Learning with labeled and unlabeled data

Matthias Seeger
Institute for Adaptive and Neural Computation
University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL
seeger@dai.ed.ac.uk

February 13, 2001

Abstract

In this paper, on the one hand, we aim to give a review on literature dealing with the problem of supervised learning aided by additional unlabeled data. On the other hand, being a part of the author's first year PhD report, the paper serves as a frame to bundle related work by the author as well as numerous suggestions for potential future work. Therefore, this work contains more speculative and partly subjective material than the reader might expect from a literature review.

We give a rigorous definition of the problem and relate it to supervised and unsupervised learning. The crucial role of prior knowledge is put forward, and we discuss the important notion of input-dependent regularization. We postulate a number of baseline methods, being algorithms or algorithmic schemes which can more or less straightforwardly be applied to the problem, without the need for genuinely new concepts. However, some of them might serve as basis for a genuine method. In the literature review, we try to cover the wide variety of (recent) work and to classify this work into meaningful categories. We also mention work done on related problems and suggest some ideas towards synthesis. Finally, we discuss some caveats and tradeoffs of central importance to the problem.

Contents

1 Introduction

Learning from data can be seen as the most rigorous attempt to *drastically compress* data without losing much of the inherent information. All learning strategies must therefore be based on the belief in the *hidden inherent simplicity* of relationships, *Occam's razor*, as is the whole of modern natural science. Statistical machine learning tries to replicate the highly original and creative patterns of human learning on dull computers, using concepts from probability theory.

The key to efficient compression is the introduction of *latent variables* associated with the observables, such that knowledge of the latent variables reduces the complexity of describing the observables drastically. Namely, the combined description of both latent and observable variables should be much less costly than the straightforward description of the observables alone. An example is the invention of words in a language to describe objects within our visual experiences.

To link these variables, in order to be able to work with them in an inference or coding-decoding machinery, we need to build *models*[1]. A *model family* is a conditional probability distribution $P(\mathcal{A}|\mathcal{B}, \boldsymbol{\theta})$, where $\mathcal{A}$ and $\mathcal{B}$ are disjoint sets of variables ($\mathcal{B}$ may be empty), and $\boldsymbol{\theta} \in \Theta$ is a latent variable associated with the model family. Often, model families are written as sets of distributions $\{P(\mathcal{A}|\mathcal{B}, \boldsymbol{\theta})|\boldsymbol{\theta} \in \Theta\}$, and the elements, being conditional distributions $\mathcal{A}|\mathcal{B}$ indexed by values of $\boldsymbol{\theta}$, are called *models*[2]. Some model families, e.g. prior distributions for variables at the top of the hierarchy (see below) or certain noise models, have $|\Theta| = 1$, in which case we can get rid of the variable $\boldsymbol{\theta}$. It is very important to note that all model families used within an inference machinery are known beforehand to coding and decoding side (as are the ranges of all the variables). Furthermore, we require that the whole machinery is completely defined in the sense that the total of all models allows us to compute a joint *prior distribution* over *all* variables in a consistent way.

Often, by observing which variable is conditioned on which under the model families, one can determine a certain ordering (or "direction of data generation") which gives rise to a *hierarchy*. The notion of a hierarchy is very important in Bayesian analysis, e.g. a complex of model families describing the part of the hierarchy which specifies the joint prior distribution for a subset of the latent variables, is sometimes referred to as *hierarchical prior*. Berger [9] gives a good introduction into Bayesian analysis.

There are two important, basic mechanisms for introducing latent variables to achieve better compression. The principle of *divide and conquer* states that

[1] Such models can be very simple, e.g. in the case of nonparametric statistical methods. We will give examples below.

[2] Note that we treat $\boldsymbol{\theta}$ in the same way as any other latent variable. Our framework would be easier if we defined model families simply as conditional distributions, however it is the general convention to link a model family with an explicit variable $\boldsymbol{\theta}$ and to write the family as set indexed by $\boldsymbol{\theta}$, and we do not want to break with this.

if a relationship cannot be described easily enough as it is, we should try to separate it into a finite number of units, each of which is more accessible to efficient description. In our framework, this can be done by introducing a new *grouping* or *clustering* variable k with a finite range. A model for $\mathcal{A}|\mathcal{B}$ can then be described by a *mixture* of models for $\mathcal{A}|\{\mathcal{B}, k\}$, this also involves modeling $k|\mathcal{B}$. A second mechanism works by imposing *functional relationships* between variables, obscured by completely unstructured noise. E.g. to describe the relationship $\mathcal{A}|\mathcal{B}$, we can build a model family $\{P(\mathcal{A}|\mathcal{B}, \boldsymbol{\theta})\}$ such that each model computes a fixed mapping $\Phi_\theta(\mathcal{B})$, then $P(\mathcal{A}|\mathcal{B}, \boldsymbol{\theta}) = P_{noise}(\mathcal{A}|\Phi_\theta(\mathcal{B}))$. The conditional distribution on the right side is called a *noise model.* It is the same for all the models in the family, and often has a simple parametric form, e.g. a Gaussian. Note that if data for $\mathcal{A}, \mathcal{B}$ is to be compressed given this model family, we should prefer models $\boldsymbol{\theta}$ such that the *true* distribution $P(\mathcal{A}|\Phi_\theta(\mathcal{B}))$ is indeed as unstructured as possible, and close to $P_{noise}(\mathcal{A}|\Phi_\theta(\mathcal{B}))$. Thus, the models we prefer for compression are aimed towards *separating structure from noise*, a central goal in learning from data.

1.1 Supervised and unsupervised learning

In statistical machine learning, two different scenarios can easily be distinguished:

- Supervised learning (learning with a teacher)
- Unsupervised learning

In the *supervised learning* scenario, aspects of an unknown probabilistic relationship $P(\boldsymbol{x}, t)$ between *examples* or *input points* $\boldsymbol{x} \in X$ and *targets* or *labels* $t \in T$ are to be learned from *labeled* data $\{(\boldsymbol{x}_i, t_i)|i = 1, \dots, n\}$, where the $(\boldsymbol{x}_i, t_i)$ are drawn independently from $P(\boldsymbol{x}, t)$. This problem class includes *pattern recognition* or *classification* (T finite) and *regression estimation* ($T \subset \mathbb{R}$).

In this paper, we will almost exclusively deal with the classification scenario, although most ideas should carry over to other scenarios like regression estimation. With respect to our coding perspective, we could say that in the classification case, somebody else (humans from earlier generations, clever scientists, ...) has already done the job of identifying t as grouping variable potentially valuable for efficient compression. Since we (and the world around us) have agreed to trust this decision, t is not latent anymore, but can be observed. We almost surely possess further prior knowledge about the relationship which we can use together with the data for our inference, and this prior knowledge might come from the same source. Some readers might object here, noting that even simple classification learning schemes can be shown to be *consistent*, i.e. can learn *any* relation $P(\boldsymbol{x}, t)$ given unlimited data, so why bother with these compression ideas and with models altogether? However, learning from *limited* data is of course an *ill-posed problem*, since without any kind of prior knowledge about the relationship, the observed data does not contain any information on how to *generalize* to unseen data.

Classification schemes can be grouped into two major classes (see [23],[70]), following either the *diagnostic* or the *sampling* paradigm. Methods within the diagnostic paradigm will be referred to as *diagnostic methods* (or *discriminative methods*), while schemes within the sampling paradigm will be called *generative methods*[3]. When designing a generative method, we use t as a grouping variable in the compression sense, i.e. we assume that the class distributions $P(\boldsymbol{x}|t)$ can be described efficiently. This assumption leads us to proposing a model family for each of the class distributions. In diagnostic methods, which are more related to regression estimation, we assume that $P(t|\boldsymbol{x})$ can be described efficiently, and we use a model family to impose a noisy functional relationship (the noise model is a multinomial one). The imposed functional relationships can be very simple, e.g. in *logistic regression* (e.g. [56]), but the model family can also be parameterized in a very complex way. For example, in *kernel methods*, the model family is parameterized by a latent function or mapping (i.e. $\boldsymbol{\theta}$ represents one or more random processes).

Schemes for regression estimation usually proceed in the same way as diagnostic classification methods, i.e. model $P(t|\boldsymbol{x})$. Traditionally, in such diagnostic schemes, the parameter $\boldsymbol{\theta}$ of the model family $\{P(t|\boldsymbol{x},\boldsymbol{\theta})\}$ and the input variable $\boldsymbol{x}$ are a-priori independent. This leads to schemes in which it is not necessary to learn anything about the marginal $P(\boldsymbol{x})$ from data. In coding terms, these schemes do not need to describe the input points in the data efficiently. We can easily modify our coding perspective in such cases, namely by allowing the input points in the dataset (to be compressed) to be sent for free. However, an important point we will make in this paper (see subsections 1.3.2,1.3.3 and 2.4) is that this kind of independence assumption is not sensible if we want to learn from additional unlabeled data. As soon as we drop this assumption (see e.g. [81]), efficient description of the input points becomes an issue.

While supervised learning usually follows a well-defined goal, e.g. minimizing the generalization error in classification or minimizing the expected loss in regression estimation, there are no such definitive criteria for *unsupervised learning* scenarios[4]. Here, we are required to find "interesting structures" within a sample $\{\boldsymbol{x}_i|i = 1,\dots,m\}$, independently drawn from the unknown distribution $P(\boldsymbol{x})$ (also called *source*). According to Occam's razor, what we are really looking for are structures which are inherently very simple, however obscured by unpredictibly random noise. This problem seems to be very much harder

[3] The term "sampling paradigm" is used for "historical" reasons, but actually clashes in the most unfortunate way with the terminology for methods employing *Monte Carlo sampling*, such are often referred to as *sampling methods*.

[4] There is a general criterion which has already been mentioned above. Namely, the best solution to an unsupervised learning problem is the one that enables us to encode the data source in the most efficient way. However, while this criterion can lead us successfully during the design of the model family, model selection or the search within a model family, it is too strong as an absolute criterion in practice, where it is often only feasible (or desirable) to focus on particular aspects instead of attacking the whole problem of the optimal representation.

than the supervised learning scenario, since it requires the algorithm (or the designer thereof) to *identify* latent variables suitable for efficient compression of the source. In essence, an unsupervised method performs *density estimation*, and many of the most successful unsupervised algorithms are formulated as *generative models* for $P(\boldsymbol{x})$ whose complexity is carefully controlled and regularized while being fitted to the data. By choosing appropriate restrictive model families, we can also aim for lower targets, i.e. learning particular kinds of structure in the data, with no intention of representing $P(\boldsymbol{x})$ faithfully by the final result, however the basic "drive" in the optimization is always to fit the data in the best possible way.

Examples of unsupervised techniques include latent subspace models like *principal component analysis (PCA)* (e.g. [11]), *factor analysis* (e.g. [29]) or *principal curves* [39]. Here, we introduce a latent "compression" variable $\boldsymbol{u}$, living in a low-dimensional space, furthermore impose a noisy functional relationship on $\boldsymbol{x}\,|\boldsymbol{u}$. The functional relationships are represented either by linear or by more powerful nonlinear models, in the latter case the model family is tightly regularized by an appropriate prior $P(\boldsymbol{\theta})$ on the model parameter $\boldsymbol{\theta}$. The noise model is usually a Gaussian. Other examples are *mixture models* (e.g. [58],[90],[69]) where the latent variable is a grouping variable from a finite set (similar to the class label in supervised classification), and the conditional models come from simple families such as Gaussians with structurally restricted covariance matrices. Combinations of mixture and latent subspace models have also been considered in numerous variants (e.g. [88], [34],[93],[5]).

Finally note how, within all these models, complexity can be regulated at various levels. The relations between latent and observable variables are kept simple by choosing relatively narrow model families or by regularizing models, i.e. by penalizing complex models within a family. This is naturally achieved by placing a *prior distribution* over the parameter of the model family, which judges simple models as more probable than complex ones. But also the complexity of other latent variables needs to be tightly controlled, e.g. the number of components in a mixture or the number of dimensions of latent subspaces. Often, there is considerable interplay between levels in this hierarchy. For example, in split-and-merge heuristics for mixture density estimation (e.g. [93]), components are split once their densities grow wide or unusually elongated, while components are merged once their densities overlap strongly. In this example, the prior distributions over the model families for the components favour more concentrated over strongly elongated distributions, while the prior on the number of components favours small numbers.

1.2 Supervised learning aided by additional unlabeled data

There are problems which do not belong to *either* of the principal classes discussed in the previous subsection 1.1, and they are of immense practical importance. For example, we might face a supervised classification problem for the relationship $P(\boldsymbol{x}, t)$ for which it is easy to obtain a large sample of *unlabeled* data, but the process of labeling points $\boldsymbol{x}$ drawn from $P(\boldsymbol{x})$, according to $P(t|\boldsymbol{x})$, is very expensive, computationally hard or difficult to perform out of other reasons. For example, the labeling might require human insight, such as in speech recognition, object recognition in images or classifying hypertext pages, or the performance of expensive tests or experiments, such as in medical diagnosis or functional proteomics[5]. In short, the practical interest in methods to attack the problem of *supervised learning aided by additional unlabeled data* (short: the *labeled-unlabeled problem*), to be defined in the following, is considerable.

Given an unknown probabilistic relationship $P(\boldsymbol{x}, t)$ between input points $\boldsymbol{x}$ and class labels $t \in T = \{1, \dots, c\}$, the problem is to *predict* t from $\boldsymbol{x}$, i.e. to find a *predictor* $\hat{t} = \hat{t}(\boldsymbol{x})$ such that the *generalization error* of $\hat{t}$,

$$P_{\mathbf{x},t}\left\{\hat{t}(\boldsymbol{x}) \neq t\right\}, \tag{1}$$

is small, ideally close to the *Bayes error*, being the minimum of the generalization errors of all predictors. We are looking for algorithms to compute $\hat{t}$ from

- a *labeled* sample $D_l = \{(\boldsymbol{x}_i, t_i)|i = 1, \dots, n\}$, where the $(\boldsymbol{x}_i, t_i)$ are drawn independently from $P(\boldsymbol{x}, t)$,
- an *unlabeled* sample $D_u = \{\boldsymbol{x}_i|i = n+1, \dots, n+m\}$, where the $\boldsymbol{x}_i$ are drawn independently from the marginal input distribution $P(\boldsymbol{x}) = \sum_{t=1}^{c} P(\boldsymbol{x}, t)$. D_u is sampled independently from D_l,
- prior knowledge (or assumptions) about the unknown relationship.

If D_u is empty, this is the traditional supervised learning problem. The most interesting case from a practical viewpoint arises for $n = |D_l|$ rather small and $m = |D_u| \gg n$.

Let us define some additional notation which we will use throughout the paper. Define $\boldsymbol{X}_l = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$, $T_l = (t_1, \dots, t_n)$, so that $D_l = (\boldsymbol{X}_l, T_l)$. Let $\boldsymbol{X}_u = (\boldsymbol{x}_{n+1}, \dots, \boldsymbol{x}_{n+m})$, i.e. $D_u = \boldsymbol{X}_u$. Furthermore, denote the missing labels on the points from D_u by $T_u = (t_{n+1}, \dots, t_{n+m})$. The combined evidence (i.e. the complete observed data) is $D = (D_l, D_u)$.

Availability of *prior knowledge* about the relationship (often in a form of Occam's razor) is crucial, as argued in subsection 1.1. However, it is very important to note that prior knowledge (or assumptions) are used to a quite different degree and with different final impact, if we compare supervised and unsupervised

[5] Here, one tries to deduce the function of certain proteins in a cell. In the moment, the ultimate, but often very difficult and expensive method to do this is to grow a crystal, and then to determine the three-dimensional protein structure using x-ray crystallography. Other features, such as the expression level of the protein in a certain type of cell under certain conditions, its linear amino acid sequence or flow characteristics under gel electrophoresis, can be determined very much cheaper and on a large scale, often fully automatized.

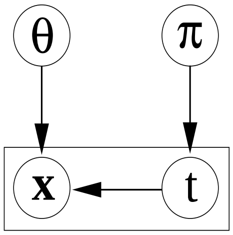
learning. In supervised learning, prior knowledge is used merely as a kind of "security belt", to prevent the algorithm to run ahead and use its most fancy model to fit all bits and pieces of the dataset. In the limit of a large labeled dataset, this belt becomes looser and looser, until its impact on the final prediction almost vanishes. In unsupervised learning, prior assumptions will always have a strong impact on the final result. There is not something like an "a-priori interesting structure" in data, i.e. any kind of structure we discover in data always depends on our view on the examples, e.g. on the features we use to describe them or the distance we use to relate them. Having made this observation, for any algorithm to attack the general problem of supervised learning with additional unlabeled data, it is crucial to balance the impact of prior assumptions very carefully between these two extremes.

1.3 Paradigms for supervised classification

The basic paradigms for supervised classification have already been introduced in subsection 1.1. Here, we discuss them in more detail and describe the role unlabeled data plays in each of them. The relative merits of typical methods within the paradigm, especially w.r.t. possible extensions of such methods to solve the labeled-unlabeled problem are discussed in subsection 5.2.

1.3.1 The sampling paradigm

We refer to architectures following the sampling paradigm as *generative methods*. Within such, we model the class distributions $P(\boldsymbol{x}|t)$ using model families $\{P(\boldsymbol{x}|t,\boldsymbol{\theta})\}$, furthermore the class priors $P(t)$ by $\pi_t = P(t|\boldsymbol{\pi})$, $\boldsymbol{\pi} = (\pi_t)_t$. We also refer to an architecture of this type as a *joint density model* architecture, since we are modeling the full joint density $P(\boldsymbol{x},t)$ by $\pi_t P(\boldsymbol{x}|t,\boldsymbol{\theta})$. For any fixed $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\pi}}$, an estimate of $P(t|\boldsymbol{x})$ can then be computed by Bayes' formula:



$$P(t|\boldsymbol{x},\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\pi}}) = \frac{\hat{\pi}_t P(\boldsymbol{x}|t,\hat{\boldsymbol{\theta}})}{\sum_{t'=1}^{c} \hat{\pi}_{t'} P(\boldsymbol{x}|t',\hat{\boldsymbol{\theta}})}. \tag{2}$$

Alternatively, one can obtain the Bayesian predictive distribution $P(t|\boldsymbol{x},D_l)$ by averaging $P(t|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\pi})$ over the posterior $P(\boldsymbol{\theta},\boldsymbol{\pi}|D_l)$. Within the sampling paradigm, a model for the marginal $P(\boldsymbol{x})$ emerges naturally as

$$P(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\pi}) = \sum_{t=1}^{c} \pi_t P(\boldsymbol{x}|t,\boldsymbol{\theta}). \tag{3}$$

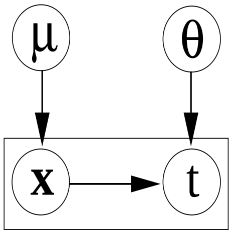
Therefore, if labeled and unlabeled data is available, a natural criterion to maximize would be the *joint log likelihood* of both D_l *and* D_u,

$$\sum_{i=1}^{n} \log \pi_{t_i} P(\boldsymbol{x}_i | t_i, \boldsymbol{\theta}) + \sum_{i=n+1}^{n+m} \log \sum_{t=1}^{c} \pi_t P(\boldsymbol{x}_i | t, \boldsymbol{\theta}), \tag{4}$$

or alternatively the posterior $P(\boldsymbol{\theta}, \boldsymbol{\pi} | D_l, D_u)$.[6] This is essentially an issue of maximum likelihood in the presence of missing data (treating t as latent variable), which can in principle be attacked by the *expectation-maximization (EM)* algorithm (see subsection 2.2).

While the implicit representation of $P(\boldsymbol{x})$ is appealing, generative methods often exhibit significant drawbacks on supervised learning tasks, and we expect some of these to be even more expressed if we add unlabeled data in the learning process, as described above. We discuss some of these issues in subsection 5.2. Furthermore, even in situations where the joint density model families are appropriate, EM might exhibit severe local maxima problems, as discussed in subsection 2.2.

1.3.2 The diagnostic paradigm



In *diagnostic methods*, we model the conditional distribution $P(t|\boldsymbol{x})$ directly using the family $\{P(t|\boldsymbol{x}, \boldsymbol{\theta})\}$, as discussed in subsection 1.1. 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 t 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

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

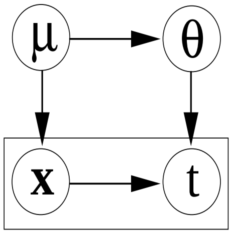
which implies that $P(\boldsymbol{\theta}|D_l, D_u) \propto P(T_l|X_l, \boldsymbol{\theta})P(\boldsymbol{\theta})$, i.e. $P(\boldsymbol{\theta}|D_l, D_u) = P(\boldsymbol{\theta}|D_l)$, and $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are *a-posteriori independent*. Furthermore, $P(\boldsymbol{\theta}|D_l, \boldsymbol{\mu}) = P(\boldsymbol{\theta}|D_l)$. 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 for diagnostic methods, unlabeled data cannot be used for Bayesian inference, and modelling the input distribution $P(\boldsymbol{x})$ is not necessary. The advantages and drawbacks of diagnostic methods, as compared to generative ones (see subsection 1.3.1), are discussed in subsection 5.2. In order to make use of unlabeled data in diagnostic methods, the data generation model discussed above has to be modified, and this is the topic of the next subsection.

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

1.3.3 Regularization depending on the input distribution

We have seen in subsection 1.3.2 that within traditional diagnostic methods for classification, we cannot make use of additional unlabeled data D_u. The principal reason for this is that $\boldsymbol{\theta}$ (to model $P(t|\boldsymbol{x})$) and $\boldsymbol{\mu}$ (to model $P(\boldsymbol{x})$) are a-priori independent. In other words, the model family $\{P(t|\boldsymbol{x}, \boldsymbol{\theta})\}$ is regularized *independently* of the input distribution.

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



We conclude that to make use of additional unlabeled data within the context of diagnostic 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.* This argument is explored in more detail in [81]. We remark that while the modification to the standard data generation model for diagnostic methods suggested here is straightforward in principle, choosing appropriate conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ which on the one hand represent available prior knowledge in an appropriate way[7], on the other hand render the whole inference machinery tractable, at least in an approximative sense, can be very challenging. An important example is given by the *Co-Training* paradigm (see subsection 3.3). Here, the idea of exploiting redundancies between two or more views on examples is used to regularize a hypothesis class based on information about $P(\boldsymbol{x})$. This idea, which originates from earlier work on unsupervised learning, can be seen as a quite general way to construct conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ for a task at hand, although a reasonably general formulation of such a construction process has not yet been given (to our knowledge). Some ideas in that direction can be found in [81].

Some readers might feel a bit uneasy at this point. If we use a-priori dependent $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$, the final predictive distribution *depends* on the prior $P(\boldsymbol{\mu})$ over the input distribution. This forces us to model the input distribution itself, in strong contrast to the situation for traditional diagnostic methods (see subsection 1.3.2). In this case, will our method still be a diagnostic one? Diagnostic methods have the clear advantage over generative techniques that they often require orders of magnitude less free parameters which have to be adjusted while learning from data (see subsection 5.2). However, we do not model each individual class distribution, but the marginal of $\boldsymbol{x}$ only. Furthermore, if the

[7] As discussed in detail in [81], $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ should enforce prior assumptions in a way which is neither too restrictive, so as not to introduce a systematical bias, nor too loose, so that the prior can be expected to have sufficient impact on the final prediction.

input-dependent regularization is done sensibly (see [81] for a discussion), the impact of an oversimple model for $P(\boldsymbol{x})$ on the final prediction is much less severe than in typical methods belonging to the sampling paradigm. Finally, it is generally assumed that unlabeled data is abundant, therefore in theory we need not restrict ourselves to simple models for $P(\boldsymbol{x})$. To conclude, while it is true that "diagnostic" techniques which use input-dependent regularization, share the need for density modeling with generative methods, in our opinion they should still be classified as belonging to the diagnostic paradigm[8].

Theoretical studies of supervised learning methods within the *probably approximately correct (PAC)* framework (e.g. [51]) focus on diagnostic schemes and consequently ignore the input distribution $P(\boldsymbol{x})$ in that they either do not restrict it at all or assume it to be uniform over X. The " ... question of how unlabeled examples can be used to augment labeled data seems a slippery one from the point of view of standard PAC assumptions" (citation from Blum and Mitchell [12]). PAC bounds analyze deviations between training and generalization error for *certain* predictors, drawn from a hypothesis set of limited complexity. Complexity measures for hypothesis sets, such as the *Vapnik-Chervonenkis (VC) dimension* (see [96]), usually do not depend on the input distribution. In other words, a PAC result applies *uniformly* for *any* distribution $P(\boldsymbol{x}, t)$, which is nice, but it merely bounds the probability of drawing an i.i.d. sample of a given size from $P(\boldsymbol{x}, t)$ and then measuring a "more-than-ε" deviation between training error (on this sample) and generalization error for any of the hypotheses from the restricted class. If *all* the hypotheses in this class have unacceptably high training error on the *given* training sample, the only thing we can do is to make the hypothesis class larger and more complex, leading to a worse large deviation bound. In most practical real-world applications, where samples do not have astronomical size, even the best known PAC bounds on the generalization error are usually ridiculously far from being tight.

However, in principle nothing stops us from considering PAC bounds which do not hold *uniformly* for all $P(\boldsymbol{x})$. Although such bounds would hold for *all* $P(\boldsymbol{x}, t)$, their value would *depend* on characteristics of D_u, therefore on $P(\boldsymbol{x})$. Indeed, the bounds given in [74] can be interpreted in this sense. Such bounds might be tighter than uniform ones in cases where $P(\boldsymbol{x}, t)$ does not strongly violate our prior assumptions. Such bounds could then possibly be used to motivate regularization depending on the input distribution.

There are principled frameworks other than Bayesian analysis with conditional priors that attack the labeled-unlabeled problem. Some of them will be discussed in the review section 3 of this paper. For example, subsection 3.6 discusses *restricted Bayes optimal classification* [91] and relates it directly to input-dependent regularization.

[8] In the same way as *restricted Bayes optimal classification* is considered to be a diagnostic technique (see subsection 3.6).

1.4 Overview of the paper

We conclude this introductionary section by giving an overview of the remainder of the paper. Although the essential aim of this paper is to give a review of the literature on the labeled-unlabeled problem, parts of this work emphasize and reflect the author's subjective beliefs in what is important and original versus what should be criticized and how. We would be very happy to get into discussion with readers of this paper, and to validate our views based on their arguments. However, we have done our best to report and to argue fair. We do not claim to be exhaustive, especially not with respect to less recent work. Castelli and Cover [16] collect some references to older work.

Some important topics have been "outsourced" into separate papers, such as the issue of input-dependent regularization of conditional density models (see [81] and subsection 1.3.3). Others are subject of current work, such as the generalization of Fisher kernels mentioned briefly in subsection 3.5. We would also like to emphasize that the paper is somewhat *biased* towards mentioning recent work done by the author himself. Besides giving a comprehensive literature review, another aim of this paper is to build a frame around this work, to put it into a context and onto a basis, all within the effort to give an overview over the first year of the author's PhD period.

In section 2, we identify baseline methods for the problem of supervised learning aided by additional unlabeled data (the "labeled-unlabeled problem"). These methods attack the problem in a generic way and are straightforward transformations of already existing standard methods (for supervised or unsupervised learning) to the new problem domain. Any method claiming to solve the labeled-unlabeled problem should ideally be compared with all of them. Section 3 contains the literature review and forms the main part of the paper. Section 4 describes some problems that we think are related to the labeled-unlabeled problem, together with selected work done on these problems. In section 5, we discuss some caveats and tradeoffs linked with the labeled-unlabeled problem. Section 6 presents conclusions.

2 Baseline methods

In this section, we define some baseline methods which can in principle be used to attack any realization of the labeled-unlabeled problem, as defined in subsection 1.2. The criteria used to select these methods are not rigid, but some of them might be:

1. The method is generic, i.e. not only applicable to a special task.
2. The method is a relatively straightforward transformation of already existing standard techniques for supervised and/or unsupervised learning.

It is our opinion that these baseline methods, or straightforward variants thereof, should not be considered as solutions to the labeled-unlabeled problem, even though they might work well on certain tasks. Each of them has several severe shortcomings (these will be discussed) which, in our opinion, have to be addressed using (probably) genuinely new ideas. However, baseline methods are very useful to compare genuinely new techniques against.

The simplest baseline method is of course to discard the unlabeled data and to predict based on the labeled training data D_l (and the available prior knowledge) alone, using our favourite supervised algorithm. However, some authors seem to overemphasize the importance of this baseline method. For example, if $n = |D_l|$ is rather small[9], most supervised methods will naturally perform poorly. Therefore, if a labeled-unlabeled algorithm outperforms this baseline method significantly, we cannot necessarily conclude that the labeled-unlabeled algorithm is suitable to solve our problem, but rather that the problem cannot be sensibly attacked based on the sparse data D_l only. Another, possibly more subtle issue is the use of *cross-validation* to set free parameters in supervised learning methods. Cross-validation on small training sets is doomed to fail due to very high variance. In our opinion, the only sensible way to predict from small training sets is by using Bayesian inference together with any kind of available prior information.

On certain, artificially created tasks we can compare our labeled-unlabeled method against the "non-plus-ultra" method, namely a supervised algorithm which is given D_l, D_u *and* the missing labels on D_u. This is of course not a baseline method, but it might be quite useful in a case study to set limits. Expecting that our method comes very close to this ideal for rather small D_l is naive, though. Although our aim is the same as for a supervised method, the given data information is much closer to an unsupervised setting.

Some readers might object that we are presenting or suggesting some methods here without having tested them on data. However, most of the methods presented here have been proposed in the literature, at least on special tasks. This will be made clear in the review section 3 of this paper. Second, rather than concrete baseline algorithms we suggest methods or schemes. The concrete realization is left to anyone who might want to use them for comparative studies.

[9] It is difficult to get a "working definition" for what we mean by "rather small" D_l. If our model and prior assumptions for $P(\boldsymbol{x}, t)$ are correct, we could define n to be a rather small dataset size if Bayesian analysis using these assumptions performs on average significantly worse than the (optimal) Bayes classifier which achieves the Bayes error. In practice, it is necessary to study *learning curves* for both the supervised baseline method and the labeled-unlabeled algorithm, in which test errors (averaged over some trials) are plotted against the relative size of D_l, i.e. the percentage of labeled data, $n/(n+m)$.

2.1 Unsupervised learning, followed by assignment of clusters to classes

Let us assume that $m \gg n$, i.e. while labeled data is sparse, unlabeled data is abundant. In this case, we can use a sophisticated unsupervised algorithm to generate a model (or a posterior distribution over models) which fits the unlabeled data only. The design of the unsupervised method of course depends on the prior knowledge we have about $P(\boldsymbol{x}, t)$. For example, assume $X = \mathbb{R}^d$ and the validity of the "cluster assumption", namely that two points $\boldsymbol{x}, \boldsymbol{x}'$ should have the same label t if there is a path between them in X which passes only through regions of relatively high $P(\boldsymbol{x})$. Recently, Tipping [87] and Rattray [68] proposed ways to construct sophisticated distance measures which are aimed at finding such clusters. To this end, we first fit a Gaussian mixture model to the unlabeled data (the number of mixture components can be much larger than the number of classes $c = |T|$). From the fitted model, the distance between any two points can then be computed. We can now use a simple algorithm like k-nearest neighbor (see [28]) together with the labeled data D_l and the distance inferred from D_u. Note that the "cluster assumption" is a very general and weak assumption, therefore applicable as prior assumption to many unsupervised tasks. If prior knowledge of a stronger nature is available, it might be possible to use simpler distance measures, but it is important that the distance can be learned from the unlabeled data D_u.

In case of $X = \mathbb{R}^d$, prior knowledge about the task might allow us to assume that data from each class can faithfully be modeled as coming from an underlying low-dimensional manifold (or, more generally, from a mixture of such manifolds), convolved with Gaussian noise[10]. The *generative topographic mapping (GTM)* [10] is a very powerful architecture in such situations, obtaining the latent manifold as a smooth nonlinear mapping of a uniform distribution over a low-dimensional space, represented by a regular grid. One could try to fit a mixture of GTM's with a rather small number of components K (however, $K \geq c = |T|$) to D_u, keeping the component manifolds smooth using Occam priors[11].

An even easier method can be constructed by imposing the existence of a (latent) *separator variable* k between $\boldsymbol{x}$ and t, i.e. $\boldsymbol{x}$ and t are conditionally independent given k. k lives in $\{1, \dots, K\}$, where typically $K > c = |T|$. We first fit a mixture model to the relationship between $\boldsymbol{x}$ and k (where k is the variable selecting the component). Then, we fix $P(k)$ as well as the component models $P(\boldsymbol{x}|k)$ and train the $P(t|k)$, e.g. by maximizing the likelihood of the labeled data D_l. This technique is further discussed in subsection 2.3, giving rise to another baseline method.

[10] One can use spherical Gaussian noise, but a more reasonable model would be to use Gaussians whose principal axes are aligned with the tangent space of the manifold at each center. Both alternatives are discussed in [10].

[11] However, fitting mixtures of GTM has been proved very difficult in practice (thanks to Chris Williams for pointing this out).

The conceptually simple method based on the cluster assumption might work surprisingly well on real-world problems. It is, however, restricted to the case $X = \mathbb{R}^d$, and the computation of the distances mentioned is computationally quite heavy. A more fundamental aspect with respect to the labeled-unlabeled problem is that the cluster assumption is usually not true *everywhere* in the region of interest. Now, even if the labeled dataset D_l is small, it might point us to such critical places, e.g. suggest splitting a cluster even though it is not crossed by low-density regions of $P(\boldsymbol{x})$. Ideally, the final distance should depend on this information. One could for example start with the distance based on D_u only, then carefully "inject" the label information, thereby modifying the distance at places where label evidence suggests so, leaving it unchanged everywhere else. We are not aware of any principled work been done in this direction.

The technique based on the separator variable k is straightforward to run. However, we do not expect it to work very well in general, given that standard Gaussian mixture models are used to fit $P(\boldsymbol{x})$. One strength of mixture models for density estimation is that many simple component densities can "connect" together to model quite complicated, maybe elongated, connected high-density regions. However, this notion of connectedness is not supported by the method at all. Therefore, in general, connected high-density regions will only be labeled consistently correct if labeled data falls within most of the components modeling the region. This is improbable if D_l is small.

The (hypothetical) mixture of GTM's method would alleviate this problem significantly in that here, centers belonging to the same cluster (i.e. being assigned to the same component GTM) are constrained to lie on the same smooth low-dimensional manifold. Unfortunately, inference is computationally quite expensive for GTM's, growing exponentially in the number of dimensions of the latent manifold. However, simple extensions of GTM, possibly employing more elaborate noise models could be used as very powerful component models in a mixture approach *even if* each component is restricted to a small latent dimensionality.

We finally mention a general idea for "injecting" the label information from D_l *after* having learned a probabilistic partitioning $P(k|\boldsymbol{x}, \hat{\boldsymbol{\theta}})$ of X from D_u. This idea is in line with a method suggested in subsection 2.3 and works by fitting simple local "experts" $P(t|\boldsymbol{x}, k, \boldsymbol{\tau})$ (e.g. logistic regression) to data in the different clusters, where $\boldsymbol{\tau}$ is the parameter vector. Expert k is trained on a reweighted version of D_l, namely each point $(\boldsymbol{x}_i, t_i)$ is weighted by $P(k|\boldsymbol{x}_i, \hat{\boldsymbol{\theta}})$. For example, suppose that $T = \{-1, +1\}$ and $P(t|\boldsymbol{x}, k, \boldsymbol{\tau}) = \sigma(t(\boldsymbol{\omega}_k^T \boldsymbol{x} + b_k))$, where $\sigma(u) = 1/(1 + \exp(-u))$ is the *logistic function*, and $\boldsymbol{\tau} = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_K, b_1, \dots, b_K)$. Fitting such logistic regression models to maximize the likelihood of weighted data can by done by the *iteratively reweighted least squares (IRLS)* technique (see e.g. [56]). In our case, the data is sparse, so one would also employ an Occam prior $P(\boldsymbol{\omega}) = N(\boldsymbol{\omega}|\mathbf{0}, \alpha\mathcal{I})$. If we are only after an MAP approximation to full Bayesian analysis[12], we can still use the IRLS method to compute this. The

[12] The MAP approximation to Bayesian analysis is briefly discussed (in another context) in subsection 2.4. Details can be found e.g. in [53].

advantage of this method over labeling the clusters by assuming that k acts as separator between $\boldsymbol{x}$ and t, as discussed above in this subsection, is that clusters *can* be split between several classes if the labeled data suggests so. Note that if all points in D_l which have significant weight for cluster k, belong to one class, by our Occam prior on $\boldsymbol{\omega}$ the MAP model $P(t|\boldsymbol{x}, k, \hat{\boldsymbol{\tau}})$ will be almost constant.

2.2 Expectation-maximization on a joint density model

Rather than treating D_u as genuinely unlabeled data, we can also view the labels on these points as *missing data. Expectation-maximization (EM)* (see [26],[2]) is a general technique for maximum likelihood estimation in the presence of latent variables or missing data. The idea of the basic batch version of EM is simple. We can distinguish between a *complete* likelihood function over observed and unobserved data and a *marginal* likelihood function which is obtained from the complete one by integrating the latent variables out. The goal is to maximize the marginal likelihood. This is done by iterating the following two steps. In the so-called *E step*, we compute the conditional distribution of the latent variables, given the observed data and the current model estimate. In the *M step*, we compute the expectation of the complete log likelihood function under this conditional distribution, and then choose a new model which maximizes this criterion. To be more specific, let $\boldsymbol{z}_v$ be the observed, $\boldsymbol{z}_h$ be the hidden variables. By *Jensen's inequality* (e.g. [21]), applied to the concave log, we have

$$\log P(\boldsymbol{z}_v|\boldsymbol{\theta}) = \log \int P(\boldsymbol{z}_v, \boldsymbol{z}_h|\boldsymbol{\theta})\, d\boldsymbol{z}_h \geq E_{\boldsymbol{z}_h \sim Q(\boldsymbol{z}_h)} \left[\log \frac{P(\boldsymbol{z}_v, \boldsymbol{z}_h|\boldsymbol{\theta})}{Q(\boldsymbol{z}_h)} \right] \tag{5}$$

for *any* distribution $Q(\boldsymbol{z}_h)$. For fixed $\boldsymbol{z}_v = \bar{\boldsymbol{z}}_v$ and the current model estimate $\hat{\boldsymbol{\theta}}$, we choose $Q(\boldsymbol{z}_h) = P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}})$ in the E step in order to achieve the tightest possible bound. EM can therefore be seen as successive maximization of (varying) lower bounds to the marginal log likelihood. A crucial fact about the EM criterion as a lower bound to the marginal log likelihood is that they are equal to first order if expanded w.r.t. $\boldsymbol{\theta}$ around $\hat{\boldsymbol{\theta}}$, and one can show that a local maximum point $\hat{\boldsymbol{\theta}}$ of the bound also maximizes $\log P(\boldsymbol{z}_v|\boldsymbol{\theta})$ locally.

It springs into mind to construct a model family for the joint distribution $P(\boldsymbol{x}, t)$ and to determine a model maximizing the *joint likelihood* by using the EM algorithm, in order to attack the labeled-unlabeled problem. This is most easily be done by choosing model families for the class-conditional distributions $P(\boldsymbol{x}|t)$. The joint log likelihood is

$$\sum_{i=1}^{n} \log \pi_{t_i} P(\boldsymbol{x}_i|t_i, \boldsymbol{\theta}) + \sum_{i=n+1}^{n+m} \log \sum_{t=1}^{c} \pi_t P(\boldsymbol{x}_i|t, \boldsymbol{\theta}), \tag{6}$$

where $\pi_t = P(t|\boldsymbol{\theta})$. The derivation of the EM equations then parallels very closely the case of mixture models, which can be found in many textbooks, e.g. [11].

Indeed, using EM to “fill in” labels on D_u has already been suggested very early, namely in a note by R. J. Little in the discussion of [26]. Chapter 1.11 of [58] gives the idea and further references, however it is not clear whether the authors suggest using the approach for classification or merely for partially unsupervised learning, where unsupervised fitting of a mixture model to $P(\boldsymbol{x})$ is aided by a few labeled points D_l. It has been used to attack the labeled-unlabeled problem, e.g. for text classification [65]. However, usage of EM in this context is somewhat dangerous, as we will argue next. First of all, we are required to model the class-conditional distributions. In the terminology of subsection 1.3, we operate within the sampling paradigm and not within the often more robust diagnostic paradigm. It has frequently been observed that, in the purely supervised setting, fitting the class-conditional distributions with rather poor models, then estimating $P(t|\boldsymbol{x})$ by using these models in Bayes' formula, works surprisingly well w.r.t. prediction, but poor as estimate of $P(t|\boldsymbol{x})$, in that the estimates at most points $\boldsymbol{x}$ have too low entropy (i.e. the prediction is too confident at most points). Suppose we choose quite narrow families including only simple models for the classes. We initialize EM by fitting the models to the labeled data D_l. In the first E step, the missing labels are “filled in” by their expected values, given the current model and the observed data. By the overconfident nature of the estimates based on the poor class models, these “pseudo-labels” will be quite definitive on many of the points from D_u. In other words, the first E step will assign a large number of points from D_u to classes quite confidently, based only on the initial poor model fitted to D_l. In the subsequent M step, these artificially labeled points can outweigh the labeled points from D_l, leading to a model which might even exhibit worse predictive performance than the initial one. In any case, we expect EM applied in this way to quickly converge into a poor local maximum of the joint likelihood, largely determined by the “pseudo-labels” given to the points from D_u during the *first* E step. This problem could be alleviated by allowing more complex class density models. However, in this case it is not clear how to fit these models initially if D_l is small. A typical situation where straightforward EM fails empirically, is shown in [64].

2.2.1 A general view on expectation-maximization techniques

This subsection contains advanced material not required for the general understanding of the remainder of the paper. Although we think the view on EM techniques presented here is very useful for anybody who applies the standard EM algorithm or variants to learning problems, the reader is invited to jump to the last paragraphs of this subsection, where we state the consequences of the view relevant for this paper. In later versions of this paper, the present subsection will probably be “outsourced” into a separate, more comprehensive technical report.

Although the standard batch EM algorithm, as applied straightforwardly to the labeled-unlabeled problem, suffers from severe robustness problems mentioned above, other notions of EM might be much better suited to attack the problem. The key here is to adopt a very general view on EM procedures which allows us to modify the standard algorithm in a variety of ways without losing the convergence guarantees.

EM is a special case of an *alternating minimization procedure* in the context of information geometry (see [22]), as has been observed by several authors (e.g. [2],[41]). Several important problems in information theory, such as computation of the capacity of a (discrete memoryless) channel or of the rate-distortion function, can be shown to be equivalent to the following problem (see e.g. [21]): given two convex sets $\mathcal{Q}$ and $\mathcal{P}$ of distributions over $\boldsymbol{z} \in Z$, what is the minimum divergence $\min_{Q\in\mathcal{Q},P\in\mathcal{P}} D(Q\|P)$ between them, and for which $Q^* \in \mathcal{Q}, P^* \in \mathcal{P}$ this minimum distance is attained? Here, the divergence D is given by the *relative entropy* (or *Kullback-Leibler divergence*)

$$D(Q \,\|\, P) = E_Q\left[\log \frac{Q(\boldsymbol{z})}{P(\boldsymbol{z})}\right]. \tag{7}$$

D is a very useful divergence measure between probability distributions with a clear information-theoretic interpretation (see [21]) and strong, yet somewhat deep, motivations through information geometry (e.g. [2]). Since D is convex in both arguments, the solution to this problem is unique, and the following very simple alternating minimization procedure is guaranteed to find it: start with some $Q \in \mathcal{Q}, P \in \mathcal{P}$, then alternate *E steps* in which $Q \leftarrow \operatorname{argmin}_{Q\in\mathcal{Q}} D(Q\|P)$, and *M steps* in which $P \leftarrow \operatorname{argmin}_{P\in\mathcal{P}} D(Q\|P)$. The minimization procedures (as well as their outcomes in this context) in E and M step are called *e-projection* and *m-projection* (e.g. [2]). These steps are iterated until no more improvement in $D(Q\|P)$ is observed.

EM can be seen as a variant of this algorithm, as will be shown next. In this context, $\mathcal{P}$ will be a family of models for $\boldsymbol{z}$, while $\mathcal{Q}$ contains distributions related to the empirical distribution of the data, as determined by the observed sample. Unfortunately, in all nontrivial applications of EM, it turns out that $\mathcal{P}$ is *not* a convex set[13], therefore there can be many global solutions, and the algorithm will in general not converge to any of these. However, given sufficient smoothness conditions on $\mathcal{P}$,[14] the algorithm finds a *local solution*, i.e. a pair (Q^*, P^*) which minimizes $D(Q\|P)$ among all $Q \in \mathcal{Q}$ and $P \in \mathcal{P}$ in an "environment" of P^*.[15]

Let $\mathcal{S}$ be a convex manifold[16] of distributions $P(\boldsymbol{z})$ over Z. Let $\boldsymbol{z} = (\boldsymbol{z}_v, \boldsymbol{z}_h)$ where $\boldsymbol{z}_v$ is visible, $\boldsymbol{z}_h$ is hidden. Define the *model submanifold* $\mathcal{P}$ as manifold em-

[13] In many cases, it is the direct product of several convex sets. EM can also be regarded as alternating minimization procedure between three or more convex sets. As an aside, the recently proposed *information bottleneck* learning algorithm can also be regarded as such a procedure between three convex sets, therefore has the same theoretical basis than the EM algorithm (see [89]).

[14] We do not discuss these conditions in detail here, they are usually fulfilled in practice.

[15] To be able to talk about "smoothness" and "environments", we first have to impose a *manifold structure* on $\mathcal{P}$. In the context of EM, this is usually done by defining $\mathcal{P}$ to be a *model family* parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$ or some submanifold thereof.

[16] We shall not use geometrical properties of $\mathcal{S}$ here.

bedded in $\mathcal{S}$ and parameterized by $\boldsymbol{\theta}$. The EM algorithm, as described at the beginning of subsection 2.2, is an iterative procedure to, given a sample $\bar{\boldsymbol{z}}_v$ from $\boldsymbol{z}_v$, find a (local) maximum $\hat{P}(\boldsymbol{z}) = \hat{P}(\boldsymbol{z}|\hat{\boldsymbol{\theta}}) \in \mathcal{P}$ of the *marginal likelihood function* $P(\boldsymbol{z}) \mapsto P(\bar{\boldsymbol{z}}_v) = \int P((\bar{\boldsymbol{z}}_v, \boldsymbol{z}_h))\, d\boldsymbol{z}_h$. Now define the *(universal) data manifold* $\mathcal{Q}_U$ to contain all $Q(\boldsymbol{z}) \in \mathcal{S}$ such that the marginal $Q(\boldsymbol{z}_v) = \int Q((\boldsymbol{z}_v, \boldsymbol{z}_h))\, d\boldsymbol{z}_h$ is equal to the (marginal) empirical point distribution $\delta(\boldsymbol{z}_v, \bar{\boldsymbol{z}}_v)$ of $\bar{\boldsymbol{z}}_v$.[17] $\mathcal{Q}_U$ is clearly convex. It can be seen as to contain all possible beliefs about the complete data after having observed $\boldsymbol{z}_v = \bar{\boldsymbol{z}}_v$. Now, to show the equivalence of EM with the alternating minimization procedure discussed in this section, we first look at the E step. Let $\hat{P}(\boldsymbol{z}) = P(\boldsymbol{z}|\hat{\boldsymbol{\theta}})$ denote the current model, and we are looking for the e-projection $\hat{Q}$, i.e. $\hat{Q} \in \mathcal{Q}_U$ to minimize $D(\cdot\|\hat{P})$. First, we can write $\hat{Q}(\boldsymbol{z}) = \hat{Q}(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v)\delta(\boldsymbol{z}_v, \bar{\boldsymbol{z}}_v)$, by the definition of the data manifold $\mathcal{Q}_U$. Then, it is easy to see that

$$\begin{aligned} D(\hat{Q}\,\|\,\hat{P}) &= \int \hat{Q}(\boldsymbol{z}) \log \frac{\hat{Q}(\boldsymbol{z})}{P(\boldsymbol{z}|\hat{\boldsymbol{\theta}})}\, d\boldsymbol{z} = \int \hat{Q}(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v) \log \frac{\hat{Q}(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v)}{P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}})}\, d\boldsymbol{z}_h + C \\ &= D(\hat{Q}(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v)\,\|\,P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}})) + C, \end{aligned} \tag{8}$$

where C is some constant independent of $\hat{Q}$. By the nonnegativity of the relative entropy, the minimizer is $\hat{Q}(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v) = P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}})$, i.e. the posterior distribution employed by EM in the E step. Furthermore, with this choice of $\hat{Q}$ we have that

$$\begin{aligned} -D(\hat{Q}\,\|\,P(\cdot|\boldsymbol{\theta})) - H(\hat{Q}) &= E_{\hat{Q}}\left[\log P(\boldsymbol{z}|\boldsymbol{\theta})\right] \\ &= E_{\mathbf{z}_h \sim P(\mathbf{z}_h|\bar{\mathbf{z}}_v, \hat{\theta})}\left[\log P((\bar{\boldsymbol{z}}_v, \boldsymbol{z}_h)|\boldsymbol{\theta})\right]. \end{aligned} \tag{9}$$

Since the right-hand side of this equation is the usual EM criterion to be maximized in M step, and $H(\hat{Q})$ does not depend on $P(\cdot|\boldsymbol{\theta}))$, we see that EM performs an m-projection onto $\mathcal{P}$ in the M step.

There are two things to note if one wants to make use of this new view. First of all, the quality of the final solution the algorithm presents, as well as its convergence speed, depends very much on the *initial choice* of the model P. A natural idea is to employ a *sequence* of EM algorithms, all having different model submanifolds $\mathcal{P}$, and using the solution computed by each algorithm as initialization for the next one in the sequence. Since the EM algorithm is iterative anyway, this "chaining" does not even change its character. Of course we have to make sure that the model family $\mathcal{P}$ we are really after stands at the end of the sequence. Several advantages can in principle be realized by employing a suitably chosen sequence instead of one monolithic EM run:

- For our model submanifold $\mathcal{P}$, it might be the case that EM only converges into a satisfactory solution if initialized very carefully, otherwise gets stuck into poor local solutions. Such an initialization often requires an expensive

[17] This distribution concentrates all the mass on the point $\boldsymbol{z}_v = \bar{\boldsymbol{z}}_v$.

search. Another common practice is to start the algorithm a lot of times from different randomly chosen initial points. By employing a cleverly designed sequence (see discussion below), the search can be done much more principled and often more efficient.[18]

- For our $\mathcal{P}$, it might be the case that EM takes an unacceptable long time for convergence, unless it is initialized very carefully. Again, a suitably chosen sequence of individually quickly converging EM runs can often be expected to find a good initialization.

This idea will be generalized below, where we use it to obtain annealed versions of EM.

The second point to note is that one does not necessarily have to perform *complete* projections (i.e. minimizations) in *all* the E and M steps. Equivalently, one can restrict the search for the projections to *subsets* of $\mathcal{Q}_U$ and $\mathcal{P}$ respectively. In order not to get stuck in spurious extrema, we only require that a true reduction in $D(Q\|P)$ is achieved in each step. This means of course that in order to assess final convergence, we have to face optimizations over the full $\mathcal{Q}_U$ and $\mathcal{P}$, but especially in early stages of the run we can get away more cheaply. Such restrictions can of course affect the quality of the final solution as well as the number of iterations needed until convergence, but in many cases the benefits outweight the drawbacks by far.

Having argued in very general terms so far, we will now show how several well-known extensions of EM arise naturally within the view presented above. We will then motivate how some of these might be used to eventually make EM work better on the labeled-unlabeled problem. The (so called) *generalized* EM variant is obtained by allowing partial rather than complete minimization in the M steps. Generalized EM sometimes runs faster due to simpler searches in M steps. However, in general it requires more iterations until convergence than standard EM.

The *variational* variant of EM systematically uses partial minimization in the E steps. This variant is useful in cases where standard EM is computationally infeasible. It replaces the full data manifold $\mathcal{Q}_U$ by a parameterized convex submanifold $\mathcal{Q}$ which is chosen such that minimization of $D(\cdot\|P)$ over $\mathcal{Q}$ as well as computation of the criterion to be optimized in M step are feasible. Note that usually $\mathcal{Q}$ does *not* contain most of the posterior distributions which standard EM employs in the E steps, therefore the variational variant in general does not convergence to a local maximum of the marginal likelihood. However, if $\mathcal{Q}$ is a reasonably broad submanifold of $\mathcal{Q}_U$, the final solution can still be of high quality.

[18] For example, fitting of Gaussian mixtures via EM is often initialized by first fitting a mixture from a restricted model family, such as the family of mixtures of Gaussians with the identity as covariance matrix, or by running the k-means procedure which is a limit case of the latter.

There are *sequential* versions of EM which, in every E step, compute the posterior distributions only over a subset of all latent variables in $\boldsymbol{z}_h$, and use posteriors computed in earlier E steps on the remaining ones. In our view, this corresponds to partial E step minimization as follows. Let Q be the current data distribution and $\hat{P} = P(\cdot|\hat{\boldsymbol{\theta}})$ be the current model. Let $\boldsymbol{z}_h = (\boldsymbol{z}_{h1}, \boldsymbol{z}_{h2})$, and only the posteriors on $\boldsymbol{z}_{h1}$ should be computed. We have to assume that, by choice of the model family $\mathcal{P}$, $\boldsymbol{z}_{h1}$ and $\boldsymbol{z}_{h2}$ are independent in the posterior $P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}})$. Now, we restrict the search for the new data distribution $\hat{Q}$ to distributions of the form $\hat{Q}(\boldsymbol{z}_{h1}|\bar{\boldsymbol{z}}_v)Q(\boldsymbol{z}_{h2})\delta(\boldsymbol{z}_v, \bar{\boldsymbol{z}}_v)$, and under this restriction the e-projection is $P(\boldsymbol{z}_{h1}|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}})Q(\boldsymbol{z}_{h2})\delta(\boldsymbol{z}_v, \bar{\boldsymbol{z}}_v)$, which is what sequential EM variants use.

Note that while we have the freedom to choose the model submanifold $\mathcal{P}$, the universal data manifold $\mathcal{Q}_U$ is fixed by the definition above and the observed data. On many tasks, it can be useful to sensibly restrict $\mathcal{Q}_U$, e.g. to derive variational variants of EM, as discussed above. We define the *data submanifold* $\mathcal{Q}$ of an EM algorithm to be a submanifold of the data manifold $\mathcal{Q}_U$, given the observed data $\bar{\boldsymbol{z}}_v$. An *EM algorithm* is formally defined by the pair $(\mathcal{Q}, \mathcal{P})$. In this definition, we allow $\mathcal{Q}$ to be nonconvex as well as incomplete in the sense that it might not contain all of the (potential) posteriors $P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \boldsymbol{\theta})$. In general, the convergence guarantee of an EM algorithm holds only if $\mathcal{Q}$ is convex and complete.

In general, the EM algorithm suffers from two basic problems. The first one is that it often gets stuck into shallow local optima instead of finding high-quality solutions (of reasonably high marginal likelihood). This is due to the fact that the model submanifold $\mathcal{P}$ is usually not convex. The second one is that on models involving structural choices (such as connectivity in a network), the M step optimizations are often intractably hard. Both problems are especially severe if the model family exhibits symmetries in parameterization on different levels. These problems can in principle be addressed by carefully choosing the initial model P, but on many models this is as difficult as finding a good fit to the data in the first place. A standard technique to attack such problems is *simulated annealing* [52]. In the context of EM, the basic idea is to run a *sequence* of EM algorithms on the data, each having its own model and data submanifold. After convergence of one algorithm, we use the solution to initialize the next one. The "art" is to choose the $(\mathcal{Q}, \mathcal{P})$ sequence in order to achieve a somewhat continuous transition between early stages where hardly any shallow local optima are present, and where it is rather easy to explore large parts of the model family in the M steps, to late stages where model and data manifolds are close to the ones we are aiming for. The successive solutions are, if annealing is done carefully, better and better suited as initial models to guarantee that the final hard EM run will find a reasonably deep optimum. For example, it has been suggested to combine standard EM with *deterministic annealing* (e.g. [101],[92],[71]) to alleviate the local optima problem. In our framework, this can be seen as running a sequence of EM algorithms, all sharing the same model submanifold, but employing different data submanifolds obtained by constraining elements of $\mathcal{Q}_U$ in a particular way. We therefore call it *E-step annealing.* If our model family $\mathcal{P}$

involves structural choices, we can also use *M-step annealing* (see [79]), which involves running a sequence of EM algorithms, all sharing the same data manifold $\mathcal{Q}_U$, but employing different model submanifolds constructed from $\mathcal{P}$ by a process of "controlled randomization".

The last variant to be discussed here could be called *robust EM*, since it is aimed towards alleviating the robustness problems of standard EM on the labeled-unlabeled problem mentioned above (subsection 2.2). Two problems were identified there. First, we need to train models to fit each class distribution separately. Since a class distribution can be very complicated, we would like to use a broad model class. However, complicated models cannot be reliably fit using the sparse labeled data D_l. Second, early "pseudo-labeling" of major parts of the unlabeled points in D_u, based on poor models trained on D_l, can have a devastating effect on the final prediction. A robust variant of EM would start by fitting rather simple class models to D_l. Let us separate D_u into two sets $D_u^{(a)}$ ("active") and $D_u^{(i)}$ ("inactive"). Initially, $D_u^{(a)} = \emptyset$. In a sweep over $D_u^{(i)}$, we extract a few points most confidently labeled to one of the classes by the current model, and place them into the active set $D_u^{(a)}$. We now run EM on the data D_l and $D_u^{(a)}$, i.e. the latent variables are the labels of the points in $D_u^{(a)}$. The more data we are looking at, the more complex models we can consider for fitting, therefore it seems reasonable to broaden the model family slowly while "injecting" more and more points from $D_u^{(i)}$. Formally, this can be incorporated in our view on EM as follows: $\boldsymbol{z}_v$ consists of all variables in D_l and D_u, $\boldsymbol{z}_h$ are the labels corresponding to input points in D_u. Robust EM consists of running a sequence of EM algorithms, each having its own model submanifold. For example, at a certain stage D_u might be divided into $D_u^{(a)}$ and $D_u^{(i)}$. We divide $\boldsymbol{z}_v = (\boldsymbol{z}_v^{(a)}, \boldsymbol{z}_v^{(i)})$ and $\boldsymbol{z}_h = (\boldsymbol{z}_h^{(a)}, \boldsymbol{z}_h^{(i)})$ accordingly, furthermore let $\boldsymbol{z}^{(a)} = (\boldsymbol{z}_v^{(a)}, \boldsymbol{z}_h^{(a)})$, $\boldsymbol{z}^{(i)} = (\boldsymbol{z}_v^{(i)}, \boldsymbol{z}_h^{(i)})$. Then, the model family $\mathcal{P}$ has the general structure

$$\left\{ P(\boldsymbol{z}|\boldsymbol{\theta}) = P(\boldsymbol{z}^{(a)}|\boldsymbol{\theta}) U(\boldsymbol{z}^{(i)}) \right\}, \tag{10}$$

where $U(\cdot)$ denotes the uniform distribution. Each model is therefore really only a model over $\boldsymbol{z}^{(a)}$, since it "models" the remaining variables by the uninformative uniform distribution. By the i.i.d. assumption for D_l, D_u,

$$P(\boldsymbol{z}^{(a)}|\boldsymbol{\theta}) = \prod_{(\mathbf{x}_i, t_i) \in D_l \cup D_u^{(a)}} P(\mathbf{x}_i, t_i|\boldsymbol{\theta}). \tag{11}$$

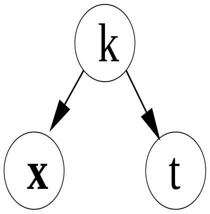
Here, "$(\boldsymbol{x}_i, t_i) \in D_u^{(a)}$" means that $\boldsymbol{x}_i \in D_u^{(a)}$, and t_i is the associated latent label. To specify $\mathcal{P}$, we therefore only need to specify $P(\boldsymbol{x}, t|\boldsymbol{\theta})$. For this specification, we can use broader and broader model classes together with increasing size of $D_u^{(a)}$. Now it is easy to see that the posterior in the E step becomes $P(\boldsymbol{z}_h|\bar{\boldsymbol{z}}_v, \hat{\boldsymbol{\theta}}) = P(\boldsymbol{z}_h^{(a)}|\bar{\boldsymbol{z}}_v^{(a)}, \hat{\boldsymbol{\theta}}) U(\boldsymbol{z}_h^{(i)})$, and the criterion to be maximized in the M step is, up to an additive constant, $E_{P(\mathbf{z}_h^{(a)}|\bar{\mathbf{z}}_v^{(a)}, \hat{\theta})}[\log P(\mathbf{z}_h^{(a)}, \bar{\mathbf{z}}_v^{(a)}|\theta)]$. That is, the components in $\boldsymbol{z}^{(i)}$, representing the data in $D_u^{(i)}$, are simply ignored. A variant of robust EM has been suggested in [64] (the authors call it "self-training").

To conclude, the new view on EM allows us to modify the standard version in numerous ways, many of them are yet unexplored. In the context of the labeled-unlabeled problem, the robust variant of EM discussed in the previous paragraphs might alleviate shortcomings of the standard method. Also, E step annealing (i.e. deterministic annealing) might be useful in this context. Facing the problem that the posterior distribution over the latent variables (i.e. the labels corresponding to the points in D_u) has too low entropy over many of its components during early stages of EM (we have called this "overconfident pseudo-labeling" above), one could use E step annealing to blurr (i.e. "heat up") these posteriors during early iterations. We have not talked specifically about the case that models themselves might incorporate latent variables, e.g. encoding structural choices. In such cases, M step annealing might be helpful.

2.3 Expectation-maximization, using an additional separator variable

In subsection 2.2, we discussed how the EM algorithm together with joint models for $P(\boldsymbol{x}, t)$ could be used to attack the labeled-unlabeled problem. This involves modeling the different classes separately by class-conditional models $P(\boldsymbol{x}|t, \boldsymbol{\theta})$, in which case the marginal model for $\boldsymbol{x}$ is $\sum_t \pi_t P(\boldsymbol{x}|t, \boldsymbol{\theta})$ (see (6) for notations). If the class-conditional models are simple, this might be a poor model family for the marginal distribution.

Another idea, already mentioned in subsection 2.1, is to introduce a (latent) *separator variable* k. Under the model, k separates $\boldsymbol{x}$ and t in the sense that $\boldsymbol{x}$ and t are conditionally independent given k. This means that, under the model, all the information $\boldsymbol{x}$ contains about its class t is already captured in k. This fact is illustrated in the independence model on the right.



Under this modeling assumptions, the joint log likelihood is

$$\sum_{i=1}^{n} \log \sum_{k} \beta_{t_i,k} \pi_k P(\boldsymbol{x}_i|k, \boldsymbol{\theta}) + \sum_{i=n+1}^{n+m} \log \sum_{k} \pi_k P(\boldsymbol{x}_i|k, \boldsymbol{\theta}), \qquad (12)$$

where $\pi_k = P(k|\boldsymbol{\theta})$ and $\beta_{t,k} = P(t|k, \boldsymbol{\theta})$. Again, it is straightforward to compute the EM equations for this model (see [59]). Note that in this case, we do not need to treat the labels of the points in D_u as latent variables. Miller and Uyar [59] present some results using this model together with Gaussian components $P(\boldsymbol{x}|k, \boldsymbol{\theta})$. The "many-centers-per-class" case in [65] can also be seen in this context.

It should be rather straightforward to weaken the assumption of k being a separator variable. This would lead to an architecture in which estimates of $P(t|\boldsymbol{x}, k)$

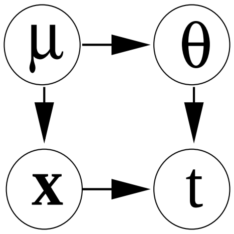
(e.g. by logistic regression) take over the role of the $\beta_{t,k}$. Typically, by the sparseness of D_l, the labeled data will not be sufficient to train these "local" predictors. This problem can possibly be alleviated by regarding the label t as latent variable and applying EM to k *and* t (in the terminology of section 2.2.1, the data distribution Q is kept definitive (or "clamped") on the observed labels). The resulting predictor is $\sum_k P(t|\boldsymbol{x}, k, \hat{\boldsymbol{\theta}})P(k|\boldsymbol{x}, \hat{\boldsymbol{\theta}})$, where $P(k|\boldsymbol{x}, \boldsymbol{\theta}) \propto \pi_k P(\boldsymbol{x}|k, \boldsymbol{\theta})$, similar to a *mixture-of-experts* architecture (see [50], [98]). However, in the latter, the *gating models* for $P(k|\boldsymbol{x})$ are diagnostic rather than generative, and the whole architecture is trained to maximize the *conditional likelihood* of the data rather than the joint one. The relationship between the two architectures is discussed in subsection 5.2.

Use of a separator variable should in most cases outperform straightforward EM (see subsection 2.2) in a comparison where $P(\boldsymbol{x}|k, \boldsymbol{\theta})$ and $P(\boldsymbol{x}|t, \boldsymbol{\theta})$ come from the same model family respectively, simply because k typically ranges over more values than the class label t. For example, if this model family solely contains unimodal distributions, only the version employing k can in principle model multimodal class distributions. However, as already mentioned at the end of subsection 2.1, it is not clear whether this advantage is really substantial. We might for example let k range over a large set of values, in which case we might be able to identify the marginal distribution $P(\boldsymbol{x})$ almost exactly, based on a large D_u only. But since the model does not encode any *prior* "force" to e.g. connect strongly overlapping components (the "cluster assumption", mentioned in subsection 2.1), or relate components a priori in any other way, a lot of labeled data D_l is needed to associate the large number of components with the classes. To be able to make real use of unlabeled data for supervised tasks, we not only have to be able to identify the marginal $P(\boldsymbol{x})$ well, but *it is also required that we identify (to a certain degree) the connected components of the class distributions.* While the former is relatively easy, given that D_u is large, the latter is the real challenge and can only be solved by employing prior knowledge about the nature of the class distributions. Only if we are able to identify the class distributions well, we can hope to realize the "exponential value" of a labeled sample D_l (see [16], which is also discussed in subsection 3.1) to our advantage.

2.4 Expectation-maximization on diagnostic models

The method to be discussed in this subsection differs in several aspects from the other methods presented in this section. First of all, we are not sure that it can be applied reasonably generically. It has been applied successfully to a special task, giving rise to the so-called *Co-Training* paradigm, as will be discussed below. Second, the method we describe here is more an algorithmic scheme than an algorithm. Before this scheme can be applied to a special task, prior knowledge has to be gathered and encoded in a way such that the algorithmic scheme can be realized exactly or approximately in a feasible algorithm. This might need genuinely new ideas, therefore particular instances of the algorithmic scheme, such as Co-Training, are *not* considered to be baseline algorithms.

As discussed in subsecion 1.3.2, probabilistic *diagnostic* methods model $P(t|\boldsymbol{x})$ directly, without using the way over class-conditional density models. Traditionally, regularization of the model class $\{P(t|\boldsymbol{x},\boldsymbol{\theta})\}$ is done independently of the input distribution, in which case unlabeled data D_u does not contain any additional information about the latent $\boldsymbol{\theta}$, given D_l (see subsection 1.3.2). In such a setting, modeling the labels of points in D_u as latent variables makes no sense. Introducing a new point from D_u into the game does not lead to any new constraints of our belief in $\boldsymbol{\theta}$, since there is no information flow betweem D_u and $\boldsymbol{\theta}$. Applying EM to incorporate the latent label of the new point means that we first predict the label, given our old belief about $\boldsymbol{\theta}$, which is independent of the new point. Next, we "update" our belief in $\boldsymbol{\theta}$, based on this prediction. However, since this update only uses information that was already know *before* the new point was introduced, it cannot make our belief in $\boldsymbol{\theta}$ sharper in a data-driven way.



The situation can change if the regularization of the model class depends on the input distribution, as discussed in subsection 1.3.3. To clarify this claim, let us give an example, employing the terminology of subsection 1.3.3. We model $P(\boldsymbol{x})$ by the family $\{P(\boldsymbol{x}|\boldsymbol{\mu})\}$ and an "Occam" prior $P(\boldsymbol{\mu})$ on the parameters $\boldsymbol{\mu}$. Input-dependent regularization means that we employ conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$, in which case $P(\boldsymbol{\theta},\boldsymbol{\mu}) = P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})$ and $P(\boldsymbol{\theta}) = \int P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{\mu})\,d\boldsymbol{\mu}$. The exact Bayesian solution, the predictive distribution $P(t|\boldsymbol{x},D)$, $D = (D_l, D_u)$, can often be approximated by an *maximum a-posteriori (MAP)* solution $P(t|\boldsymbol{x},\hat{\boldsymbol{\theta}})$, where the MAP parameters $(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\mu}})$ maximize the posterior, or equivalently the joint distribution $P(D,\boldsymbol{\theta},\boldsymbol{\mu})$. Recalling the notation defined in subsection 1.2, we have

$$\begin{aligned} &P(D,\boldsymbol{\theta},\boldsymbol{\mu}) = P(T_l|\boldsymbol{X}_l,\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{X}|\boldsymbol{\mu})P(\boldsymbol{\mu}) \\ &= \sum_{T_u} P(T_u|\boldsymbol{X}_u,\boldsymbol{\theta})P(T_l|\boldsymbol{X}_l,\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{X}|\boldsymbol{\mu})P(\boldsymbol{\mu}) \\ &\geq \exp\left[\sum_{T_u} Q(T_u)\log\frac{P(T_u|\boldsymbol{X}_u,\boldsymbol{\theta})P(T_l|\boldsymbol{X}_l,\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\mu})P(\boldsymbol{X}|\boldsymbol{\mu})P(\boldsymbol{\mu})}{Q(T_u)}\right]. \end{aligned} \tag{13}$$

In contrast to the situation where $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are a-priori independent, proposing latent labels T_u and running EM to compute an MAP solution might make sense in this case. The reader might object by noting that since the T_u simply marginalize out in the joint (13), there is really no reason to employ EM: instead of optimizing lower bounds on the marginal likehood and using elaborate sequences of alternating E and M steps, the marginal likelihood can be maximized directly. The answer to this point is the same as the one in the case of fitting mixture models. When fitting a mixture model, we can compute the marginal likelihood and the gradient thereof essentially as easily as computing the statistics needed in the E step of EM. We could therefore use any standard optimizer to maximize the marginal likelihood directly. However, in many cases optimization by EM works more *efficiently* and is conceptually *simpler*. The same facts *can* be true in case of diagnostic models together with input-dependent regularization. Here, the unlabeled points $\boldsymbol{X}_u$ constrain our belief in $\boldsymbol{\theta}$ in a certain way (via the model $\boldsymbol{\mu}$), i.e. they contain *information* about $\boldsymbol{\theta}$. However, *transferring* this information "from $\boldsymbol{X}_u$ to $\boldsymbol{\theta}$" might often be much easier using the EM way over the latent labels T_u. If $P(\boldsymbol{\theta}|\boldsymbol{\mu})$ encodes some subtle contraints on the models $P(t|\boldsymbol{x}, \boldsymbol{\theta})$, enforcing these constraints *directly* in an optimization, together with the "force" to achieve high likelihood on the small set D_l, might be very hard. Robust EM variants (see end of subsection 2.2.1) might be good candidates to attack this problem. For example, let us observe how a new unlabeled point $\boldsymbol{x}_{n+j}$ would be "injected" in this case to enlarge (13). First, in the E step, the latent labels T_u are predicted using the distribution $P(T_u|\boldsymbol{X}_u, \boldsymbol{\theta}, \boldsymbol{\mu})$, where $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ represent the current MAP models *before* having seen the new point.[19] Here, $\boldsymbol{x}_{n+j}$ has been injected into $\boldsymbol{X}_u$ and t_{n+j} into T_u. Then, in the M step, $\boldsymbol{\mu}$ is updated to $\tilde{\boldsymbol{\mu}}$ so as to maximize $P(\boldsymbol{X}|\boldsymbol{\mu})P(\boldsymbol{\mu})$, this is standard MAP on the model class $\{P(\boldsymbol{x}|\boldsymbol{\mu})\}$. Finally, we update $\boldsymbol{\theta}$ so as to maximize $E_{T_u \sim Q}[\log P(T_u|\boldsymbol{X}_u, \boldsymbol{\theta})P(T_l|\boldsymbol{X}_l, \boldsymbol{\theta})P(\boldsymbol{\theta}|\tilde{\boldsymbol{\mu}})]$. This involves fitting a diagnostic model by MAP to a dataset which is partly uncertain (the uncertainty is over T_u and is represented by the data distribution Q, see subsection 2.2.1), and there exist standard methods for this task, for example the *iteratively reweighted least squares (IRLS)* technique (see e.g. [56]).

It is possible to extend this method, using variational techniques to approximate posterior beliefs of $\boldsymbol{\mu}$. This scheme is discussed in [81]. Exploring it on concrete data and model families, such as Gaussian process classification, remains a topic for future research.

The reader might have noticed that we have been quite cautious in the formulation of this algorithmic scheme. Using EM on diagnostic models is quite "slippery", and it is not at all certain under what conditions on the input-dependent regularization, represented by the conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$, it can be successful. We have argued that it cannot work if $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ are a-priori independent. On the other hand, it works extremely well on certain special tasks where strong structural prior knowledge about the relationship between input distribution and discriminant function is available. Such a case is the *Co-Training* method which has been proposed in [12] to attack the problem of Web page classification (see subsection 3.3). The basic Co-Training algorithm can be seen as a robust variant of EM on diagnostic models if the assumptions proposed in [12] are encoded in conditional priors $P(\boldsymbol{\theta}|\boldsymbol{\mu})$. A detailed derivation and discussion of this view on Co-Training can be found in [81]. Using this view, the model algorithm introduced in this section can be seen as a generalization of Co-Training.

[19] If we talk about "predicting" latent variables using a distribution R, we really mean that we update the data distribution Q on these variables to coincide with R (see subsection 2.2.1). In the M step, we compute the EM criterion by taking the expectation of the complete log joint distribution w.r.t. Q. In simple (but important) cases, the resulting EM criterion looks like the log joint distribution over a complete dataset (i.e. containing also values for the latent variables), in which the values of the latent variables are given by their *expectations* under Q. The terminology of "predicting" latent variables should be understood in this sense.

We finally note that further design decisions come up when using unlabeled data in this context. For example, one can split the unlabeled dataset D_u, using one part for "injection" in a robust EM variant, as discussed in this subsection, while the other part is used to select a sensible prior $P(\boldsymbol{\mu})$ over the model class for the input distribution.

3 Literature review

Recently literature adressing the labeled-unlabeled problem comes from a certain variety of fields. We have attempted the "unsupervised task" to classify these approaches into a number of clusters[20]. However, the aim is purely organisatorial, since this is neither the right time nor are we in a position to formulate paradigms for the labeled-unlabeled problem. Class membership is often "fuzzy", as will be pointed out for any single reference. In this section, we discuss solely work which adress the labeled-unlabeled problem directly. Some work which is *related* to the problem, is discussed in section 4.

3.1 Theoretical analyses and early work

The idea of using EM on joint models to train on labeled and unlabeled data (see subsection 2.2) is almost as old as the seminal paper [26] on (general) EM. Titterington et al ([90], section 5.7) review early theoretical work on the problem of discriminant analysis in the presence of additional unlabeled data. Most of the authors assume 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 sampling paradigm in which the parameters of the class distributions are estimated by maximum likelihood, as discussed in subsection 1.3.1. If the two Gaussians are somewhat well-separated, the asymptotic gain of using unlabeled samples is very significant. Also, empirical studies on finite samples are promising. For details, see [67],[32],[33]. McLachlan [57] 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 $|D_u| = m \to \infty$. While these papers give some positive motivation towards the feasibility of the labeled-unlabeled problem, they start off from somewhat unrealistic assumptions. The prior assumption that the

[20] Having available limited prior knowledge and no labeled examples at all!

class distributions are Gaussian with equal covariances is much too strong to be sensibly applied (as prior knowledge) to nontrivial real-world tasks.

All the papers discussed so far in this subsection focus on generative methods, i.e. assume parametric forms of the class-conditional distributions. Anderson [3] suggests a modification of logistic regression, one of the most popular diagnostic methods. Logistic regression (see e.g. [70], also end of subsection 2.1) models the *logits* $\log(P(t|\boldsymbol{x})/P(1|\boldsymbol{x}))$, $t = 2, \dots, c = |T|$ as *linear* functions of $\boldsymbol{x}$. Here, $\boldsymbol{x}$ is augmented by a dummy attribute (dimension) whose value is constantly 1. If the true underlying populations are all normal and share the same covariance, the logits are indeed such linear functions. Supervised ML logistic regression proceeds by choosing the linear function which maximizes the (conditional) likelihood of the data D_l. A Bayesian approach would place a prior on the linear function and compute the posterior distribution. A MAP approximation to Bayesian Gaussian process classification (e.g. [99]), also called *generalized penalized maximum likelihood*, can be seen as logistic regression in a feature space (see e.g. [37]). In such purely diagnostic settings, unlabeled data cannot help narrowing our belief in the latent function, see subsection 1.3.2. Anderson [3] circumvents this problem by choosing a parameterization which is *mixed* from the diagnostic logistic regression setting and the sampling paradigm situation. Let $c = 2$. The only assumption is that $\log(P(\boldsymbol{x}|1)/P(\boldsymbol{x}|2)) = \boldsymbol{\beta}^T\boldsymbol{x}$. Then, $P(\boldsymbol{x}|1) = \exp(\boldsymbol{\beta}^T\boldsymbol{x})P(\boldsymbol{x}|2)$ and $P(\boldsymbol{x}) = (\pi_1 \exp(\boldsymbol{\beta}^T\boldsymbol{x}) + 1 - \pi_1)P(\boldsymbol{x}|2)$, where $\pi_1 = P\{t = 1\}$. He now chooses the parameters $\boldsymbol{\beta}$, π_1 and $P(\boldsymbol{x}|2)$ to maximize the likelihood of both D_l and the unlabeled data D_u, subject to the constraints that $P(\boldsymbol{x}|1)$ and $P(\boldsymbol{x}|2)$ are distributions (i.e. sum to 1). For finite X, this problem can be transformed into an unconstrained optimization w.r.t. the parameters $\boldsymbol{\beta}$, π_1, using Lagrange multipliers. For a continuous input variable $\boldsymbol{x}$, Anderson advocates using the form of $P(\boldsymbol{x}|2)$ derived for the "finite X" case, although this is not a smooth function. While this algorithm is interesting, it is rather restricted by the assumption of a linear logit. It is not clear how to generalize it to the much more powerful case of logistic regression in a feature space. Furthermore, for small samples and continuous $\boldsymbol{x}$, the form of $P(\boldsymbol{x}|2)$ obtained by non-penalized ML is inadequate. The idea of "mixing" diagnostic and generative techniques is probably represented in a more satisfying way in the scheme suggested in subsection 2.3 as an extension of the algorithm of Miller and Uyar [59].

Murray and Titterington (see [90], example 4.3.11) suggest an ad hoc procedure. Namely, they use the labeled data available for each class to obtain kernel-based estimates of the class-conditional densities $P(\boldsymbol{x}|t)$. Then, they fix these estimates and use EM to maximize the likelihood of both D_l and D_u w.r.t. the mixing coefficients (i.e. the parameters representing $\pi_t = P(t)$) only.[21] This procedure is robust, however does not make a lot of use of the unlabeled data. If D_l is small, the kernel-based estimates of the $P(\boldsymbol{x}|t)$ will be poor, and even if D_u can

[21] 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 [21].

be used to obtain better values for the mixing coefficients, this is not likely to rescue the final discrimination. The method is valuable if D_l is rather large, but the proportions of sample points from the different classes in D_l do not reflect the true mixing coefficients.

Shahshahani and Landgrebe [83] 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. [65], [102]). They do not define model classes and seem to confuse asymptotic and finite-sample terms. While it is true that there are strong consistency arguments for estimators like maximum-likelihood which hold *asymptotically*, independently of the model class, characteristics of the models are crucial in the finite-sample case. Even worse, if we want to talk about the labeled-unlabeled problem, the labeled dataset D_l is *small*. The assumption that one can find *unbiased estimators* in this case is very unrealistic. Clearly one can come up with methods that reduce the *variance* of an estimator by employing unlabeled data, but there is no reason to believe that such modifications will work without introducing new *bias*. It is quite obvious that information from different sources (here: D_l and D_u) about the latent model parameter adds up. It is less obvious how to construct a model family and a learning algorithm such that the information of D_u about $\boldsymbol{\theta}$, *conditioned* on the choice of the model family, is non-zero, and the algorithm makes significant use of this information *without introducing new bias*. For example (as also pointed out in [102]), in the standard generative model for diagnostic classification, the information of unlabeled data about the latent discriminant is zero, whether measured asymptotically or on a finite sample (see subsection 1.3.2). The parametric approach adopted in [83] is the straightforward EM algorithm (see subsection 2.2) suggested e.g. in [58].

Another analysis of the problem which also employs Fisher information techniques, is given in [102]. In this paper, the models and the data generation process are carefully defined to avoid confusions, such as arise from [83]. For diagnostic methods, the authors show that under the standard generative model, unlabeled data cannot help. We have already commented on this point in subsection 1.3.2. For generative methods, the conclusion is that unlabeled data always helps. This is true under the assumptions made in the paper. However, the analysis draws on asymptotic concepts. 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. The authors present some interesting empirical evidence concerning the performance of transduction algorithms (discussed below in subsection 3.7) on a text categorization task. These results indicate that transduction algorithms might suffer from instability (or robustness) problems similar to those mentioned in the context of the EM algorithm (see subsection 2.2). The paper also deals with active learning scenarios, which are out of the scope of our review.

In [16], the labeled-unlabeled problem is analyzed starting from a very strong assumption, namely that we are capable of identifying all class distributions $P(\boldsymbol{x}|t)$ exactly, using unlabeled data D_u only. Even if $m = |D_u| = \infty$, it is not clear how this should be achieved by an (unsupervised) learning procedure. First of all, the authors mention that (trivially) classification based on unlabeled data only is not possible. Even if we have identified all the class regions, we cannot deduce "which is which" without any label information. They then continue to show that *if* we have identified all the class regions, the optimal error probability for n labeled samples converges towards the Bayes error exponentially fast in n. Although the authors propose to address the more realistic case of finite m in a subsequent paper, we are not aware of such work in the moment. Given as such, the paper motivates the approach mentioned in subsection 2.1, namely to use a strong unsupervised technique to identify all class regions (or all connected parts of class regions), based on D_u only. The authors show that, once this task is achieved, labeling these parts is fairly easy (in terms of the size of D_l required). However, their assumptions are too strong to be met in practice. It would be more interesting to investigate the value of labeled samples under the much weaker assumption that only the marginal $P(\boldsymbol{x})$ is identifiable.

3.2 Expectation-maximization on a joint density model

We have decided to classify "solutions" of this kind as baseline methods only (see subsections 2.2,2.3), partly because filling in missing information by EM is a standard technique, partly because running EM on most labeled-unlabeled problem instances does not work well enough empirically to be called a solution.[22]

The work of Miller and Uyar [59] has already been discussed in subsection 2.3. The authors suggest, as a variant, to treat the class label t as a second latent variable alongside the separator k (the data distribution would be definitive (or clamped) on the points from D_l, see subsection 2.2.1), and to apply EM to both. An effect of this variation is to make the lower bound of the EM criterion on the marginal log likelihood worse, however both variants come with the same guarantee of convergence into a local maximum of the marginal likelihood. On the architecture proposed by Miller and Uyar, we would expect them to exhibit comparable performance, indeed this is what the authors find in their experiments. The latent label variant might be advantageous when extending this architecture, as suggested in subsection 2.3.

Nigam et al [65] present a case study which addresses the question whether unlabeled data can help to improve a Naive Bayes text classifier in the case of small D_l.[23] They use EM on a joint model, as discussed in subsection 2.2, as well as two simple extensions. First, they suggest weighting the two sums in

[22] As mentioned at the beginning of section 2, beating a purely supervised technique based on small D_l is not sufficient to conclude that a method succeeds in solving the problem.

[23] The Naive Bayes assumption proposes models for text pages under which all words on a page are conditionally independent, given the class label of the page.

the joint log likelihood (6), corresponding to D_l and D_u respectively, unequally. While it is reasonable to treat points from D_l and D_u differently, simply because otherwise we run the risk that the much more informative labeled points are dominated by the sheer amount of available unlabeled points, and while introducing a weighting factor in the joint log likelihood is straightforward, this modification is not a probabilistic one (the modified criterion is not a log likelihood function anymore), and the weighting factor has to be chosen using heuristics (using cross-validation for this purpose is unwise since D_l is very small). The other extension, namely modeling each class by more than one center, can be seen as a special case of the EM technique discussed in subsection 2.3. We have already criticized the use of standard EM to attack the labeled-unlabeled problem in subsection 2.2. This critique applies nicely to the setting discussed in [65], since the Naive Bayes assumption leads to extremely poor models for the class-conditional distributions. However, the paper is not intended as to provide genuinely new solutions, and given that, its merits are considerable in that it clarifies joint model EM techniques, notes problems related with these techniques, provides an extensive case study and contains a very detailed section on related work. A later paper [64] extends the case study, including more robust EM variants (see subsection 3.3).

In [55], the authors try to combine an EM algorithm on a joint probability model (see [65] and subsection 2.2) with an *active learning* strategy, namely the *query-by-committee (QBC)* algorithm ([82], see also [31] and subsection 4.1), to attack an instance of the labeled-unlabeled problem in text classification. The idea is to overcome stability problems of standard EM by injecting unlabeled points one at a time. Given a large pool of unlabeled data, the authors initialize EM by training on the labeled data only. They then use a criterion derived from QBC to select a few "most informative" points among the unlabeled ones, transfer them (together with their latent labels) into the EM dataset and rerun EM. This is iterated until some convergence criterion is met. While the combination of EM and active learning is a very original idea, we found the particular realization of this idea presented in [55] somewhat distorted by the use of a host of heuristic intermediates between QBC and EM. For example, the criterion used to value the informativeness of an unlabeled text page is obtained by multiplying the QBC criterion with a heuristic measure for the density of $P(\boldsymbol{x})$ around a document. And the QBC criterion is somewhat distorted, in the following sense: the basic idea of QBC is to select query points $\boldsymbol{x}$ so as to maximally reduce *variance* of the discriminant ensemble represented by the posterior distribution given by the data observed so far. On a fixed $\boldsymbol{x}$, this variance can be measured by sampling a fixed number of discriminants from the posterior and evaluating them on $\boldsymbol{x}$. McCallum and Nigam [55] also produce such a sample, but then they run EM to convergence starting from each parameter vector in the sample, and replace the initial by the final, converged vector. It is not clear to us at all if the committee based on this set of parameter vectors fulfils requirements to estimate variance, as is done in QBC. In their experiments, the authors are surprised to find that this "distorted" version of QBC does not improve upon the much more efficient undistorted variant followed by a single EM run. The authors find that the proposed method outperforms standard EM on the same joint model family (see [65]). It would be interesting to compare the active selection method with other robust variants of EM (e.g. see end of subsection 2.2.1, also "self-training" in [64], see subsection 3.3).

3.3 Co-Training algorithms

Co-Training is a learning paradigm which has recently been proposed in [12] to address problems where strong structural prior knowledge is available. It has been mentioned in subsection 2.4 that Co-Training can be seen as Bayesian inference, and the basic Co-Training algorithm as a robust variant of EM to compute an MAP approximation to Bayesian inference, if the assumption of the "compatibility" of the target concept with the input distribution is encoded via conditional priors to attain an input-dependent regularization (see subsection 1.3.3). We refer to [81] for details.

Co-Training is a simple (yet very effective) idea, therefore it does not come as a surprise that related ideas have been used in earlier work on unsupervised learning. We begin by reviewing some of this work. Becker and Hinton [6] propose the *IMAX* strategy to learn *coherence structure* in data. Quoting [7], the approach is "to maximize some measure of agreement between the outputs of two groups of units which receive inputs physically separated in space, time or modality (...). This forces the units to extract features which are coherent across the different input sources". The latter claim seems reasonable, given that the model families are carefully regularized, although we are not aware of theoretical work backing it. For simplicity, let us focus on the example of detecting shift in random dot stereograms. Here, $\boldsymbol{x} \in G_2^{2\times d}$, $G_2 = \{0, 1\}$. If $s \in \mathbb{Z}$ denotes a small, unknown offset, a point $\boldsymbol{x}$ is sampled as follows: the first row is drawn from the product of d Bernoulli variables with $p = 1/2$. The second row is the same as the first, but shifted by s positions.[24] In a strong sense, the *only* coherence between examples sharing the same s *is* exactly the amount of this shift. Now imagine two model classes $\{P_i(t|\boldsymbol{x}, \boldsymbol{\theta}_i)\}$, $i = 1, 2$. t ranges over a finite set whose size is chosen a-priori, although it is possible to learn this size from data using a sort of second-level inference (we skip this for simplicity). The idea is that models from class i only get to see a particular part of each point $\boldsymbol{x}$, defined by a window $W_i = (l_i, r_i)$, $1 \le l_i < r_i \le d$. To be specific, models from class i are fed by $[\boldsymbol{x}]_i = (x_{jk})_{j=1\dots 2, k=l_i\dots r_i}$, where $\boldsymbol{x} = (x_{jk})_{j,k}$. The two windows are non-overlapping and usually do not cover the whole range $1\dots d$. For each example drawn as detailed above, the particular amount of shift is coherent between these two views on the point. Both models classes are appropriately regularized, e.g. using Occam priors. The goal is to learn models $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ which identify the shift present in each particular pattern, i.e. group examples into the same

[24] If s is generally small compared to d, it does not really matter what we do at the margins, but let us say for simplicity that the first row is *rotated* by s positions to form the second, i.e. the free space at the one end is filled with the elements pushed out at the other end.

cluster t if they exhibit the same amount of shift. Becker and Hinton showed that this task can be solved in an unsupervised manner, by maximizing the sample mutual information between the outputs of two units, one from each of the model classes. To be specific, for $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ define the random variables t_1, t_2 to have the joint distribution $P(t_1, t_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = E_{\mathbf{x}}[P_1(t_1|\boldsymbol{x}, \boldsymbol{\theta}_1)P_2(t_2|\boldsymbol{x}, \boldsymbol{\theta}_2)]$, where the expectation over $\boldsymbol{x}$ is w.r.t. the empirical distribution given by the dataset. The marginals are $P(t_i|\boldsymbol{\theta}_i) = E_{\mathbf{x}}[P_i(t_i|\boldsymbol{x}, \boldsymbol{\theta}_i)]$. The sample mutual information, or the "IMAX criterion", is then defined as

$$I(t_1, t_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = E_{t_1,t_2}\left[\log \frac{P(t_1, t_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{P(t_1|\boldsymbol{\theta}_1)P(t_2|\boldsymbol{\theta}_2)}\right], \tag{14}$$

where the expectation is over $P(t_1, t_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. Note that this criterion is minimal if t_1, t_2 are independent, maximal if they are deterministically related, e.g. identical. The IMAX strategy is to maximize the criterion, given an appropriate regularization on the two model classes. If regularization is done using Occam priors $P(\boldsymbol{\theta}_i)$, one could for example maximize the sum of the IMAX criterion and the both log priors. The work of de Sa [24] is related in that we train two families of models (logistic regression models in her work), each being fed by a different view on examples. These views are different in modality, an example would be sound and lip images in order to decode speech. The models are seen as hard discriminants, and the system is trained to minimize the fraction of training examples on which the two units disagree. This is suboptimal for logistic regression models, since the confidence information given by these "soft" estimators is neglected.

We think that a thorough theoretical analysis of IMAX and related schemes is very difficult. However, we can give some intuitive ideas why they might work. Learning regularities (like class identity, coherence, ...) from a restricted view on examples is conceptually easier than from a complete representation. This is partly due to limited data, with which spaces of lower dimension can be covered easier, partly due to our (present) inability to construct good model families for high-dimensional data[25], partly due simply to limited computing resources. Therefore, if a certain coherence is exhibited (almost) as clearly in a restricted view as it is in a complete representation, and using this coherence pays off strongly in an attempt to efficiently encode the data (see section 1), then it is much more likely that a common learning scheme discovers this coherence on the restricted view, even if trained in an unsupervised manner. Now, by linking the units operating on different views, e.g. in the IMAX fashion, the discovered information is instantly passed to the "partner model", in much the same way as a teacher passes information to a student in a supervised learning scheme. This is a particularly nice example of Occam's razor to drive unsupervised learning. As soon as one of the units discovers a means to exploit (part of) the coherence in the data, this is, simply by its potential to compress the data drastically,

[25] We (at least, *we*!) do not even have a good understanding of how simple families of distributions, such as rather small mixtures of Gaussians, *behave* in high-dimensional spaces, in the sense that we do not clearly understand how they combine their volumes to fit data.

taken for "reliable truth" which can be passed to partners in a teacher-student manner. Note that if the discovered "coherence" is spurious, i.e. the "student" cannot detect it using its own view, its inability, within its own "regularized simplicity", to synchronize with the teacher can in turn influence the teacher to give up on the idea or to assign a lower importance to it.

The work discussed so far in this subsection could be used to attack the labeled-unlabeled problem as follows: suppose we have some prior knowledge about a particular coherence which should hold between examples of the same class, but (at least to a certain degree) *not* across class boundaries. Such coherences are of course strongly dependent on the representation (the "features") of examples. A simple means to find such coherence is to look for different views (different in modality) on the examples, e.g. coming from physically different information sources. Another idea is to identify groups of transformations acting on the representation $\boldsymbol{x}$ of examples which (most probably) are invariant w.r.t. class identity (this idea is very important in fields dealing with object recognition in images, see subsection 4.3). Given such coherence information, we can try to find different views on our examples which fulfil the following criteria: they should be "as different as possible", e.g. coming from different physical sources. Ideally, they should be conditionally independent, given the class identity of the example, i.e. the only information shared between them should be about the class label. They should also be aimed towards exhibiting the particular coherence clearly *in isolation.* Ideally, although each of the views only describes part of the information contained in an example, one should be able to learn the particular coherence from only this view on examples "essentially as easy as" from complete representations. For example, the shift in random dot stereograms, as discussed above, is represented to the same degree in the full matrix $\boldsymbol{x}$ and in a windowed part of it. Given all this prior work, we can run IMAX or the algorithm of de Sa, using model classes and regularizations of our choice, to learn a soft partitioning of the example space in an unsupervised manner, using D_u as training data. Afterwards, we use the labeled data D_l to assign clusters to classes. This is related to the general scheme discussed in subsection 2.1. The relations to Co-Training are discussed further below in this subsection.

The *Co-Training* paradigm was introduced by Blum and Mitchell [12], see also [62],[81]. The idea is to exploit a particularly strong kind of coherence (in the sense discussed above), namely the notion of *compatibility* between different views on an example $\boldsymbol{x}$. We write $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \in X = X^{(1)} \times X^{(2)}$. Blum and Mitchell are interested in classification of Web pages, they suggest to describe a Web page from two different views: a representation of the words *on* the page, and a representation of the words in all hyperlinks *pointing to* the page. Note that if a classification system makes sense, within-class coherence between examples should be learnable from each of these views *in isolation.* A hypothesis $\boldsymbol{\theta}$ on X is *compatible* with the input distribution $P(\boldsymbol{x})$ if there are hypotheses $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ on $X^{(1)}, X^{(2)}$ respectively, such that for any $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$ from the *support* of $P(\boldsymbol{x})$ (i.e. $P(\boldsymbol{x}) > 0$) we have that $\boldsymbol{\theta}, \boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ predict the *same* class label on $\boldsymbol{x}, \boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ respectively. The compatibility assumption of

Co-Training restricts the latent hypothesis $\boldsymbol{\theta}$ to be compatible with the (latent) input distribution $P(\boldsymbol{x})$. Given this assumption, Blum and Mitchell suggest a simple algorithm to learn the classification using a small labeled dataset D_l together with a large unlabeled one, D_u. This works by updating a set of labeled data, D_w, and hypotheses $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ on $X^{(1)}, X^{(2)}$ respectively. $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ are fitted to D_w, and are ideally error-free on this set. Initially, $D_w = D_l$. One now injects new points from D_u into D_w sequentially, by labeling them using one of the $\boldsymbol{\theta}^{(j)}$ and retraining the other on the augmented D_w. The roles of "teacher" and "student" are alternated between $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ after each injection. This algorithm can easily be understood as an instance of the robust EM scheme on diagnostic models, described in subsection 2.4. The notion of compatibility between the views is encoded in a conditional prior $P(\boldsymbol{\theta}|\boldsymbol{\mu})$. Instead of a model for the input distribution, we estimate its support only, which we refer to as $\boldsymbol{\mu}$ here. We then set $P(\boldsymbol{\theta}|\boldsymbol{\mu}) = 0$ for all $\boldsymbol{\theta}$ which are *not* compatible with the support $\boldsymbol{\mu}$. A detailed derivation of Co-Training as a variant of diagnostic EM can be found in [81]. Blum and Mitchell also give some interesting theoretical analysis of Co-Training. Unfortunately, they employ the very strong assumption that the views on $\boldsymbol{x}$ are conditionally independent, given the class label. It would be very interesting to get theoretical insight into cases with weaker assumptions.

Both IMAX-like schemes and Co-Training employ a feature split and coherence between the different views to learn from unlabeled data D_u. The coherence is a kind of *redundancy* in unlabeled examples, and this redundancy is useful information towards a meaningful grouping of the examples. However, there are important differences. IMAX and related schemes are purely unsupervised, while Co-Training leads to supervised methods. IMAX is designed to learn a grouping which corresponds to the coherence which is expected, by prior knowledge, to hold between the different views on examples. If t denotes the corresponding grouping variable, IMAX requires the views on $\boldsymbol{x}$ to be selected such that they are conditionally independent, or at least only weakly conditionally dependent, given t. If this does not hold, IMAX will probably learn a different grouping, or fail to learn a meaningful grouping at all. For example, if the two windows in the random dot stereograms setting (discussed above) overlap, IMAX might fail because of low-order conditional dependencies (given t) between the views[26]. In Co-Training, we use a feature split which is chosen such that coherence between the views is compatible with class identity. However, this split is used merely as a sort of "information bridge" between unlabeled data and our belief in the latent hypothesis $\boldsymbol{\theta}$, *not* as essential characteristic of the grouping induced by $\boldsymbol{\theta}$. While unlabeled data can be used most effectively if the views on $\boldsymbol{x}$ are conditionally independent or only weakly conditionally dependent, given t, we would expect that D_u can boost the performance, compared to classification based on D_l alone, given somewhat weaker assumptions.

Collins and Singer [20] apply the Co-Training paradigm to the problem of *named entity classification*. Here, one is interested in classifying entities which are

[26] Thanks to Chris Williams for pointing this out.

uniquely represented by names. The classification system would be, for example, persons, cities and companies. Although the correspondence is one-to-one, it cannot be represented by a small system of simple rules. One therefore augments the description of the entity by features which can be extracted automatically from samples in which the entities occur. An example would be to extract the context of the name in text pages in which the names occur. Another idea is to look at the concrete spelling of the name. It is clear that one can find different views on the named entities satisfying the Co-Training requirements. The most interesting part of the paper is the development of *co-boosting*, namely an extension of the very powerful *AdaBoost* algorithm (see [30], [72]) for supervised classification to attack the labeled-unlabeled problem. This extension is surprisingly simple, yet very elegant, and the algorithm could prove very competitive among existing labeled-unlabeled algorithms. However, as the authors report, it suffers in principle from the same robustness problems than EM (see subsection 2.2) or existing transduction algorithms (see subsection 3.7). The algorithm was in some cases fooled by apparent simplicity in the structure of the unlabeled data and exhibited large bias, although the information in the labeled data could have been used to detect the failure.

Nigam and Ghani [64] present a case study comparing standard EM (see subsection 2.2, also [65]), basic Co-Training (as suggested in [12]) and some robust EM variants (see end of subsection 2.2.1), partly on data from natural sources, partly on artificially created datasets. The task is again text classification. This paper is very valuable as comparative study and contains some interesting ideas of how to increase the robustness of standard EM or how to combine the "best of both worlds" (Co-Training and EM employing class-conditional models). However, the authors confuse some points. First of all, they do not realize that basic Co-Training is a form of (diagnostic) EM, see subsection 2.4. Therefore, if they compare Co-Training against versions of generative EM (i.e. EM employing class-conditional models) which try to make use of the feature split prior knowledge[27], this has more the notion of comparing the diagnostic versus the sampling paradigm for the labeled-unlabeled problem. Second, they criticize "EM — the algorithm with a strong probabilistic foundation" ([64], section 6.2), because it performs worst in their experiments. They do not realize that their "best performer", an algorithm they call "self-training" (see end of subsection 2.2.1), is just *another version of EM*, based on the *same* strong probabilistic foundation than the worst-performing standard batch EM. This foundation is detailed in subsection 2.2.1.

Goldman and Zhou [36] propose an algorithm to address the labeled-unlabeled problem which is related to Co-Training. It does not employ a feature split (or different views on the examples), but incorporates two very different model classes. At any time, the current discriminant is represented by two models, one from each class. The models are initially chosen by training on D_l only. The algorithm then works by, in turn, identifying points among the remaining ones

[27] The authors call this variant "co-EM".

in D_u on which one of the models predicts very confidently, and adding them to D_l together with the predicted pseudo-label. On the positive side, the paper addresses some important robustness issues, such as the danger of accumulating "classification noise" in D_l by incorporating a certain number of *incorrectly* labeled points from D_u. The authors propose safeguards against these issues which seem rather conservative to us. Our problem with this paper is that the authors apply a host of tests from classical statistics, sometimes testing hypotheses against each other which are conditioned on completely different events (such as different partitions of the input space — their model classes are classification trees). We do not think that the assumptions on which these classical tests are based really hold in all the different situations the tests are applied here. A concrete weakness seems to be the frequent use of 10-fold cross-validation using the labeled data D_l. If $n = |D_l|$ is rather small, cross-validation exhibits high variance and is not very useful for model selection, especially if it is used very frequently to address all sorts of different modeling questions. In short, the idea to employ very different model classes instead of a feature split, if this is supported by prior knowledge, is interesting and should be investigated[28], but the particular algorithm suggested in [36] should probably be shaved a bit under Occam's razor. Note that Seeger [81] proposes a generalization of Co-Training in which the issues that Goldman and Zhou attack heuristically, are dealt with in a principled Bayesian way. We have not checked whether this approach is feasible for the architecture of Goldman and Zhou.

3.4 Adaptive regularization criteria

The basic idea behind adaptive regularization criteria is that criteria to be minimized in supervised settings, like generalization error or expected loss, involve an expectation over the (unknown) input distribution $P(\boldsymbol{x})$. Simply stated, making mistakes in regions where $P(\boldsymbol{x})$ is large, hurts more (in terms of these criteria) than mistakes in regions of low density of $P(\boldsymbol{x})$. The *overfitting problem* is likely to arise if a complex model is fitted to sparse data. When we "fit a complex model to data", what we really do is we choose, among all the functions (or relations) this model is able to represent, *one* which is compatible with the data. Since the model is complex, it usually can represent a large number of functions which are all compatible with the data, but show very different behaviour away from the data. Our criterion gives us no further rules or constraints of how to choose among these, and a random choice is likely to generalize badly to unseen data. *Regularization* based on Occam's razor gives us an additional rule: prefer simple over complex functions. The concrete meanings of "simple" and "complex" of course depend on the task and on available prior knowledge. The point to be made in this subsection is, however, that the Occam assumption of simplicity should really only be enforced in regions where input points $\boldsymbol{x}$ are

[28] It is related to "learning how to learn" or multitask learning, see subsection 4.2.

likely to be found.[29] In other words, regularization backed by Occam's razor should be dependent on the input distribution $P(\boldsymbol{x})$, see subsection 1.3.3.

Schuurmans [76] captures this kind of input dependence by defining a *natural metric* between hypotheses. Given a symmetric loss function l on $T \times T$ (recall that T is the set of possible values for the target t), the *expected loss* of a hypothesis $h : X \to T$ is $d(h, P(t|\boldsymbol{x})) = E_{\mathbf{x}}[E_{t\sim P(t|\mathbf{x})}[l(h(\boldsymbol{x}), t)]]$. This suggests the definition of a natural metric between hypothesis g, h by $d(g, h) = E_{\mathbf{x}}[l(g(\boldsymbol{x}), h(\boldsymbol{x}))]$. Now, if we are given an hypothesis space $\mathcal{H}$ and define $\mathcal{H}_{ext} = \mathcal{H} \cup \{P(t|\boldsymbol{x})\}$, it is easy to show that if l is a *pseudometric* on $T \times T$ (as is the case for common loss functions like squared error in regression estimation or zero-one loss in classification), d, as defined above, is a pseudometric on $\mathcal{H}_{ext}$, in particular d fulfils the *triangle inequality.* Now, suppose we construct a hierarchy $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \dots$ of increasing complexity, and within each $\mathcal{H}_i$ we select $h_i \in \mathcal{H}_i$ which best fits the data D_l. Then, by the triangle inequality, we must have

$$d(h_i, h_j) \leq d(h_i, P(t|\boldsymbol{x})) + d(h_j, P(t|\boldsymbol{x})) \qquad (15)$$

for any $i < j$. Schuurmans argues that one can estimate the left-hand side fairly accurately, given a large *unlabeled* sample D_u, while the expected losses on the right-hand side are estimated using D_l. Overfitting occurs if these *empirical losses* are grossly different from the true expected losses, and in this case the former are almost always strong *underestimates* of the latter. Therefore, once inequality (15) is violated after plugging the *estimates* in place of the true, unknown quantities, we can conclude that overfitting has occured. Schuurmans proposes a model selection procedure in which we select hypotheses $h_j \in \mathcal{H}_j,\ j = 1, 2, \dots,$ until the estimated version of (15) is violated for some $i < j$. The procedure then outputs h_{j-1}. Empirical results are promising in the case of regression estimation with polynomials. However, the technique might still exhibit overfitting, simply because the triangle inequality (15) is usually far from tight. An extension of this strategy, called ADJ, attempts a first-order bias correction between the pseudometric d and its estimated version, say $\hat{d}$. In a later paper [77], Schuurmans and Southey get rid of the a-priori hierarchy and focus on criteria which are additive or multiplicative combinations of the empirical loss $\hat{d}(h, P(t|\boldsymbol{x}))$ and a penalty. They then propose penalties based on the idea that overfitting of h can sometimes be detected by comparing, for some fixed *origin function* ϕ, the distances $d(h, \phi)$ (which can be estimated reliably using

[29] It is not our plan to go into discussions about foundations of Occam's razor here. Some people think Occam's razor is backed by evolutionary theories in which complex structures evolve from very simple initial conditions by accumulation of many small *random* changes together with selection processes. In this context, simple solutions to a problem are more likely to be found than complicated ones, given that the sequence of selection processes which have been active during the evolution of the solution is not too unusual. Evolution of such solutions is, however, *conditioned* on the particular situation surrounding the problem, since this situation creates the selection processes which give evolution a non-random direction. For example, human cells work very effectively and in many aspects surprisingly simple, but *only* given that the temperature lies within a narrow range. In conclusion, "simplicity" has to be graded *conditioned* on the situation which the particular task is expected to be in.

D_u) and $\hat{d}(h, \phi)$ (estimated based on the input points of D_l only). It is important that ϕ is chosen a-priori, without having seen any data. The penalties can be motivated nicely if $\phi = P(t|\boldsymbol{x})$, while such a choice is of course unrealistic. The motivation gets somewhat weaker in the case where ϕ is chosen arbitrarily. Empirical results on a polynomial regression estimation task show that the method is very competitive, while results on classification are less convincing. It is interesting that, empirically for regression estimation, a quite aggressive *multiplicative* penalty outperforms an additive penalty based on the same idea, since most regularization strategies currently in use (including Bayesian *MAP estimation*) employ additive penalties. We further note that the additive criterion in [77], as applied to regression estimation with squared-error loss, has already been suggested in [17]. However, Cataltepe et al [17] do not give a very convincing theoretical motivation.

To conclude, adaptive regularization criteria are based on the notion of input-dependent regularization (see subsection 1.3.3 and [81]). While the criteria suggested in [76], [77] are reported to work well on regression estimation tasks, the reported results for classification are less promising. In the latter, the information each labels contains about the latent function is less directly accessible than in regression estimation. It is also not clear how to inject available prior knowledge into the procedures suggested in [77], but we are very interested in following up this very recent line of research.

3.5 The Fisher kernel

The *Fisher kernel*, as proposed in [46], is the first general and principled attempt to exploit information from a generative model fitted to the input distribution $P(\boldsymbol{x})$ in one of the most powerful currently available classes of *discriminative classifiers*, namely *kernel methods* such as *Gaussian processes* (e.g. [100],[97],[54]) or *Support Vector machines* (e.g. [96],[14]). In a nutshell, kernel methods are diagnostic schemes (see subsection 1.3.2) in which the prior distribution over the latent function[30] is a *Gaussian process*, specified by a *positive definite covariance kernel* (e.g. [40]). The covariance kernel induces a "natural" distance in a feature space, and the Fisher kernel attempts to adapt this distance in a highly genuine and interesting way to information about the distribution of input points, drawn from a model fitted to $P(\boldsymbol{x})$. One of the main difficulties in constructing adaptive kernels is that they need to fulfil the requirement of positive definiteness, i.e. that they can be seen as *inner products* in some linear space (e.g. [40]). Finding decent kernels on "unusual" (but very important) input spaces X (such as spaces of variable-length sequences or of discrete structures) is challenging, and not many general solutions are available (e.g. [40]). But even in the case $X = \mathbb{R}^d$ it seems questionable to employ families of standard kernels on highly specific tasks (as is done usually) where labeled data D_l is sparse,

[30] In the two-class case (i.e. $|T| = 2$), the latent function represents the log-ratio between the two classes at each point $\boldsymbol{x}$, also called the *logit*.

simply because the kernel encodes our prior knowledge about the task, and the standard kernels available offer few, rather vague possibilities for doing so. In a sense, when running kernel methods together with standard kernels, we ignore the (probabilistic) geometry of X in the same way as if we employ standard distances like squared-error or impose uncorrelated Gaussian noise in situations where we should know better (see e.g. [87]).

The naive Fisher kernel is

$$K(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) = \nabla_\theta \log P(\boldsymbol{x}^{(1)}|\boldsymbol{\theta})^T \mathcal{F}^{-1} \nabla_\theta \log P(\boldsymbol{x}^{(2)}|\boldsymbol{\theta}), \tag{16}$$

where $\{P(\boldsymbol{x}|\boldsymbol{\theta})\}$ is a model family used to fit $P(\boldsymbol{x})$. If $\hat{\boldsymbol{\theta}}$ denotes a maximum likelihood estimate based on D_u, the gradients are evaluated at $\hat{\boldsymbol{\theta}}$, and $\mathcal{F}$ denotes the *Fisher information* matrix, given by

$$\mathcal{F} = E_{\mathbf{x} \sim P(\mathbf{x}|\hat{\theta})} \left[\nabla_\theta \log P(\boldsymbol{x}|\boldsymbol{\theta}) \nabla_\theta \log P(\boldsymbol{x}|\boldsymbol{\theta})^T \right], \tag{17}$$

where again the gradients are evaluated at $\hat{\boldsymbol{\theta}}$. By exponential embedding, one can construct a family of infinitely divisible kernels (see e.g. [40] for these terms) based on the naive Fisher kernel. Several motivations have been given for the Fisher kernel (see [46],[44]), none of them are easily accessible or in the scope of this report. Probably the most direct line goes via an information-theoretic perspective, given by the authors of [46] in an unpublished workshop talk. This motivation is picked up and extended in recent, ongoing work by the author (see [80]). There, a number of ways are suggested of how one might improve upon the basic Fisher kernel, although these are not yet sufficiently tested empirically. The Fisher kernel has been applied successfully to discrimination between protein families ([46], [43]), where the proteins are represented by their amino acid sequence and families are fitted using *hidden Markov models (HMM).* It has also been applied to document retrieval [42]. Attempts to apply the Fisher kernel to the case $X = \mathbb{R}^d$, where $P(\boldsymbol{x})$ is fitted by a Gaussian mixture, are reported to have failed so far (e.g. [87]). Whether this is a problem of the way Fisher kernels make use of the generative information, or more to do with the nature of the Fisher kernel being a very crude approximation to an information score, the latter still being sensible for mixture models, is unclear so far. In [80], suggestions towards more accurate approximations of the underlying (intractable) information score in case of mixture models will be given.

To conclude, the Fisher kernel covers new ground in that it is a general technique of using information from generative models within diagnostic classification schemes, as opposed to the straightforward scheme dictated by the sampling paradigm, in which all classes have to be modeled separately (see subsection 1.3.1). While in the case $X = \mathbb{R}^d$, diagnostic methods often outperform generative methods by far, the latter are still the primary option in cases where X cannot be feasibly imbedded into an $\mathbb{R}^d$ (e.g. X consists of variable-length sequences, for example in speech recognition or BioInformatics tasks). Support Vector (or Gaussian process) classification, together with a Fisher kernel, can outperform a generative scheme (using comparable modeling efforts) significantly, as has been demonstrated in [43]. The claim of the authors in [46], namely that the Fisher kernel consistently outperforms an "equivalent" generative scheme, can only be proved under certain strong assumptions (not made very clear in the paper) which often do not hold in practice. The geometry of feature spaces induced by kernels is imperfectly understood so far, although some work has been done in this direction (e.g. [13]). Since kernel methods are essentially linear machines in these feature spaces, understanding properties of this geometry might be valuable for encoding available prior knowledge about a task (e.g. [75]). Haussler [40] gives a comprehensive introduction to the problems of kernel design and suggests general methods to construct kernels for "unusual" X (e.g. containing discrete structures) which are different from the Fisher kernel.

We finally remark that the Fisher kernel can be seen as an instance of regularization dependent on the input distribution (see subsection 1.3.3). Details can be found in [81].

3.6 Restricted Bayes Optimal Classification

Tong and Koller [91] suggest a general framework for combining generative and diagnostic methods for classification, which differs from Bayesian analysis with conditional priors (see subsection 1.3.3). The usual framework for regularized discrimination uses a *loss function* $L(h(\boldsymbol{x}), t)$ (where $h(\boldsymbol{x})$ is a hypothesis) and a *regularization functional* $\mathcal{R}(h)$, both mapping to the positive real axis. $\mathcal{R}$ is used to enforce characteristics of hypotheses that we a-priori believe in, by *penalizing* hypotheses violating these characteristics with larger values. According to Occam's razor (see section 1), many regularization functionals actually penalize complicated hypotheses[31]. The idea is now to select a hypothesis which minimizes the tradeoff

$$E_{P_{emp}}[L(h(\boldsymbol{x}), t)] + \lambda \mathcal{R}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(\boldsymbol{x}_i), t_i) + \lambda \mathcal{R}(h), \tag{18}$$

where $P_{emp}(\boldsymbol{x}, t) = n^{-1} \sum_i \delta((\boldsymbol{x}, t), (\boldsymbol{x}_i, t_i))$ denotes the *empirical distribution* of the data D_l, and λ is some tradeoff parameter, e.g. chosen by cross-validation. Examples of this framework include Support Vector and MAP Gaussian process classification. In general, the MAP approximation to Bayesian discrimination (see subsection 2.4) falls into this class.

For *restricted Bayes optimal classification*, Tong and Koller use a *generative* method to estimate the joint data distribution $P(\boldsymbol{x}, t)$ from the complete observed data D_l, D_u. Call the estimate $\hat{P}(\boldsymbol{x}, t)$. Now, instead of minimizing (18), they suggest to minimize

$$E_{\hat{P}}[L(h(\boldsymbol{x}), t)] + \lambda \mathcal{R}(h). \tag{19}$$

[31] However, it seems to be frequently overlooked by some researchers criticizing the "subjectivity" of Bayesian priors, and preferring Occam regularization, that the notion of "complexity" depends very much on the task, i.e. on the *prior knowledge* we have about it.

In other words, they replace the empirical loss by the expected loss under the joint estimate $\hat{P}$. They prove some interesting theorems, giving yet another interpretation to *maximum margin hyperplanes*, a generalization of which is Support Vector classification (e.g. [96]). Namely, suppose we estimate $P(\boldsymbol{x}|t)$ by Parzen Windows with Gaussian kernels, i.e. by the sum of radial Gaussians centered on the points $\boldsymbol{x}_i$ of D_l, with common width σ. For a given $\sigma > 0$, let h_σ be the hyperplane in X which attains the lowest error $E_{\hat{P}(\mathbf{x},t|\sigma)}[h(\boldsymbol{x}) \neq t]$. This amounts to using the *zero-one loss* $L(h(\boldsymbol{x}),t) = I_{\{h(\mathbf{x})\neq t\}}$. Then they show that for $\sigma \to 0$, h_σ converges to the maximum margin hyperplane, i.e. the one hyperplane which classifies the data D_l correctly and lies most distant from the convex hulls of positive and negative points (in two-class classification).

We can compare this framework directly with the MAP approximation to Bayesian analysis with conditional priors. There, we employ the *negative log likelihood loss* $L(h(\boldsymbol{x}),t) = -\log P(t|\boldsymbol{x},h)$. Standard supervised MAP classification then amounts to minimizing (19) with $\mathcal{R}(h) = -\log P(h)$, where $P(h)$ denotes the prior distribution for h. Let us now assume that we model the input distribution $P(\boldsymbol{x})$ by $P(\boldsymbol{x}|\boldsymbol{\mu})$, and that we employ *conditional priors* $P(h|\boldsymbol{\mu})$ (see subsection 1.3.3). Then, it is easy to show that MAP within this modified data generation model amounts to minimizing (19) with

$$\mathcal{R}(h) = -\log \int P(h|\boldsymbol{\mu})P(\boldsymbol{\mu}|D_u,\boldsymbol{X}_l)\,d\boldsymbol{\mu}. \tag{20}$$

In others words, while restricted Bayes optimal classification modifies the empirical loss part in (19) based on a generative model fitted to the input data, in MAP with conditional priors we modify the regularization functional, i.e. the "effective" prior. Note that if the labeled dataset D_l is small, we would not expect that changing the loss part in (19) has a major effect on the final choice, as is somehow reflected in the experimental results reported in [91].

3.7 Transduction

Transductive inference, as opposed to *inductive inference*, is a principle which has been introduced into learning theory by Vapnik (see [94],[96]). Suppose we are given a labeled training set D_l as well as a set of test points D_u,[32] and we are required to predict the labels of the test points. The traditional way is to propose the existence of a latent function, linking training and test points via marginal dependence, then infer this function (or a posterior distribution) by *induction* and the latent labels by *deduction* (in most cases, by simple evaluation of the function). However, Vapnik's principle is that in order to solve a problem, one should not come up with *subproblems* which are harder than the whole, and *transduction*, i.e. estimating the test labels *directly* from D_l and D_u, is at least

[32] In the original formulation of transduction, the test points are the only additional points we have from $P(\boldsymbol{x})$. Later algorithms consider the more realistic case where the test set is only a subset of D_u.

not harder than *induction* to infer the latent function. From a philosophical viewpoint, this principle leading to transductive inference is very appealing.

Vapnik [96] goes on to try to prove PAC bounds specifically tailored for the transduction case. The technical details are messy, but the broad idea is to view the whole set of input points (from D_l and D_u) as a basic pool from which $n = |D_l|$ points are drawn at random and without replacement. These points are labeled then, i.i.d. according to $P(t|\boldsymbol{x})$. The points remaining in the pool form D_u. One now can employ concentration inequalities on the multinomial distribution and other common tools from *Vapnik-Chervonenkis (VC) theory* to derive large deviation bounds between the empirical error on D_l and the empirical error on D_u. We find it rather difficult to compare these bounds with the tightest VC "induction" bounds known so far.

From these transduction bounds, Vapnik derives an algorithmic scheme for transduction in an attempt to transfer the notion of *large margin discrimination* from supervised learning to the transduction case. In case of binary classification ($T = \{-1, +1\}$) with linear discriminants $(\boldsymbol{w}, b)$, $\|\boldsymbol{w}\| = 1$, the scheme works as follows. Let $T_u = \{t_{n+1}, \dots, t_{n+m}\}$ denote the latent labels on the points in D_u. For a discriminant $(\boldsymbol{w}, b)$, we define the *artificial empirical margin (ae-margin)* as

$$\rho_{art}(\boldsymbol{w}, b) = \max_{T_u} \min_{i=1,\dots,n+m} t_i(\boldsymbol{w}^T \boldsymbol{x}_i + b). \tag{21}$$

Let us suppose, for simplicity, that there exists a discriminant for which the ae-margin is positive, a necessary condition for this is that D_l is linearly separable. In words, the ae-margin is the largest empirical margin the discriminant can attain on any completion of the data D_l, D_u. It is not larger than the empirical margin (e-margin) on D_l, which is defined $\rho(\boldsymbol{w}, b) = \min_{i=1,\dots,n} t_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)$. Vapnik's transduction scheme in this case is to choose a discriminant $(\boldsymbol{w}, b)$ which *maximizes* the ae-margin. This should be compared to Vapnik's induction scheme, in which we choose $(\boldsymbol{w}, b)$ to maximize the e-margin. While the e-margin has an intuitive interpretation as sort of an estimator of the *(true) margin* (being $E_{t,\mathbf{x}}[ty(\boldsymbol{x})]$, where $y(\boldsymbol{x})$ is the discriminant, in our case $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$), with the latter obviously closely related to the generalization error, we do not know of such a link between the ae-margin and the generalization error.

Bennett et al [8] suggest a variant of Vapnik's scheme for the case of linear discriminants (i.e. SVM). They focus on a variant of SVM which employs the 1-norm $\|\boldsymbol{w}\|_1 = \sum_j |w_j|$ for penalization (instead of the Euclidean norm $\|\boldsymbol{w}\|_2 = \|\boldsymbol{w}\|$). On purely supervised tasks, this variant has been found to perform similarly to Vapnik's original linear SVM. In this setting, the optimization over the discriminants *and* the latent labels can be computed by mixed-integer programming. This is interesting, since a straightforward implementation of Vapnik's transduction scheme is exponential in $m = |D_u|$, and we know of no efficient realization. However, even the algorithm based on mixed-integer programming does not scale well with m, for example on several of the datasets tested in [8], subsamples of size 50 were used instead of the full D_u. Experiments presented in [8] show significant improvements over using induction on D_l only, on about a third of the tasks tested. However, the experimental design is somewhat unusual, for example they choose the usually very important scaling (or variance) parameter C using a heuristic which depends on $n + m$ only, instead of adapting it to the data. In [25], the authors suggest a more efficient variant, using a nonconvex quadratic problem, local solutions of which can be found by block-coordinate descent algorithms. The most substantial drawback is, however, that the scheme does not seem to be "kernelizable", i.e. the algorithm cannot be used together with a feature space mapping. Joachims [49] presents a greedy approximative implementation of Vapnik's transduction scheme, again for the case of linear discriminants (or SVM). The algorithm is not guaranteed (or expected) to find the true optimum and can get stuck in poor local optima. However, it runs much faster than the algorithm of [8], especially if D_u is large, furthermore it can be used with nonlinear kernels. The author presents experiments on text classification tasks.

Jaakkola et al [44],[45] suggest an interesting transduction algorithm within their *minimum relative entropy (MRE) discrimination* framework. Several authors have tried to relate discriminative classifiers like the SVM to diagnostic Bayesian prediction with Gaussian processes. This is problematic because the loss function used in *Support Vector classification (SVC)* is not a proper noise model, it cannot be normalized (e.g. [78]). One way around this problem is to regard SVC as an approximation to Gaussian process classification with an unusual noise model. Another, probably more satisfying way is to drop the idea that SVC is doing Bayesian inference at all, then try to find a paradigm (which is *not* the Bayesian one) of which SVC is a natural member. The usual derivation of SVC via *margin constraints* points towards the *maximum entropy (ME) principle* (e.g. [21]) which has long been used *in parallel* to the Bayesian paradigm to induce distributions which are in some way constrained on observed data. If we combine the ME principle, originating from Statistical Physics, with the Bayesian idea of a prior, we arrive at the *minimum relative entropy (MRE) principle* (e.g. [21]). The authors of [44] apply MRE to large margin discrimination and show how SVC arises as a special case. The advantages of this view include that within MRE, there are (as within the Bayesian paradigm) natural ways to handle missing data. Transduction within MRE discrimination is straightforward, exploiting the fact that latent labels do not marginalize out contrary to the diagnostic Bayesian case with regularization independent of the input distribution (see subsection 2.4). In other words, the symmetry w.r.t. the latent labels is broken by the drive towards large margin even on points from D_u. MRE transduction resembles diagnostic EM, as discussed in subsection 2.4, and comes with similar convergence guarantees to local optima. Preliminary experiments in [44] on a task where splice sites during DNA transcription have to be predicted, show promising results. However, their algorithm *is* a transduction algorithm aiming to maximize an artificial empirical margin, therefore builds on the same theoretical foundation as Vapnik's scheme discussed above. It will be interesting to compare their algorithm to diagnostic EM on Bayesian settings employing priors on the latent functions which are conditioned on the input distribution. The MRE formulation makes it possible to consider hybrids (as has been remarked in [44]), in which the unlabeled dataset D_u is split into two subsets, one of them used with latent labels in the transduction algorithm, the other one to narrow down the prior on latent functions. We also refer to Jebara and Jaakkola [48] and the *Conditional EM* algorithm [47] for recent work in this exciting line of research.

3.7.1 A subjective critique of SLT transductive inference

A discussion of the theory behind transduction in *statistical learning theory (SLT)* (see [96]) is out of the scope of this paper. To be honest, we feel somewhat confused by Vapnik's arguments, and although much of this confusion is probably due to our own ignorance of computational or statistical learning theory, it seems to be shared (to a certain degree) by many other people working in the field of the labeled-unlabeled problem. Therefore, if in the following we put forward our uneasiness about some aspects of SLT transduction, we might be wrong or unfair on certain points, and we would be happy to get into discussions about such. On the other hand, we are quite certain about one thing: if realizable SLT transduction offers significant advantages over SLT induction (within the PAC framework used there, see subsection 1.3.3), and if this can be proven theoretically, these advantages and their foundations have to be made much more clear and transparent in publications, otherwise transduction will probably remain academic and be regarded with doubts by the majority of researchers.

First of all, diagnostic prediction on finite data needs regularization, and the favourite way to do this is to impose the existence of a latent function, then to propose a model family (or, in SLT terms, a hypothesis class) for this function as well as a prior distribution which penalizes complex functions by giving them small volume (see subsection 1.3.2). In fact, when dealing with the problem of *how* to formulize the connection between training and test points, we do not see any way to get around the assumption of a latent function and the modeling thereof based on prior knowledge, also SLT transduction needs to employ this step. Now given these assumptions, the Bayesian way of inferring information about the latent function "inductively" by computing the posterior process, then predicting the test labels "deductively" by computing the predictive distribution, is *exactly the same* as what might be called "Bayesian transduction". It is just a convenient way to write the expectation over the predictive distribution. *Any information* about the input distribution $P(\boldsymbol{x})$, *not only* the test points, can help in *diagnostic* Bayesian schemes if the latent function and the input distribution (seen as random variables) are *not* a-priori independent under the generative model, as has been discussed in 1.3.3. This, however, has nothing to do with the assumption that "Bayesian transduction" is easier than the way over induction.

SLT transduction comes from a *frequentist* viewpoint where induction means picking *one* "best" function, and deduction means evaluating it at test points[33]. It is possible that in this context, transduction is easier than the way over induction and can consistently outperform the latter, however this might simply be due to the fact that the latter is *invalid* (or at least *inaccurate*) as a way of inference in the first place.

Vapnik tries to motivate SLT transduction by presenting bounds specifically tailored for the transduction setting. While reading the formidable book [95], we have been fascinated by the way Vapnik presents his results for inductive inference. He starts from philosophical principles about the nature of learning and induction, derives PAC bounds and complexity measures from these principles using sophisticated statistical techniques and then *infers* algorithmic schemes guided by the bounds. However, this way of presentation works less well in the transduction case. Although it is difficult to compare Vapnik's transduction bounds to the best known VC "induction" bounds[34], we do not see why the former should be significantly tighter than the latter, especially in the case $m = |D_u| \gg n$ which is most important in practice. In fact, the argument that transduction is easier than the way via induction, gets weak in the case $m \gg n$, and the distinction between transduction and induction vanishes for $m \to \infty$. Also, using information about the input distribution is expected to be most effective if the labeled data is sparse. In this case, the transduction as well as the induction bounds are usually very far from being tight, sometimes even trivial.

Some authors (e.g. [102]) have criticized that while there is a connection between the true margin (as defined above in this subsection) and the generalization error of a discriminant, and VC theory on supervised settings uses this link together with the fact that the e-margin is an estimator of the true margin, there is no such motivation for the ae-margin. Especially in the case $m \gg n$ we run the risk that a discriminant which maximizes the ae-margin has a much smaller e-margin (on D_l) than other choices. Even worse, it is likely for large m that no discriminant achieves positive ae-margin, *even if* D_l is separable by some function in the hypothesis class, and in this case Vapnik suggests using a "soft ae-margin" variant, trading margin violations versus margin size. Although it is evident that a tradeoff between "believing the unlabeled data" and "believing the labeled data" (the observed labels are noisy quantities) has to be faced in order to solve the labeled-unlabeled problem, we are not convinced by the publications presently known to us that this tradeoff can reliably be based on a

[33] It is interesting to note that one of the architectures most frequently discussed in SLT publications, namely the *Support Vector machine (SVM)*, can be understood as special case of *minimum relative entropy (MRE) discrimination* (see discussion above in this subsection), within which induction means computing a MRE distribution over functions conditioned on the data, somewhat parallel to the Bayesian posterior distribution, and deduction is equivalent to Bayesian prediction, but with the posterior replaced by the MRE distribution.

[34] To us, non-experts however, these different bounds employ quite similar techniques. For example, there exists an *induction bound* (see [27]) which proceeds via a double sample, such that the original and the "ghost" sample have very different sizes. This seems to be very close to the situation from where a *transduction bound* would start.

soft ae-margin. Further research in this direction might, however, put the (soft) ae-margin on a basis as solid as the (soft) e-margin in supervised learning.

Finally, we note that Vapnik's transduction scheme seems to be quite similar in spirit to a class of estimation methods discussed e.g. in [90] (end of section 4.3.4). In these so-called *clustering methods*, the labels of the points in D_u are treated as *parameters* rather than latent variables. The method then consists of maximizing the likelihood of D_l, D_u w.r.t. model parameters *and* labels on D_u. It is quite obvious that such a procedure will exhibit *bias*, and indeed this problem has been pointed out by several authors (see citations in [90]), in some cases this ML estimator is not even asymptotically consistent. It has been reported that the method behaves better in the context of *robust* estimators, so one could speculate that it might be more suitable for large-margin discrimination. However, citing from [90], "a basic flaw in this *clustering method* for parameter estimation is the treatment of the [...] [labels on D_u] as if they were parameters, rather than treating them as missing random variables". In this sense, the approach of [44] might exhibit less bias than Vapnik's scheme in some situations, although this issue clearly needs to be looked after in greater detail.

4 Related problems

In this section, we briefly discuss some problems related to the labeled-unlabeled task and mention some work which has been done on these. This review is less detailed than the one presented in section 3 and is by no means exhaustive. The reason for discussing work on related problems is of course that we feel that many ideas from this work might be successfully applied in algorithms for the labeled-unlabeled problem.

The most trivially related problems are *supervised* and *unsupervised learning*. We have already discussed these large classes in subsection 1.1. A very prominent project for Bayesian unsupervised learning is *AutoClass* (see [38],[18]), it might be used for a straightforward implementation of baseline methods discussed in subsection 2.1. We found the discussion in [84] of quantization (probably the most important special case of unsupervised learning) in the context of source compression and rate distortion theory very helpful.

4.1 Active learning

In (pool-based) *active learning* of classification, one is given a pool D_i of input points sampled i.i.d. from $P(\boldsymbol{x})$. One also has access to an oracle, producing $t \sim P(t|\boldsymbol{x})$ upon the input $\boldsymbol{x}$, and the labels produced by different calls to the oracle are conditionally independent. The goal is the same as in supervised classification (see subsection 1.1). Of course, any algorithm for the labeled-unlabeled problem can be used for active learning: simply pick n points from D_i at random, label them using the oracle to form D_l and collect the remaining points in D_i to form D_u. However, the declared goal of active learning is to outperform such schemes. We have the freedom to call the oracle *selectively* on points from D_i, therefore by focusing early on the "most informative" points we might narrow down our belief in the (latent) relationship between $\boldsymbol{x}$ and t very fast, i.e. by using only a small number of calls to the oracle. We remark that some authors have considered non-pool-based active learning, in which one does not have access to a sample from $P(\boldsymbol{x})$. We focus on pool-based active learning (also called *query filtering*) for several reasons. Non-pool-based active learning has no obvious connections to the labeled-unlabeled problem. Within the PAC framework, it has been shown that the ability to *actively* query for labels cannot be of any significant advantage if one does not have access to a sample from $P(\boldsymbol{x})$. Furthermore, in practice, it might be unexpectedly difficult to produce a sample from $P(t|\boldsymbol{x})$ if $P(\boldsymbol{x})$ is very small[35]. All these points are discussed in detail in [31], section 1.

MacKay [53] discusses Bayesian active learning for multi-layer perceptrons. Cohn et al [19] introduce the general problem, then focus on joint density models of the kind discussed in [35] (see also [38] and subsection 2.2). A very general query filtering algorithm is *query by committee (QBC)* (see [82], also [31]), which has already been mentioned in subsection 3.2. QBC is sequential in nature, i.e. the pool has the character of a stream. For each incoming $\boldsymbol{x}$, one has to decide either to label it and place it in D_l, or to discard it. In the latter case, $\boldsymbol{x}$ may not be used again at a later time. Similar to most sequential learning (or filtering) algorithms, QBC maintains and updates a belief in (i.e. a distribution over) the latent function conditioned on the data seen so far, this belief can be seen as approximation to the optimal belief, namely the Bayesian posterior distribution. QBC judges "informativeness" of an example $\boldsymbol{x}$ by the (expected) amount of variance which would be removed in the belief if we conditioned it on $\boldsymbol{x}$ and its label. This can be estimated by sampling a committee of discriminants from the current belief, evaluating the predictions on $\boldsymbol{x}$ for all committee members and compute some measure of disagreement between these predictions. For example, if the committee consists of only two discriminants, we could employ a symmetrized variant of the relative entropy between the predictive distributions of the discriminants. It is then a matter of algorithmic design taste to derive a criterion for "use or discard", based on this measure, e.g. a threshold criterion with the threshold being annealed over time. The algorithm is stopped whenever it discards a certain (large) number of points in a row (under a small threshold).

We can speculate about how to use active learning ideas in the labeled-unlabeled problem context. For example, in subsection 2.1, we suggested starting with an unsupervised learning technique, then somehow "inject" the labeled points. It might be advantageous to inject the points in D_l in an ordering suggested by active learning "informativeness" criteria.

[35] Suppose you use a human expert to label images supposed to represent hand-written digits. For an $\boldsymbol{x}$ with very small $P(\boldsymbol{x})$, the true $P(t|\boldsymbol{x})$ might have rather low entropy (w.r.t. t), but the bitmap $\boldsymbol{x}$ is most probably a mess of pixels which the expert will not be able to associate with *any* of the digits.

4.2 Coaching. Learning how to learn

The *coaching* problem is analyzed in [86]. The goal is to infer the probabilistic relationship $\boldsymbol{x} \mapsto t$, for example to estimate the regression $E_{t\sim P(t|\mathbf{x})}[t]$. Suppose we have a third variable $\boldsymbol{z}$, and the three variables are distributed according to the unknown law $P(\boldsymbol{x}, t, \boldsymbol{z})$. We are given a complete i.i.d. sample D_l from this law, but since examples of $\boldsymbol{z}$ are assumed to be difficult or expensive to collect, we are forced to predict t from $\boldsymbol{x}$ alone in the future. A trivial approach towards coaching would be to discard the examples from $\boldsymbol{z}$ entirely and to employ a standard supervised algorithm. The authors of [86] ask if and how one can do better, using the knowledge contained in the $\boldsymbol{z}$ examples. They call $\boldsymbol{z}$ a *coaching variable*, by its potential ability to coach the estimation of $\boldsymbol{x} \mapsto t$. First of all, it is clear that the $\boldsymbol{z}$ sample cannot help if t and $\boldsymbol{z}$ are conditionally independent given $\boldsymbol{x}$. Otherwise, $\boldsymbol{z}$ contains information about t given $\boldsymbol{x}$, and the nature of this observation can be learned from the common sample.

Tibshirani and Hinton propose two different coaching schemes. *Mixture coaching* builds upon the representation

$$P(t|\boldsymbol{x}) = \int P(t|\boldsymbol{x}, \boldsymbol{z}) P(\boldsymbol{z}|\boldsymbol{x})\, d\boldsymbol{z}. \tag{22}$$

Now, if t and $\boldsymbol{z}$ are dependent given $\boldsymbol{x}$, $P(t|\boldsymbol{x}, \boldsymbol{z})$ will be quite different for different values of $\boldsymbol{z}$. Therefore it seems reasonable to try to learn a partitioning of $\boldsymbol{z}$-space and to fit experts locally to the regression on each partition. We furthermore need to learn the conditional distributions over the $\boldsymbol{z}$ partition given $\boldsymbol{x}$. The algorithm suggested by the authors is essentially a *mixture of experts* (see [50]) where the choice of expert is observed on the training set. The other scheme, *response coaching*, builds on $P(t|\boldsymbol{x}) = \int P(t, \boldsymbol{z}|\boldsymbol{x})\, d\boldsymbol{z}$. The idea is to train a model to *jointly* predict t and $\boldsymbol{z}$ given $\boldsymbol{x}$. A convient way to achieve this is to use a class of factorized models $P(t, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}) = P(t|\boldsymbol{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) P(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_2)$. The conditional dependence of t and $\boldsymbol{z}$ is represented by the *shared* parameter vector $\boldsymbol{\theta}_0$. An example would be to use regression trees for t and $\boldsymbol{z}$ which share the same partition $\boldsymbol{\theta}_0$ of $\boldsymbol{x}$-space. The art is to choose the common and separate parameters and the parameter priors (i.e. the regularization) guided by available prior knowledge or assumptions.

Response coaching can be seen as a special case of the problem of *learning how to learn* or *multitask learning* (e.g. [73],[4], [15],[85],[61]). The relationship $\boldsymbol{x} \mapsto \boldsymbol{z}$ is a second task which is learned *together* with the primary one in an attempt to employ information flow through latent, shared variables. A very general approach to this problem is suggested in [66] (the author refers to the problem as “family discovery”). Here, the model family $\{P(\boldsymbol{x}|\boldsymbol{\theta}) \,|\, \boldsymbol{\theta} \in \Theta\}$, $\Theta \subset \mathbb{R}^p$ is assumed to be a smooth, low-dimensional manifold embedded in $\mathbb{R}^p$ (this view on model families comes from *information geometry*, see e.g. [1],[63]). The task is to learn the prior $P(\boldsymbol{\theta})$ which enforces this assumption, from multiple tasks. In this work, the manifold is modeled by connecting locally linear patches using kernel smoothing. Alternatively, the *generative topographic mapping (GTM)* (see [10], also subsection 2.1), as a probabilistic generative modeling approach to *manifold learning*, could be considered in this context. Another promising approach, as suggested in [61], is to use the multiple task data to learn a covariance kernel for Gaussian process (or Support Vector) classification (see subsection 3.5).

The relation between the labeled-unlabeled problem and the coaching or learning how to learn problem is not a strong one. In multitask learning, of which coaching is a special case, we try to grasp information which is shared, on a low level, between the tasks, this can be seen as *prior learning* in a *hierarchical Bayesian* setting (e.g. [4],[61]). We have already pointed out that we think that prior knowledge, whether available through human insight or learned from other sources, is crucial for solutions to the labeled-unlabeled problem, and learning priors by combining related tasks might be useful in this context. We come back to learning how to learn in subsection 4.3, when we relate it to another method for prior learning.

4.3 Transfer of knowledge learned from a related task

Learning how to learn (as discussed in subsection 4.2) is a (largely) data-driven way to learn prior distributions for a task. The assumption is that related tasks share a common (low-level) basis. Therefore, given related tasks, among them the one we are really interested in, we might gain information about this common basis by learning to solve them all together, using models which share parameters at a low level, together with a well-chosen regularization which forces the learning algorithm to make use of these shared parameters effectively. We can then employ this information as prior knowledge for the primary task.

Another approach to prior learning employs a lot more human insight into the nature of the primary task. This insight is used to choose a convenient representation to encode the prior knowledge, and to select related tasks which we believe share much in their nature with the primary task, so that the same representation of prior knowledge applies. Given this structure, it is usually easy to learn the free parameters in the representation from related tasks and then simply plug them into an architecture for the primary one. A concrete example for a handwritten signs recognition task is given in [60]. There, the generative assumption is that an observation is created by mapping a (latent) reference image by a (latent) transformation, and each class has a small number of reference images. Given a large number of observations from the same class, it is possible to learn good reference images in a MAP fashion, the authors use a simple hillclimbing approach for multiple alignment they call *congealing*. It is also possible to learn a (posterior) distribution over transformations, conditioned on the class, by applying techniques like kernel smoothing. The authors use this distribution as prior over transformations when dealing with another class for which only very few examples are available. For example, if only *one* example of the new class is available, they take this example as a reference image for the class, then create an artificial dataset by sampling transformations from the (learned) prior and applying them to the reference image.

A motivation for learning priors in the context of the labeled-unlabeled problem has already been given in subsection 4.2. For the methods discussed in the present subsection, we need a large labeled dataset for a related task. The reason for this is that our goal is to learn priors which are very strong in the following sense. In the example above, even transformations which deform the image significantly have a chance to get a high a-priori weight, if they are supported by the data. Learning a good representation of a class from a single example is possible only if such strong invariance knowledge is available.

5 Caveats and tradeoffs

In this section we discuss some issues we think are important to be addressed when dealing with the labeled-unlabeled problem. Most of them have popped up frequently in the text above, and we use this section to collect and relate them. They include some more or less obvious *caveats* as well as *tradeoffs* that have to be faced. We think that much more insight into these issues need to be gained, probably on model tasks. Such insight may have a big payoff for algorithms dealing with the problem.

5.1 Labels as missing data

Formally, we can treat labels on points from D_u in the labeled-unlabeled problem as missing data. If we compare these to other kinds of missing data, such as missing or uncertain attribute values, we find very important differences. First of all, while missing attribute values are more or less a nuisance which should be marginalized out because of lack of information in the data, labels are the *essential target.* Our whole job is to *predict* the labels, while nuisance attributes are never the target for prediction. Second, if we talk about a missing attribute, we usually mean an attribute which is missing or uncertain only in a rather small fraction of the examples in the dataset. If an attribute is missing in the large majority of examples, one should really consider not to include it at all in the representation of examples, thereby making the model building task easier and the learning process easier to control (see discussion of IMAX in subsection 3.3). However, in interesting instances of the labeled-unlabeled problem we have that a lot more labels are missing than given.

The special character of a label as latent variable raises the question of how to *value* the information provided by labels, as compared to the information provided by an input point. Any sensible valuation will of course depend on the current belief in the unknown relationship $P(\boldsymbol{x}, t)$. For example, Castelli and Cover ([16], see also subsection 3.1) show that if we know all the class distributions $P(\boldsymbol{x}|t)$, labels have "exponential value" (w.r.t. reducing the generalization error towards the Bayes error), but such a concrete belief can usually not be gained in practice. Suppose our goal would be to learn a (soft) partitioning of X (into a reasonably small number of clusters) such that with high probability none of the clusters is cut by a class boundary. Clearly, this is only slightly easier than solving the labeled-unlabeled problem. We could apply an unsupervised technique to this problem, training on D_u only, and one could argue that in this context the role of labels (i.e. of D_l) is to point out possible flaws in the present (soft) partitioning. In this context, a label can only be valued in conjunction with other labels, and if unlabeled and labeled data suggest different partitionings in a local area, we have to find a way to conciliate between them.

The question about how to value labeled data against (abundant) unlabeled data is far from being well understood. For example, we could derive a smoothed-out estimate of $P(t|\boldsymbol{x})$ from D_l alone which would at least capture some of the label information, then treat this estimate (or belief) as one basic "unit" of information to be injected into unsupervised clustering. On the other hand, we could value each element from D_l on its own, conditioned on our belief about $P(\boldsymbol{x}, t)$ gained so far. The latter approach is clearly much more flexible, but also more "slippery". If we associate a high value with elements from D_l which change our belief in the relationship most drastically, we run the risk of not being robust to outliers and classification noise. If we prefer such elements from D_l which are most compatible with our belief so far, we may not be able to correct major inaccuracies in our belief, simply because we do not trust the label information which can help us to do so. This is a tradeoff between *robustness* and *information gain.*

We face a similar tradeoff if we start from training on the labeled data, then inject points from D_u together with pseudo-labels. If we inject a point whose label is confidently predicted using the current belief, the injection does not provide much new information and will therefore not lead to more than a minor change in the belief. The operation may be considered *robust.* On the other hand, injecting a point whose label is quite uncertain (but not completely random) given the current belief, leads to a phase in the learning process in which two (or more) quite different alternatives are tested against each other, this phase has the potential to change the belief significantly, but the risk of a wrong update is higher. However, the operation might result in a much higher *information gain.*

5.2 Diagnostic versus generative methods

Suppose we use model classes $\{P(\boldsymbol{x}|t, \boldsymbol{\theta})\}$ which are faithful for the problem at hand, i.e. the class-conditional distributions are indeed members of these classes, with latent $\boldsymbol{\theta}$. Furthermore assume that in the limit of $|D_u| = m \to \infty$, the class-conditional distributions can be identified using D_u only, this is the assumption made by Castelli and Cover [16]. In this case, we can always employ a generative architecture using the faithful model classes, together with standard EM or one of its variants (see subsection 2.2.1) to circumvent poor local maxima of the likelihood. However, this approach can exhibit severe drawbacks when applied to non-toy problems. To be faithful, the model classes together with the regularization (given by $P(\boldsymbol{\theta})$) would have to be very broad, and the posterior belief would narrow down sufficiently only for very large m. We might not have a large enough D_u available, and even if unlabeled data is abundant, the cost of dealing with large D_u and searching through broad model classes (for the MAP solution) might be much higher than we can tolerate. In order to be able to make the method work efficiently, we might have to narrow down the model classes and/or tighten the regularization, in which case we risk to run into severe *robustness* problems. Namely, if the true relation $P(\boldsymbol{x}, t)$ is far from what we a-priori believe to be possible, our method might fit a simple structure supported as well by the priors as by the abundant unlabeled data, which however is likely to be quite different from $P(\boldsymbol{x}, t)$.

Diagnostic methods model $P(t|\boldsymbol{x})$ directly, without wasting resources on modeling the class distributions. While generative methods also fit an "induced" model to $P(t|\boldsymbol{x})$, derived from the class models via Bayes' formula, this model comes with heavy requirements towards resources like training data and computing time, and often many of these resources are wasted on training aspects of the class models which do not affect the "induced" model significantly. However, diagnostic methods neglect information about $P(\boldsymbol{x})$ and focus on maximizing the *conditional likelihood* of the (labeled) data, which might be harmful if D_l is small. This distinction between generative and diagnostic methods can be made clear by comparing the diagnostic *mixture-of-experts* architecture (see [50]) against an extension of the generative method proposed in [59], as discussed in subsection 2.3. We make use of the terminology defined there. Both methods make use of the *divide-and-conquer* principle, in that they construct a soft partitioning of the input space X and fit experts which estimate $P(t|\boldsymbol{x})$ locally. The diagnostic architecture achieves a soft partitioning of X using a *gating network*[36] which is a *diagnostic* model of $P(k|\boldsymbol{x})$, and the whole architecture (i.e. experts and gating network) is trained to maximize the conditional likelihood of the data. Therefore, the architecture will divide up its resources (here: the "simple" experts, being *logistic regression models* for $P(t|\boldsymbol{x})$) to achieve a good *discrimination* of the data. For example, we would expect the gating network to "position" the experts along the true decision boundaries between classes, so that if the experts locally fit the boundary well, the total mixture gives a decent estimate of $P(t|\boldsymbol{x})$ *globally*. On the other hand, if $m \gg n$, the generative architecture partitions X largely by fitting a mixture to $P(\boldsymbol{x})$. If the component models $P(\boldsymbol{x}|k, \boldsymbol{\theta})$ are unimodal (e.g. Gaussian), most of them will fit clusters in D_u. This means that the architecture tends to position the experts in the middle of such clusters, generally far from the decision boundaries between classes. It can be argued that this positioning is not a very good divide-and-conquer strategy towards improved *discrimination*.

By using regularization depending on the input distribution (see subsection 1.3.3), we can in principle make use of unlabeled data in diagnostic architec-

[36] Or a hierarchy of such, in case of the more powerful *hierarchical mixture-of-experts*, see [50].

tures. This invariably requires that we model the input distribution $P(\boldsymbol{x})$, therefore apart from the difficult task to construct sensible conditional priors for $\boldsymbol{\theta}$ (notation from subsection 1.3.3) based on prior knowledge, we also face a tradeoff of how to distribute resources between training the diagnostic models and the models for $P(\boldsymbol{x})$. Furthermore, when constructing the conditional priors, we can enforce prior assumptions to various degrees of strength, and this induces a weighting of information from D_l against such from D_u in the final prediction. Again, we face a tradeoff between *robustness* (weak a-priori influence of $P(\boldsymbol{x})$ on the belief in $\boldsymbol{\theta}$) and *information gain* from D_u (strong a-priori influence of $P(\boldsymbol{x})$ on the belief in $\boldsymbol{\theta}$).

5.3 The sampling assumption

The generative assumption for D_l, D_u is detailed in subsection 1.2. It is equivalent to the one used in the context of *transduction* (see subsection 3.7). First sample $n + m$ points i.i.d. from $P(\boldsymbol{x})$. Then, pick n points from this pool at random and without replacement, label these points and put them in D_l. The remaining pool becomes D_u.

While this assumption is fulfilled or at least reasonable for many real-world tasks, we note that it might be violated in certain settings. For example, in situations where labeled data is sparse because the process of labeling input points is very expensive, it is often *not* reasonable to assume that the input points to be labeled in the end (to form D_l) are picked at random from a large pool. Often, they are *selected* to be somewhat *representative*, where "representativeness" is judged using fluffily defined measures based on insight into the problem.

In some cases it may be possible to incorporate the selection process into the generative model. In most situations, however, we will stick with the standard i.i.d. generative assumption even at the risk of ignoring a bias in the sample D_l. We can try to use valuation rules (see subsection 5.1) in order to alleviate this bias. For example, if we knew characteristics of the selection process for D_l, we could apply them to resample D_u, i.e. inject more representative points earlier.

6 Conclusions

With respect to some key aspects, the labeled-unlabeled problem lies in the middle between well-founded areas. The problem itself is situated somewhere between *unsupervised* and *supervised learning*. It is our opinion that prior knowledge is crucial for a labeled-unlabeled method, but the *roles* of prior knowledge in unsupervised and supervised learning are very different, as discussed in subsection 1.2. Supervised learning can be quite robust against false or inaccurate prior knowledge, whereas solutions delivered by unsupervised methods very much depend on the encoded prior knowledge. In the labeled-unlabeled problem, we expect robustness to a certain degree, although the sparseness of D_l and the abundance of unlabeled data makes usage of unsupervised techniques almost obligatory.

There are two basic paradigms for supervised learning, the *diagnostic* and the *sampling* one (see subsection 1.3). The significances of these paradigms for the labeled-unlabeled problem have been discussed in subsection 5.2. In order to make use of unlabeled data in diagnostic methods, input-dependent regularization and therefore, to a certain degree, modeling of (aspects of) the input distribution is necessary (see subsection 1.3.3). Again, we think it is most promising to combine methods from the diagnostic and the sampling paradigm, i.e. to employ both diagnostic and generative model families in one architecture, in order to attack the labeled-unlabeled problem.

We think that the role of the class label as latent variable is a very special and intricate one (see subsection 5.1), and therefore in general we *cannot* treat the labeled-unlabeled problem in the same way as other common problems of missing or uncertain data. We believe that *prior knowledge* is of central importance, although the role it plays for a method might strongly vary between solutions (and tasks). Although powerful ways to encode prior knowledge have been proposed in the context of unsupervised learning, genuinely new "interfaces" towards supervised methods have to be found, either to "inject" label information into an unsupervised setting, or to adapt the distance of a supervised method using information about the input distribution, or to do something "in between". Also, methods of learning priors (see subsections 4.2 and 4.3) and ideas of using redundant representations of examples, such that the kind of redundancy is strongly linked to class identity (see subsection 3.3), may be applicable successfully.

Acknowledgments

We thank Chris Williams, Amos Storkey, Ralf Herbrich, Hugo Zaragoza, Neil Lawrence, Tong Zhang and Kristin Bennett for helpful discussions. The author gratefully acknowledges support through a research studentship from *Microsoft Research Ltd.*

References

[1] Shun-ichi Amari. *Differential-Geometrical Methods in Statistics.* Number 28 in Lecture Notes in Statistics. Springer, 1st edition, 1985.

[2] Shun-ichi Amari. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9):1379–1408, 1995.

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

[4] Jonathan Baxter. A Bayesian/information theoretic model of bias learning. In *Proceedings of COLT*, pages 77–88, 1996.

[5] M. Beal and Z. Ghahramani. Variational inference for Bayesian mixtures of factor analyzers. In *Advances in Neural Information Processing Systems 12.* MIT Press, 1999.

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

[7] Suzanna Becker. JPMAX: Learning to recognize moving objects as a model-fitting problem. In *Advances in NIPS 7*, pages 933–940, 1995.

[8] Kristin Bennett and Ayhan Demirez. Semi-supervised Support Vector machines. In Kearns, Solla, and Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 368–374. MIT Press, 1998.

[9] James O. Berger. *Statistical decision theory and Bayesian analysis.* Springer, 2nd edition, 1985.

[10] C. Bishop, M. Svensén, and C. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–235, 1998.

[11] Christopher Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[12] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with Co-Training. In *Proceedings of COLT*, 1998.

[13] Christopher J. C. Burges. Geometry and invariance in kernel based methods. In *Advances in Kernel methods – Support Vector learning.* MIT Press, Cambridge, 1998.

[14] Christopher J.C. Burges. A tutorial on Support Vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[15] Rich Caruana. Learning many related tasks at the same time with backpropagation. In *Advances in Neural Information Processing Systems 7.* MIT Press, 1995.

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

[17] Zehra Cataltepe and Malik Magdon-Ismail. Incorporating test inputs into learning. In *Advances in Neural Information Processing Systems 10.* MIT Press, 1997.

[18] Peter Cheeseman and John Stutz. Bayesian classification (AutoClass): Theory and results. In Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining.* MIT Press, 1996.

[19] David Cohn, Zoubin Ghahramani, and Michael Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[20] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of EMNLP*, 1999.

[21] Thomas Cover and Joy Thomas. *Elements of Information Theory.* Wiley Series in telecommunications. Wiley, New York, 1st edition, 1991.

[22] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. In et. al. E. F. Dedewicz, editor, *Statistics and decisions*, pages 205–237. Oldenburg Verlag, Munich, 1984.

[23] A. P. Dawid. Properties of diagnostic data distributions. *Biometrics*, 32:647–658, 1976.

[24] Virginia de Sa. Learning classification with unlabeled data. In Cowan, Tesauro, and Alspector, editors, *Advances in Neural Information Processing Systems 6.* Morgan Kaufmann, 1993.

[25] Ayhan Demirez and Kristin Bennett. Optimization approaches to semi-supervised learning. In M. Ferris, O. Mangasarian, and J. Pang, editors, *Applications and Algorithms of Complementarity.* Kluwer Academic Publishers, Boston, 2000.

[26] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[27] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition.* Applications of Mathematics: Stochastic Modelling and Applied Probability. Springer, 1st edition, 1996.

[28] R. O. Duda and P. T. Hart. *Pattern Recognition and Scene Analysis.* Wiley, 1973.

[29] B. S. Everitt. *In introduction to latent variable models.* Chapman and Hall, 1984.

[30] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the 13th international conference*, 1996.

[31] Yoav Freund, H. Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the Query By Committee algorithm. *Machine Learning*, 28:133–168, 1997.

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

[33] 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.

[34] Zoubin Ghahramani and Geoffrey Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1996.

[35] Zoubin Ghahramani and Michael Jordan. Supervised learning from incomplete data via an EM approach. In Cowan, Tesauro, and Alspector, editors, *Advances in Neural Information Processing Systems 6.* Morgan Kaufmann, 1993.

[36] Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *International Joint Conference on Machine Learning*, 2000.

[37] P.J. Green and Bernhard Silverman. *Nonparametric Regression and Generalized Linear Models.* Monographs on Statistics and Probability. Chapman and Hall, 1994.

[38] Robin Hanson, John Stutz, and Peter Cheeseman. Bayesian classification theory. Technical Report FIA-90-12-7-01, NASA Ames Research Center, 1990.

[39] T. Hastie and Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

[40] David Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California, Santa Cruz, July 1999. Available at `http://www.cse.ucsc.edu/~haussler/pubs.html`.

[41] Geoffrey E. Hinton and Radford M. Neal. A new view on the EM algorithm that justifies incremental and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1997.

[42] Thomas Hofmann. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In Solla, Leen, and Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 914–920. MIT Press, 1999.

[43] Tommi Jaakkola, David Haussler, and M. Diekhans. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of ISMB*, 1999.

[44] Tommi Jaakkola, Marina Meila, and Tony Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems 12*. MIT Press, 1999.

[45] Tommi Jaakkola, Marina Meila, and Tony Jebara. Maximum entropy discrimination. Technical Report MIT-AITR 1668, Massachusetts Institute of Technology, Cambridge, August 1999. Available at `http://www.ai.mit.edu/~tommi/papers.html`.

[46] Tommi S. Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, 1998.

[47] Toni Jebara. On reversing Jensen's inequality. In *Advances in Neural Information Processing Systems 13*, 2000.

[48] Toni Jebara and Tommi Jaakkola. Feature selection and dualities in Maximum Entropy Discrimination. In *Proceedings of UAI*, 2000.

[49] Thorsten Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods*. MIT Press, 1998.

[50] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[51] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, 1994.

[52] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecci. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[53] David MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1991.

[54] David MacKay. Introduction to Gaussian processes. Technical report, Cavendish Laboratories, Cambridge University, 1997. Available at `http://wol.ra.phy.cam.ac.uk/mackay/README.html`.

[55] Andrew McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. In *International Joint Conference on Machine Learning*, 1998.

[56] P. McCullach and J.A. Nelder. *Generalized linear models.* Monographs on Statistics and Applied Probability. Chapman and Hall, 1st edition, 1983.

[57] 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.

[58] G. McLachlan and K. Basford. *Mixture Models.* Marcel Dekker, New York, 1988.

[59] 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, 1996.

[60] Erik Miller, Nicholas Matsakis, and Paul Viola. Learning from one example through shared densities on transforms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[61] Thomas Minka and Rosalind Picard. Learning how to learn is learning with point sets. Unpublished manuscript. Available at `http://wwwwhite.media.mit.edu/~tpminka/papers/learning.html`., 1997.

[62] Tom Mitchell. The role of unlabeled data in supervised learning. In *Proceedings of the 6th International Colloquium on Cognitive Science*, 1999.

[63] Michael Murray and John Rice. *Differential Geometry and Statistics.* Number 48 in Monographs on Statistics and Applied Probability. Chapman and Hall, 1st edition, 1993.

[64] Kamal Nigam and Rayid Ghani. Understanding the behaviour of Co-Training. Submitted to KDD-2000 Workshop on Text Mining, 2000.

[65] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 1998.

[66] S. Omohundro. Family discovery. In *Advances in Neural Information Processing Systems 8*. MIT Press, 1995.

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

[68] Magnus Rattray. A model-based distance for clustering. In *Proceedings of IJCNN*, 2000.

[69] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–239, 1984.

[70] Brian D. Ripley. *Pattern Recognition for Neural Networks.* Cambridge University Press, 1996.

[71] M. Sahani. *Latent Variable Models for Neural Data Analysis.* PhD thesis, California Institute of Technology, 1999.

[72] R. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the 11th annual conference on computational learning theory*, 1998.

[73] J. Schmidhuber. On learning how to learn learning strategies. Technical Report FKI-198-94, Technische Universität München, 1995.

[74] B. Schölkopf, J. Shawe-Taylor, A. Smola, and R. Williamson. Kernel-dependent Support Vector error bounds. In *Proceedings of ICANN 9*. IEE Conference Publications, 1999.

[75] Bernhard Schölkopf, P. Simard, Alexander Smola, and Vladimir N. Vapnik. Prior knowledge in Support Vector kernels. In *Advances in Neural Information Processing Systems 10*, 1997.

[76] Dale Schuurmans. A new metric-based approach to model selection. In *Proceedings of AAAI*, 1997.

[77] Dale Schuurmans and Finnegan Southey. An adaptive regularization criterion for supervised learning. In *International Joint Conference on Machine Learning*, 2000.

[78] Matthias Seeger. Bayesian model selection for Support Vector machines, Gaussian processes and other kernel classifiers. In *Advances in Neural Information Processing Systems 12*. MIT Press, 1999.

[79] Matthias Seeger. Annealed Expectation-Maximization by Entropy Projection. Available at `http://www.dai.ed.ac.uk/~seeger/papers.html.`, 2000.

[80] Matthias Seeger. Covariance kernels from Bayesian generative models. In preparation. Check `http://www.dai.ed.ac.uk/~seeger/papers.html.`, 2000.

[81] Matthias Seeger. Input-dependent regularization of conditional density models. Submitted to ICML 2001. Available at `http://www.dai.ed.ac.uk/~seeger/papers.html.`, 2000.

[82] H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of 5th workshop on COLT*, pages 287–294. Morgan Kaufmann, 1992.

[83] 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.

[84] Noam Slonim and Naftali Tishby. Agglomerative Information Bottleneck. In *Advances in Neural Information Processing Systems 12*. MIT Press, 1999.

[85] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems 8*. MIT Press, 1996.

[86] Robert Tibshirani and Geoffrey Hinton. Coaching variables for regression and classification. Technical report, University of Toronto, September 1995.

[87] M. Tipping. Deriving cluster analytic distance functions from Gaussian mixture models. In *Proceedings of the 9th International Conference on ANN*. IEE, London, 1999.

[88] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[89] Naftali Tishby, Fernando Pereira, and William Bialek. The Information Bottleneck method. In *Proceedings of the 37th annual Allerton Conference on Communication, Control, and Computing*, 1999.

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

[91] Simon Tong and Daphne Koller. Restricted Bayes optimal classifiers. In *Proceedings of AAAI*, pages 658–664, 2000.

[92] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282, 1998.

[93] N. Ueda, R. Nakano, Z. Ghahramani, and G. Hinton. SMEM algorithm for mixture models. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1998.

[94] Vladimir N. Vapnik. *Estimation dependences based on Empirical Data*. Springer Series in Statistics. Springer, 1st edition, 1982.

[95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[96] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[97] Grace Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series. SIAM, 1990.

[98] S.R. Waterhouse and A.J. Robinson. Classification using hierarchical mixtures of experts. In *Proceedings of the 4th IEEE workshop on neural networks for signal processing*, pages 177–186, 1994.

[99] Christopher K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1997.

[100] Christopher K. I. Williams and Carl E. Rasmussen. Gaussian processes for regression. In *Advances in Neural Information Processing Systems 8*. MIT Press, 1996.

[101] A. L. Yuille, P. Stolorz, and J. Utans. Statistical physics, mixtures of distributions and the EM algorithm. *Neural Computation*, 6(2):334–340, 1994.

[102] Tong Zhang and Frank Oles. A probability analysis on the value of unlabeled data for classification problems. In *International Joint Conference on Machine Learning*, 2000.