u)$ can be much more concentrated than $P(\boldsymbol{\theta}|D_l)$. This argument shows that unlabeled data D_u can hurt instead of help classification. If the priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ enforce certain constraints rigidly, but these happen to be violated in the true distribution $P(\boldsymbol{x}, y)$, the conditional "prior" $P(\boldsymbol{\theta}|D_u)$ will assign much lower probability than $P(\boldsymbol{\theta})$ to models $P(y|\boldsymbol{x}, \boldsymbol{\theta})$ compatible with the truth. Robustness is a major issue with SSL, where unlabeled data has to be infused carefully in order to avoid such pitfalls.

Within this extension of the diagnostic paradigm, we have to model aspects of $P(\boldsymbol{x})$. Are we not doing generative modelling then? While there is certainly some ambiguity here, we can resolve this question by noting that we only model aspects of $P(\boldsymbol{x})$ explicitly relevant for regularizing $P(y|\boldsymbol{x})$, ignoring all other features of $P(\boldsymbol{x})$. For example, the cluster assumption (see Section 3.3.1) states that labels of "nearby" input points should be highly correlated. Since corresponding conditional priors only depend on usable distance information between $\boldsymbol{x}$ points, it is only this aspect which has to be faithfully represented in the $P(\boldsymbol{x})$ model.

3 Examples

In this section, we provide examples of SSL methods falling in each of the categories introduced above. We do not aim for a comprehensive review (see [19] for a review of work up to about 2001), but rather pick examples in order to illustrate our taxonomy, and to introduce specific SSL concepts transcending the field.

3.1 Generative SSL Techniques

As noted in Section 2.1, an obvious baseline for SSL is to use a conventional model for mixture density estimation, treating y as latent class variable. For example, maximum likelihood estimation can be done on such models by the *expectation-maximization* (EM) algorithm [10]. The labeled data D_l can either be added to the likelihood as in (1), or can be used afterwards in order to assign detected clusters to classes. If there are inconsistencies, the information in D_l should be given higher weight. Castelli and Cover [5] provide a simple analysis of the latter baseline method, but they use fairly unrealistic identifiability conditions.

The idea of using EM to maximize the joint likelihood of labeled and unlabeled data is almost as old as EM itself. Early theoretical work on the problem of discriminant analysis in the presence of additional unlabeled data is reviewed in [22], Sect. 5.7. 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, where 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 [18, 12, 13]. McLachlan [15] gives a practical algorithm for this case which is essentially a "hard" version of EM (note that EM 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 almost surely converge to the true mixture distribution with $|D_u| = m \to \infty$.

The EM algorithm has been applied to text classification in [17]. From (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 of unknown size is present[1]. The labeled and unlabeled part in (1) can be weighted by $(1-\lambda)/n$ and λ/m respectively, as suggested in [17], and λ can be adjusted by cross-validation on D_l.

We already noted that a significant problem with generative techniques for SSL is that in those, y is treated as nuisance grouping variable used to obtain a good fit to $P(\boldsymbol{x})$, rather than as a query variable of interest. This problem can be alleviated somewhat by a "many-centers-per-class" extension [16, 17], where an additional discrete separator variable k is introduced for the grouping, with the assumption that $P(\boldsymbol{x}, y, k) = P(\boldsymbol{x}|k)P(y|k)P(k)$. The reweighted joint log likelihood is

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

where $\pi_k = P(k|\boldsymbol{\theta})$ and $\beta_{y,k} = P(y|k,\boldsymbol{\theta})$. It is straightforward to maximize this criterion using EM.

Drawbacks of this approach include that λ has to be chosen properly, based on a small labeled sample D_l only. Furthermore, the joint log likelihood (1) has many local maxima, and EM is bound to get stuck in one of them. Both problems are addressed in [7]. They trace the path of ML solutions from $\lambda = 0$ (labeled data only) towards $\lambda = 1$ (unlabeled data only), using a homotopy continuation method. For $\lambda = 0$, there is in general a single global maximum, but the path eventually bifurcates at a first critical $\lambda_* \in (0, 1)$. The authors argue that the ML model at this λ^* provides a promising solution to the SSL problem, in that it still fully incorporates the label information. 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, so any choice will lead to non-robust behaviour.

Murray and Titterington (see [22], Ex. 4.3.11) suggest to use D_l for each class to obtain kernel-based estimates of the densities $P(\boldsymbol{x}|y)$. 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[2]. This procedure is robust, but does not make a lot of use of the unlabeled data. Furthermore, if D_l is small, the kernel-based estimates of the $P(\boldsymbol{x}|y)$ will be fairly poor.

3.2 Diagnostic Techniques

We noted in Section 2.2 that unlabeled data cannot be used in Bayesian diagnostic methods if $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are *a priori* independent, so in order to make use of D_u we have to employ conditional priors

[1] This bias is present not only in the relative numbers n and m, but also in the fact that instances for labeled databases are often carefully selected in order to be "representative", while unlabeled data is usually just obtained in bulk. Working with such data goes beyond the SSL setup, where it is assumed that inputs in D_l and D_u come from the same distribution $P(\boldsymbol{x})$.

[2] 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, see [9].

$P(\boldsymbol{\theta}|\boldsymbol{\mu})$. Unlabeled data may still be useful in non-Bayesian settings. A simple example has been given by Tong and Koller [23] under the name of *restriced Bayes optimal classification* (RBOC). They estimate a discriminant function by minimizing the sum of an empirical loss and a regularization term. The empirical loss can be written as expectation over the empirical distribution coming from D_l. In RBOC, this distribution is replaced by an estimate of $P(\boldsymbol{x}, y)$, which is obtained jointly from $D_l \cup D_u$. The regularization term is not changed. We can compare this method directly with input-dependent regularization from 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 make a significant impact beyond the corresponding diagnostic technique using D_l only, especially in the most important case of rather small n, and the results presented in [23] are fairly weak. A very similar idea is proposed in [6] in order to modify the diagnostic *support vector machine* (SVM) framework.

3.3 Input-Dependent Regularization

Recall from Section 2.3 that unlabeled data D_u can be useful in a diagnostic Bayesian setting if $\boldsymbol{\theta}$ (for $P(y|\boldsymbol{x})$) and $\boldsymbol{\mu}$ (for $P(\boldsymbol{x})$) are dependent *a priori*. In order to implement this idea, we have to specify conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ encoding our belief in how characteristics of $\boldsymbol{x} \to y$ depend on knowledge about $P(\boldsymbol{x})$. Some generic concepts for doing so have emerged in the SSL literature, and we will describe them in this section.

3.3.1 The Cluster Assumption

It is not hard to construct "malicious" examples of $P(\boldsymbol{x}, y)$ which defy any given dependence assumption on $\boldsymbol{\theta}$, $\boldsymbol{\mu}$, and in fact lead to SSL methods following the assumption to fail badly. However, in practice it is often the case that cluster structure in the data for $\boldsymbol{x}$ indeed is mostly consistent with the labeling. Certainly there is a selection bias towards features (*i.e.* components in $\boldsymbol{x}$) which are *relevant* w.r.t. the labeling process, which means they should group in the same way (w.r.t. a simple distance such as Euclidean) as sensible labelings. The *cluster assumption* (CA) [19] provides a general way of exploiting this observation for SSL. It postulates that two points $\boldsymbol{x}'$, $\boldsymbol{x}''$ 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 $P(\boldsymbol{x})$ only. In other words, a discrimination function between the classes should be smooth (essentially constant) in connected high-density regions of $P(\boldsymbol{x})$. The CA can be contrasted with *global* smoothness assumptions, requiring the discriminant to change smoothly everywhere in the region of interest. While following the latter means that sharp changes are penalized also in regions sparsely populated by (labeled and unlabeled) data, the CA remains indifferent there.

The CA is implemented (to different extents) in a host of methods proposed for SSL. Most prominent are probably *label propagation* methods [21, 3, 24]. The rough idea is to construct a graph with vertices from $\boldsymbol{X}_l \cup \boldsymbol{X}_u$ which contains the test set to be labeled and all of $\boldsymbol{X}_l$. Nearest neighbors are joined by edges with a weight proportional to local correlation strength (usually obtained from a simple local kernel, such as a Gaussian). We then initialize the nodes corresponding to $\boldsymbol{X}_l$ with the labels Y_l and propagate label distributions over the remaining nodes in the manner of a Markov chain on the graph [21]. It is also possible to view the setup as a Gaussian field with the graph and edge weights specifying the inverse covariance matrix [24]. This matrix can be understood as a graph Laplacian, encouraging functions to behave smoothly in the sense of the CA. Label propagation techniques implement the CA relative to unsupervised spectral clustering [3].

A generalization of the CA has been given by Corduneanu and Jaakkola [8], who show how to obtain a regularizer for the conditional distribution $P(y|\boldsymbol{x})$ from information-theoretic arguments.

3.3.2 The Fisher Kernel

The *Fisher kernel* was proposed in [14] in order to exploit additional unlabeled data within a kernel-based SVM framework for detecting remote protein homologies. The idea is to fit a generative model $P(\boldsymbol{x}|\boldsymbol{\mu})$ to D_u by maximum likelihood (resulting in $\hat{\boldsymbol{\mu}}$, say). If $\boldsymbol{x}$ are DNA sequences, a hidden Markov model can be employed. $P(\boldsymbol{x}|\hat{\boldsymbol{\mu}})$ represents the knowledge extracted from D_u, and the Fisher kernel is a general way of constructing a covariance kernel $K_{\hat{\boldsymbol{\mu}}}$ which depends on this knowledge. We can then fit an SVM or a Gaussian process classifier to D_u using the kernel $K_{\hat{\boldsymbol{\mu}}}$. Identifying this setup as an instance of input-dependent regularization is easiest in the GP context. Here, $\boldsymbol{\theta}$ is a process representing the discriminant function (we assume $c = 2$ for simplicity), and $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ is a GP distribution with zero mean function and covariance kernel $K_{\boldsymbol{\mu}}$. In the ML context, $P(\boldsymbol{\mu}|D_u)$ is approximated by the Delta distribution $\delta_{\hat{\boldsymbol{\mu}}}$.

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

3.3.3 Co-training

Co-training was introduced in [4] and is related to earlier work on unsupervised learning [1]. The idea is to make use of different "views" on the objects to be classified (we restrict ourselves to binary classification, and to two views). For example, a WWW 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*. Co-training can be seen as a special case of Bayesian inference using conditional priors, as is demonstrated below.

The Co-training idea is a concept for SSL which is different from the CA. Features used in the different views do not have to behave similarly so as to induce a grouping in feature space consistent with labeling. The separate classifiers can use these features both in their own way. Quite in the contrary, features should be maximally uncorrelated (conditioned on y), which renders larger impact to the "consistency constraints" on the unlabeled data D_u.

Let $\mathcal{X} = \mathcal{X}^{(1)} \times \mathcal{X}^{(2)}$ be a finite or countable input space. If $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$, the $\boldsymbol{x}^{(j)}$ are different views on $\boldsymbol{x}$. $\Theta^{(j)}$ are spaces of concepts (binary classifiers) $\boldsymbol{\theta}^{(j)}$. Elements $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}) \in \Theta = \Theta^{(1)} \times \Theta^{(2)}$ are treated as concepts over $\mathcal{X}$, although we may have $\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)}) \neq \boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})$ for some $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in \mathcal{X}$. Whenever the $\boldsymbol{\theta}^{(j)}$ agree, we write $\boldsymbol{\theta}(\boldsymbol{x}) = \boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)})$. If $A \subset \mathcal{X}$, $\boldsymbol{\theta}$ is said to be *compatible* with A if $\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)}) = \boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})$ for all $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in A$. $\Theta(A)$ denotes the space of all concepts compatible with A. $\boldsymbol{\theta}$ is compatible with a distribution Q over $\mathcal{X}$ iff it is compatible with its support $S = \operatorname{supp} Q(\boldsymbol{x}) = \{\boldsymbol{x}|Q(\boldsymbol{x}) > 0\}$.

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 $P(y|\boldsymbol{x}) = I_{\{\theta(\mathbf{x})=t\}}$ if $\boldsymbol{\theta} \in \Theta(\{\boldsymbol{x}\})$, $1/2$ otherwise. The central assumption is that the target concept $\boldsymbol{\theta}$ is *compatible* with the unknown input distribution $P(\boldsymbol{x})$. If the current best concept estimates disagree on some sample $\boldsymbol{x}$ from $P(\boldsymbol{x})$, one of them must be wrong. Unlabeled data D_u can be used, in that the true concept must lie in $\Theta(D_u \cup \boldsymbol{X}_l)$, so the effective concept space can be shrunk from Θ to $\Theta(D_u \cup \boldsymbol{X}_l)$.

We demonstrate that Co-training can be understood as Bayesian inference with conditional priors encoding the compatibility assumption. We model $P(\boldsymbol{x})$ by $\{P(\boldsymbol{x}|\boldsymbol{\mu})\}$ and introduce the variable

$S = \operatorname{supp} P(\boldsymbol{x}|\boldsymbol{\mu})$ for convenience, then define $P(\boldsymbol{\theta}|\boldsymbol{\mu}) = P(\boldsymbol{\theta}|S)$ as

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

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 $f_S(\boldsymbol{\theta}) = |\Theta(S)|^{-1}$. The likelihood is given by $P(y|\boldsymbol{x}, \boldsymbol{\theta}) = (1/2)(\mathrm{I}_{\{\boldsymbol{\theta}^{(1)}(\boldsymbol{x}^{(1)})=t\}} + \mathrm{I}_{\{\boldsymbol{\theta}^{(2)}(\boldsymbol{x}^{(2)})=t\}})$ (noiseless case). Since $P(\boldsymbol{\theta}|S) = 0$ for $\boldsymbol{\theta} \notin \Theta(S)$, the conditional prior encodes the compatibility assumption. The posterior belief about $\boldsymbol{\theta}$ is given by

$$P(\boldsymbol{\theta}|D_l, D_u) \propto \mathrm{I}_{\{\boldsymbol{\theta}(\boldsymbol{x}_i)=y_i, \, i=1,\ldots,n\}} \int P(\boldsymbol{\theta}|S) P(S|\boldsymbol{X}_l, D_u) \, dS,$$

so that $P(\boldsymbol{\theta}|D_l, D_u) \neq 0$ iff $\boldsymbol{\theta}$ is consistent with the labeled data D_l and $\boldsymbol{\theta} \in \Theta(D_u \cup \boldsymbol{X}_l)$. Namely, if $\boldsymbol{\theta} \notin \Theta(D_u \cup \boldsymbol{X}_l)$, then $P(\boldsymbol{\theta}|S) = 0$ for all S which contain $D_u \cup \boldsymbol{X}_l$, and $P(S|D_u, \boldsymbol{X}_l) = 0$ for all other S. On the other hand, if $\boldsymbol{\theta} \in \Theta(D_u \cup \boldsymbol{X}_l)$, then we have $P(\boldsymbol{\theta}|\hat{S}) > 0$ and $P(\hat{S}|D_u, \boldsymbol{X}_l) > 0$ at least for $\hat{S} = D_u \cup \boldsymbol{X}_l$. In the terminology of Blum and Mitchell, $\operatorname{supp} P(\boldsymbol{\theta}|D_l, D_u)$ 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 $f_S(\boldsymbol{\theta})$.

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 [20] 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 [20] for details.

4 Conclusions

In this paper we have described a simple taxonomy of methods for semi-supervised learning, and have given 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, discussed some concepts for constructing such priors, and have given several examples of methods proposed in the literature which fall into this category.

REFERENCES

References

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

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

[3] M. Belkin and P. Niyogi. Using manifold structure for partially labeled classification. In Becker et al. [2].

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

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

[6] 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.

[7] 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.

[8] A. Corduneanu and T. Jaakkola. On information regularization. In C. Meek and U. Kjaerulff, editors, *Uncertainty in Artificial Intelligence 19*. Morgan Kaufmann, 2003.

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

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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`.

[20] 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`.

[21] Martin Szummer and Tommi Jaakkola. Partially labeled classification with Markov random walks. In Dietterich et al. [11], pages 945–952.

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

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

[24] 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.