Active Learning for Parzen Window Classifier

Olivier Chapelle
Max Planck Institute for Biological Cybernetics
Spemannstr 38, 72076 Tübingen, Germany
olivier.chapelle@tuebingen.mpg.de

Abstract

The problem of active learning is approached in this paper by minimizing directly an estimate of the expected test error. The main difficulty in this "optimal" strategy is that output probabilities need to be estimated accurately. We suggest here different methods for estimating those efficiently. In this context, the Parzen window classifier is considered because it is both simple and probabilistic. The analysis of experimental results highlights that regularization is a key ingredient for this strategy.

1 Introduction

In the standard supervised framework, the goal is to estimate a function based on a given training set. *Active learning* is an extension of this framework where the learning machine does not only receive the training points passively, but can also choose the points to be included in the training set. An active learner may start with a small training set and at each iteration carefully selects one or several points for which it asks the labels to a human expert.

The main motivation for active learning is that it usually requires time and/or money for the human expert to label an example and those resources should not be wasted to label non-informative samples, but be spent on interesting ones.

Optimal Experimental Design (Fedorov, 1972) is closely related to active learning as it attempts to find a set of points such that the variance of the estimate is minimized. In contrast to this "batch" formulation, the term *active learning* often refers to an incremental strategy (Roy & McCallum, 2001; Sugiyama & Ogawa, 2000; Cohn et al., 1995; Sung & Niyogi, 1995; MacKay, 1992b).

We will concentrate on *pool-based* active learning (also called *selective sampling*): the learner can only query the labels of some points which belong to a large unlabeled set. Note that in this standard definition of pool-based active learning, the search is greedy: at each iteration, the goal is to find *one* point which will result in the smallest expected generalization error when added to the training set.

There has been various heuristic proposed for active learning, such as *uncertainty sampling* (Lewis & Gale, 1994) or *version space minimization* (Freund et al., 1997; Tong & Koller, 2001). However, ideally, the aim is to choose the point such that the expected test error is minimized (Roy & McCallum, 2001). Such an approach has also been suggested in (Schohn & Cohn, 2000) in the context of *Support Vector Machines* learning, but the authors argued that it would be computationally intractable.

However this "optimal" approach has been implemented for SVMs and Parzen window classifier in (Chapelle, 2003, chapter 8), but it performed terribly. In this paper, we investigate why the naive implementation of this active learning strategy does not works and suggest two remedies based on semi-supervised learning and regularization.

The paper is organized as follows: section 2 presents the strategy which consists in minimizing the expected test error and section 3 shows how to apply it to the Parzen window classifier. One of the reason for considering this simple classifier is that a more sophisticated classifier might introduce a bias in our analysis of active learning. In section 4, we propose a first improvement that takes into account the unlabeled points for the class conditional density estimates, and finally, section 5 introduces approaches based on regularization.

Note that for convenience, experimental results will be presented all along the paper in order to assess immediately the performance of a new method.

2 Optimal active learning

The optimal active learning strategy we present here has been described for instance in (Schohn & Cohn, 2000; Roy & McCallum, 2001). It consists in querying the label of the point, that once incorporated in the training set, will yield the lowest expected test error.

Let $D = (\mathbf{x}_i, y_i)_{1\leq i\leq n}$ be the training samples. Suppose that the generalization error of the function learned on this training set can be estimated. Let us denote by $T(D)$ such an estimate (which, of course, depends also on the learning algorithm). The optimal active learning strategy would be the following,

1. Train the classifier using the current training set of n points and get the hypothesis $\hat{f}_n$.
2. Fix a point $\mathbf{x}$ in the unlabeled set
   (a) Fix a label y and add the point $(\mathbf{x}, y)$ in the training set
   (b) Retrain the classifier with the additional point.
   (c) Estimate the generalization error $T(D \cup (\mathbf{x}, y))$.
   (d) Estimate the posterior probability $\hat{P}(y|\mathbf{x}, \hat{f}_n)$ of the new point under the hypothesis $\hat{f}_n$.
   (e) Compute the expected generalization error $\bar{T}_{\mathbf{x}} = \sum_y \hat{P}(y|\mathbf{x}, \hat{f}_n) T(D \cup (\mathbf{x}, y))$.
3. Choose for labeling the point $\mathbf{x}$ which has the lowest expected generalization error $\bar{T}_{\mathbf{x}}$ and add it to training set.

There are several problems with this strategy. Beside computational difficulties (at a first sight, a lot of retrainings are necessary), the fundamental problem is how to compute T and $\hat{P}(y|\mathbf{x}, \hat{f}_n)$. Note that in the rest of the paper, we will refer indifferently to $\hat{P}(y|\mathbf{x})$ as posterior or output probability.

For classification, the posterior probability can be used directly (Roy & McCallum, 2001; Zhu et al., 2003) to compute T,

$$T = \frac{1}{n_u} \sum_{i=n+1}^{N} \left(1 - \max_{y\in\{-1,1\}} P(y|\mathbf{x}_i, \hat{f}_n)\right), \qquad (1)$$

where $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N$ is the set of unlabeled data and $P(y|\mathbf{x}, \hat{f})$ is an estimate of the posterior probability for the point $\mathbf{x}$ given the function $\hat{f}_n$ learned on the training set.

Eq. (1) can be seen as the empirical counter part of

$$\frac{1}{2} \iint |y - \arg\max P(y|\mathbf{x}_i, \hat{f}_n)| \;\; dP(y|\mathbf{x}, \hat{f}_n) dP(\mathbf{x}),$$

which would be the the generalization error if $dP(y|\mathbf{x}, \hat{f}_n)$ were the true conditional distribution of y given $\mathbf{x}$.

3 Parzen window classifier

The goal is to apply this strategy with the Parzen window classifier. In this case, we have

$$\hat{P}(\mathbf{x}|y) = \frac{1}{|\{i|\ y_i = y\}|} \sum_{i,\ y_i=y} K(\mathbf{x}, \mathbf{x}_i) \qquad (2)$$

and by Bayes rule

$$\hat{P}(y|\mathbf{x}) = \frac{\sum_{i,\ y_i=y} K(\mathbf{x}, \mathbf{x}_i)}{\sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i)}, \qquad (3)$$

where K is typically a Gaussian kernel of the form (up to an irrelevant multiplicative constant),

$$K(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2 / 2\sigma^2).$$

It is thus possible to compute the estimated generalization error using (1) and perform the optimal active learning strategy described above. The reason for considering this simple classifier with active learning is that equation (3) gives directly an estimate of the posterior probability and that it does need a costly retraining when a point is added to the training set.

Experiments

Two datasets have been used for the experimental results presented in this paper: an artificial one and a real world one, and the details of the experimental setup are as indicated below.

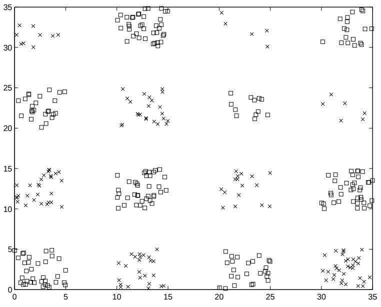

Figure 2: Toy problem: checker board dataset

Toy problem This is a modified version of the toy problem used in (Zhu et al., 2003). As plotted in figure 2, it consists of a checker board, where in each cluster, the points are drawn according to a uniform distribution and the number of point is

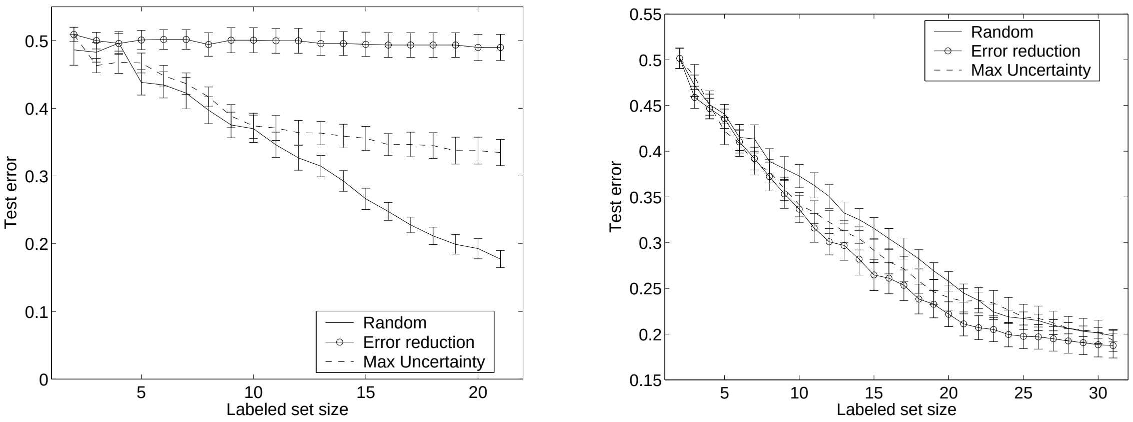

Figure 1: Test errors achieved by 3 active learning strategies on the toy problem (left) and the USPS database (right)

also drawn randomly between 1 and 40. 2 labeled points (one negative and one positive) are selected randomly and 20 samples are chosen incrementally to be labeled. Test errors are computed on the non-queried points and averaged over 100 trials. As in (Zhu et al., 2003), the variance of kernel estimate was fixed at $\sigma^2 = 2$.

Digit classification The real world database is the USPS one consisting of 7291 training samples and 2007 test ones. The training samples have been divided on 23 subsets of 317 examples each. The task is to classify digits 0 to 4 against 5 to 9. As for the toy problem, 2 random labeled points are selected and 30 samples among the 315 remaining are queried for their labels. The width of the Parzen classifier was set to $\sigma^2 = 256 \cdot (0.1)^2$, which gave the best performance in a standard supervised framework. Note that this is actually a very small value and the resulting classifier behaves almost as 1-nearest neighbor classifier.

Experimental results are provided in figure 1, where the method described in this section (entitled error reduction) is compared to random queries and to the standard max uncertainty strategy which selects the point $\mathbf{x}_i$ for which the learner is the most uncertain, i.e. whose output probability (3) is the nearest from 1/2.

The results for the error reduction strategy are really disappointing: it is not much better than random on USPS and is terrible on the toy problem.

The conjecture of why it failed is because the strategy presented in the previous section depends heavily on reliable estimates of the posterior probabilities (through the step (e) and equation (1)). For this reason, we will try in the rest of the paper to have more reliable estimates, but note that in most cases, the decision function given by $\arg\max \hat{P}(y|\mathbf{x})$ will remain unchanged (except in section 4.2).

A first observation which shows that the density estimates are not very reliable is the following: for a given point $\mathbf{x}$, the value of $P(\mathbf{x})$ can either be estimated as $1/N \sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i)$ [Parzen window on all the points] or as $\sum_y \hat{P}(\mathbf{x}|y)\hat{P}(y) = 1/n \sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i)$ [Parzen window on the labeled points]; and those two values can be quite different.

4 Class conditional density estimate using unlabeled points

As mentioned above, the standard Parzen window estimator of the class conditional densities does not take into account the unlabeled points. In other words,

$$\hat{P}(\mathbf{x}_i, y_i = 1) + \hat{P}(\mathbf{x}_i, y_i = -1) \neq \tilde{P}(\mathbf{x}_i), \qquad (4)$$

where $\tilde{P}$ is the Parzen window estimator on the labeled and unlabeled points.

We will discuss two ways to solve the "contradiction" revealed by equation (4).

4.1 Constrained Parzen window

The first idea is not to estimate both class conditional densities independently, but in such a way that equality (4) holds. In (Vapnik, 1998), it was shown that the Parzen window estimator can be seen as a solution of an optimization problem consisting of a smoothness term and a term fitting the data (the L_2 error between the empirical distribution function and the estimated one).

Based on this observation, it was suggested in (Chapelle, 2003, Chapter 7) in the context of semi-supervised learning to explicitly add equality (4) as a constraint in the optimization problem and this constrained Parzen window estimate turns out to be

$$\hat{P}_{CTR}(\mathbf{x}_i, y_i = 1) = \hat{P}(\mathbf{x}_i, y_i = 1) + \frac{1}{2}\Delta P(\mathbf{x}_i), \quad (5)$$

where ΔP is the difference between the left and right hand side of (4), i.e. $\Delta P(\mathbf{x}) = \tilde{P}(\mathbf{x}) - \sum_y \hat{P}(\mathbf{x}_i, y_i = y)$ With this new estimate of class conditional density, (4) becomes now an equality.

However, this modification raises another problem: when $\Delta P < 0$, it might happen that the output probability $\hat{P}_{CTR}(y|\mathbf{x})$ is no longer between 0 and 1. In this case, we decided to threshold the output to 0 or 1.

A middle way solution is to add only a fraction of ΔP, i.e. to replace equation (5) by,

$$\hat{P}_{CTR}(\mathbf{x}_i, y_i = 1) = \hat{P}(\mathbf{x}_i, y_i = 1) + \frac{\gamma}{2}\Delta P(\mathbf{x}_i), \quad (6)$$

where γ is chosen between 0 and 1.

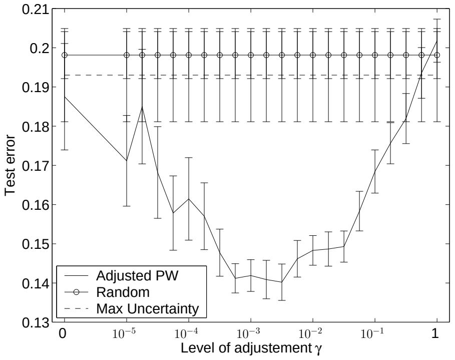

Figure 3: Test error as a function of γ in (6) after 30 points were added in the labeled set of USPS.

As plotted in figure 3, there can be a very significant improvement when γ is chosen appropriately. However, it is not clear how it should be chosen. Also, the fact of having to threshold the output probabilities because they are not always between 0 and 1 is not very satisfactory. Future research includes the derivation an improved constrained Parzen window estimate.

4.2 Expansion on the unlabeled points

A second idea is to use all the points in the expansion of the class distribution (2). First, suppose that the labels of the unlabeled points were known. Then, the class Parzen window estimate would give

$$\hat{P}(x, y|y_{n+1,\dots,N}) = \frac{1}{N} \sum_{i=1,\ y_i=y}^{N} K(\mathbf{x}, \mathbf{x}_i).$$

Introducing the variables $\lambda_i = P(y_i = 1|\mathbf{x}_i)$ and integrating over the choice of the unknown labels of the unlabeled points, we then have

$$\hat{P}(\mathbf{x}, y = 1) = \frac{1}{N} \sum_{i=1}^{N} \lambda_i K(\mathbf{x}, \mathbf{x}_i). \quad (7)$$

The λ_i for the unlabeled points are of course unknown, but we will see how to estimate them. The λ_i for the labeled points are set in this section to 0 or 1, according to the labels y_i, i.e. $\lambda_i = (y_i + 1)/2$.

Now note that by conditioning on $\mathbf{x}$ equation (7) gives the conditional probability output of a point under the Parzen window model,

$$\hat{P}(y_p = 1|\mathbf{x}_p) = \frac{\sum_{i=1}^{N} \lambda_i K(\mathbf{x}_i, \mathbf{x}_p)}{\sum_{i=1}^{N} K(\mathbf{x}_i, \mathbf{x}_p)} \equiv \tilde{\lambda}_p,$$

that we can rewrite in matrix notation as

$$\tilde{\boldsymbol{\lambda}} \equiv D^{-1} K \boldsymbol{\lambda}, \quad (8)$$

where D is a diagonal matrix with $D_{ii} = \sum_j K_{ij}$ and $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

A way to estimate $\boldsymbol{\lambda}$ is to enforce that $\lambda_i = \tilde{\lambda}_i$ for each unlabeled point $\mathbf{x}_i$. By doing so, the model is coherent. Splitting equation (8) between labeled and unlabeled blocks, this constraint writes

$$\boldsymbol{\lambda}_u = (D^{-1}K)_{u,u} \boldsymbol{\lambda}_u + (D^{-1}K)_{u,l} \boldsymbol{\lambda}_l,$$

where the subscripts l and u stand respectively for the labeled and unlabeled indices. And since $\boldsymbol{\lambda}_l = (Y + 1)/2$, we get

$$\begin{aligned} \boldsymbol{\lambda}_u &= (I - (D^{-1}K)_{u,u})^{-1} (D^{-1}K)_{u,l} (Y + 1)/2 \\ &= [(D - K)_{u,u}]^{-1} K_{u,l} (Y + 1)/2, \end{aligned}$$

which is exactly how the output probabilities are estimated in (Zhu et al., 2003).

This way of estimating of the output probabilities yield directly an active learning algorithm once it is combined with the framework presented in section 2. This algorithm was suggested in (Zhu et al., 2003) and the experimental results therein are quite impressive. An explanation for these good performances is that there is a *semi-supervised* learning step in this algorithm. Indeed, consider an unlabeled point $\mathbf{x}_i$ for which $P(\mathbf{x}_i, y_i = 1) \approx P(\mathbf{x}_i, y_i = -1) \approx 0$. Then, the constrained Parzen window estimator will correct the class conditional density estimates by adding the same value $\Delta P(\mathbf{x}_i)/2$ to both of them, whereas the semi-supervised one will choose a value λ_i between 0 and 1 according to what is the most "likely" label of $\mathbf{x}_i$.

Figure 4 confirms that significant improvements are indeed obtained using this method (referred in the plot as semi-supervised).

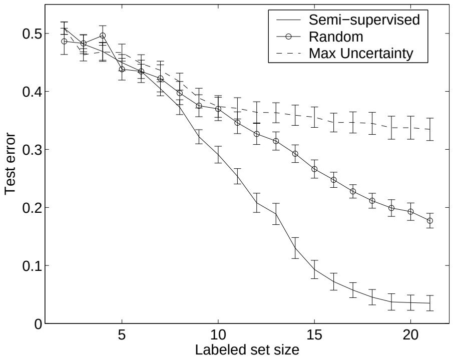

Figure 4: Results on the toy database for the method described in this section (semi-supervised), which was first introduced in (Zhu et al., 2003)

4.3 Soft margin formulation

We now consider the λ_i for the labeled points as free variables.

For a fixed value of the vector $\boldsymbol{\lambda}_l$, the posterior probabilities on the unlabeled points are given, as in the previous section by

$$\boldsymbol{\lambda}_u = (I - (D^{-1}K)_{u,u})^{-1}(D^{-1}K)_{u,l}\boldsymbol{\lambda}_l. \qquad (9)$$

We suggest to find $\boldsymbol{\lambda}_l$ as the solution of an optimization problem consisting of a likelihood term and a “coherence” term.

Firstly, conditioning on the inputs and on the output probabilities $\boldsymbol{\lambda}_l$, the log-likelihood of the labels is

$$\log P(y_{1..n}|\mathbf{x}_{1..n}, \boldsymbol{\lambda}_u) = \sum_{i=1}^{n} \frac{1+y_i}{2}\log\lambda_i + \frac{1-y_i}{2}\log(1-\lambda_i) \equiv L(\boldsymbol{\lambda}_l),$$

which should be maximized.

Secondly, note that both $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$ are estimate of the posterior probabilities and ideally those two vectors should be identical. By definition of the choice of $\boldsymbol{\lambda}_u$ in (9), we have already $\boldsymbol{\lambda}_u = \tilde{\boldsymbol{\lambda}}_u$. Even though it is impossible to get $\boldsymbol{\lambda}_l = \tilde{\boldsymbol{\lambda}}_l$, one can try to minimize the difference between $\boldsymbol{\lambda}_l$ and $\tilde{\boldsymbol{\lambda}}_l$. Their discrepancy is somehow a measure of how incoherent the model is. Since both vectors are actually estimates about the conditional distribution, $P(y|\mathbf{x})$, it seems natural to use the Kullback-Leibler divergence, which is in this case, under an independence assumption,

$$KL(\boldsymbol{\lambda}_l, \tilde{\boldsymbol{\lambda}}_l) = \sum_{i=1}^{n} \lambda_i \log\left(\frac{\lambda_i}{\tilde{\lambda}_i}\right) + (1-\lambda_i)\log\left(\frac{1-\lambda_i}{1-\tilde{\lambda}_i}\right).$$

If $\boldsymbol{\lambda}_l$ and $\tilde{\boldsymbol{\lambda}}_l$ are close enough, a first order expansion gives

$$KL(\boldsymbol{\lambda}_l, \tilde{\boldsymbol{\lambda}}_l) \approx \sum_{i=1}^{n} \frac{(\lambda_i - \tilde{\lambda}_i)^2}{\lambda_i(1-\lambda_i)} = W(\boldsymbol{\lambda}_l, \boldsymbol{\lambda}_l - \tilde{\boldsymbol{\lambda}}_l),$$

where $W(\mathbf{x}, \mathbf{y}) \equiv \sum \frac{y_i^2}{x_i(1-x_i)}$.

Let us see how to compute $\lambda_i - \tilde{\lambda}_i$. For this purpose, we introduce $S = I - D^{-1}K$, and using block matrix identities as well as (9), we get

$$\begin{aligned} \lambda_i - \tilde{\lambda}_i &= \left[(I - D^{-1}K)\begin{pmatrix}\boldsymbol{\lambda}_l \\ \boldsymbol{\lambda}_u\end{pmatrix}\right]_i \\ &= [(S_{ll} - S_{lu}S_{uu}^{-1}S_{ul})\boldsymbol{\lambda}_l]_i \\ &= [(S^{-1})_{ll}^{-1}\boldsymbol{\lambda}_l]_i \end{aligned}$$

Putting everything together, we suggest to find the vector $\boldsymbol{\lambda}_l$ which minimizes

$$-\gamma L(\boldsymbol{\lambda}_l) + W(\boldsymbol{\lambda}_l, (S^{-1})_{ll}^{-1}\boldsymbol{\lambda}_l), \qquad (10)$$

and to get $\boldsymbol{\lambda}_u$ from $\boldsymbol{\lambda}_l$ through equation (9).

Note that the minimization of (10) is a convex optimization problem as shown in appendix.

As a side remark, one might be worried by the computational complexity of this method as well as some others presented in this paper. Indeed, at each iteration, and for each candidate, the λ_i need to be re-estimated (step (b) in section 2), which would be prohibitive if those updates were done naively. However, using rank-one updates and block matrix identities, one can compute efficiently the new gradients and Hessian of the objective function (10) when a labeled point is added. From there, a Newton's step is simulated in order to update $\boldsymbol{\lambda}_l$.

Experimental results using this soft margin formulation are presented in table 1 and on both databases it did not help. This might be because those datasets are not really noisy (for the postal dataset, a hard margin SVM performs better than a soft margin one). However, in future experiments, we will experiment this algorithm on noisy datasets.

	1	10	1000	Hard margin
Checker board	20.8	20.9	21.5	21.5
USPS	16.5	15.6	15.5	15.2

Table 1: Test error as a function of the soft margin parameter γ after 10 queried points for the checker board dataset and 30 for USPS

5 Uncertain posterior probabilities

Consider the example presented in figure 5, which is quite similar to the one in (Zhu et al., 2003) and pick one point in the cluster on right hand side. Then the probability for this point to belong to either class is very small because it is far from both labeled points. However, since the point is nearer from the circle, the posterior probability estimated by Parzen window of its label being a circle is almost 1.

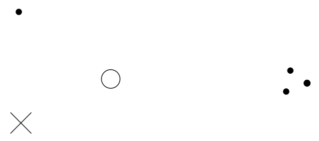

Figure 5: The unlabeled points on the right hand side will not be queried because they are (maybe wrongly) believed to be circles.

Intuitively, this is not very satisfactory: the active learning algorithm will not select a point from this cluster. This is one of the problem occurring in the approach presented above as well as in (Roy & McCallum, 2001; Zhu et al., 2003) is that it uses estimates of the posterior probabilities but ignores their variance or how reliable those estimates are.

5.1 Constrained Parzen window

The constrained Parzen window solves this problem as in regions of the space where there are unlabeled data which are far from the labeled ones, it increases the joint density estimates and the amount by which it is increased is the same for both classes. Thus in this case, the posterior probability is near from 1/2.

5.2 Regularizing

If there is an uncertainty about the posterior probability of a point, this one should be pushed towards 1/2. There are several possibilities for achieving this goal. First, let us introduce the log ratio of the posterior probabilities,

$$\alpha_i = \log \frac{P(y_i = 1|\mathbf{x}_i)}{P(y_i = -1|\mathbf{x}_i)}. \qquad (11)$$

Quadric penalty The first idea is to introduce a quadratic regularization term on $\boldsymbol{\alpha}$ in the objective function we want to minimize. In this way, the α_i which are not constrained by some other terms will have small values and thus, the corresponding λ_i nearer from 1/2.

Using the variance Suppose that there is a Gaussian error on the value of $\boldsymbol{\alpha}$ and that we know its variance $\delta\boldsymbol{\alpha}$. Then MacKay (MacKay, 1992a) suggests to replace α_i by

$$\frac{\alpha_i}{\sqrt{1 + \pi(\delta\alpha)_i^2/8}}.$$

By doing so, if $(\delta\alpha)_i$ is small, the posterior probability is almost unchanged. However, if it is large, then the new α_i is small, which means that $P(y_i = 1|\mathbf{x}_i)$ is closer to 1/2. This is exactly the desired effect.

The variance can be estimated (up to a multiplicative constant) thanks to the Hessian H of the objective function at the optimal value, $(\delta\alpha)_i^2 \propto H_{ii}^{-1}$.

Regularized Parzen window When observing n^+ positive examples and n^- negative ones, the standard way to estimate the ratio of the positive class is to use a Beta prior on the class probability, which leads to the following estimate, $\frac{n^+ + 1}{n^+ + n^- + 2}$.

Based on this observation, one can estimate the posterior probability using the Parzen window estimate as,

$$\hat{P}(Y = 1|X = \mathbf{x}_p) = \frac{\sum \delta_{y_i=1} K(\mathbf{x}_i, \mathbf{x}_p) + \varepsilon}{\sum K(\mathbf{x}_i, \mathbf{x}_p) + 2\varepsilon}, \qquad (12)$$

where ε is a small constant to be chosen.

Those three methods require a constant to be chosen, which represents the amount of regularization. We decided to the consider the last one because ε has a more direct interpretation in terms of prior probability. Also, it might be interesting to choose ε as to minimize the KL divergence between the density estimated on the unlabeled points, $\sum_{i=1}^{N} K(\mathbf{x}_i, \mathbf{x})$ and the "regularized" density on the labeled points, $\sum_{i=1}^{n} K(\mathbf{x}_i, \mathbf{x}) + 2\varepsilon$.

The results of the experiments presented in figure 6 show that this regularization is extremely useful, especially for the error reduction strategy which performed poorly without regularization (see section 3). The semi-supervised method described in section 4.2 behaves also much better with regularization. Note the local maximum in the right plot of figure 6. This is quite surprising and requires further investigation.

6 Conclusion

This paper provided an analysis of the influence of reliable posterior probabilities estimates on the performance of an active learner. In particular, it showed that regularization seems to be a very useful ingredient in those estimations.

Figure 7 shows that using this regularization, the performances achieved are not far from the best achievable ones in the case of the toy problem, and also for the USPS database after 30 queries.

The conclusions drawn from the analysis in this paper should be useful to adapt this active learning strat-

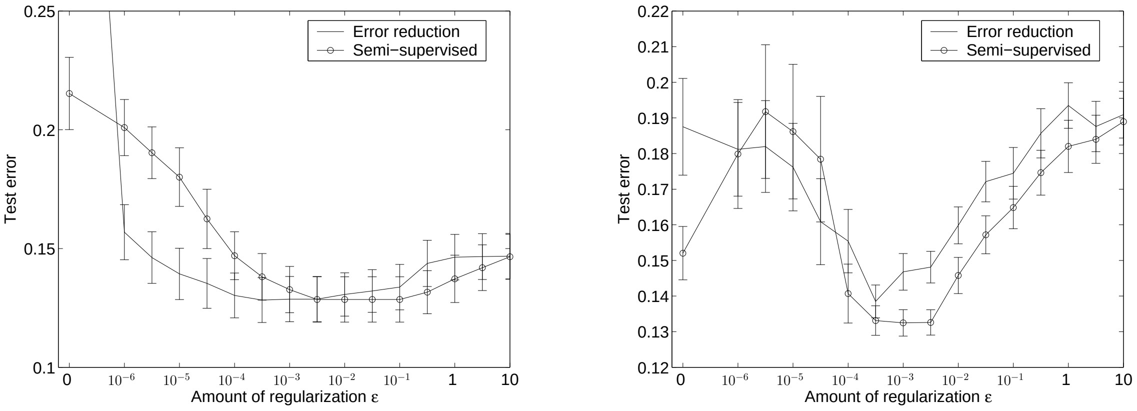

Figure 6: Experiments on the checker board dataset (left) and on USPS (right). The test error is plotted as a function of ε used to estimated the regularized posterior probability (12).

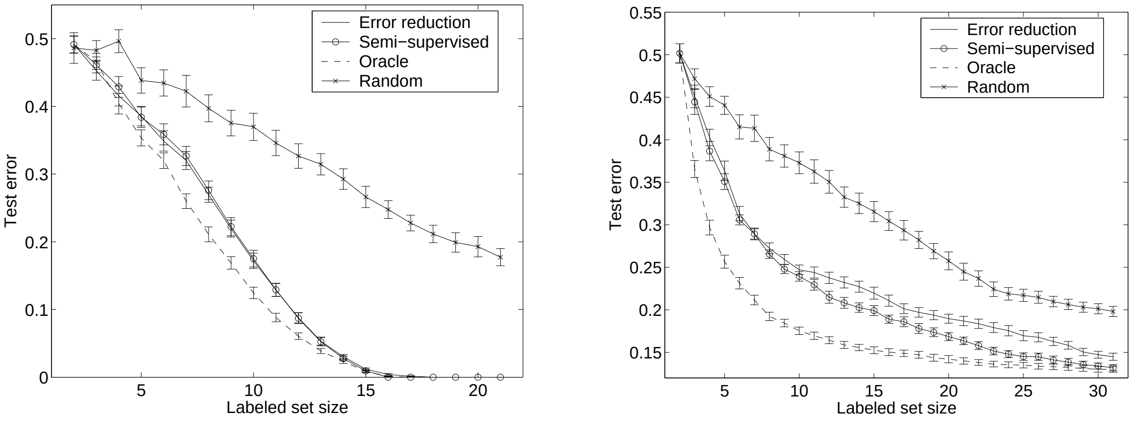

Figure 7: Performances achieved on the toy problem (left) and USPS (right). Both error reduction and semi-supervised use a regularization parameter $\varepsilon = 10^{-3}$ (see also figure 6). The Oracle strategy is the best one can achieve: it estimates the test error (1) using the *true* labels of the unlabeled set.

egy to more sophisticated classifiers such as Support Vector Machines or Gaussian Processes.

Appendix

The functional (10) is a convex function of $\boldsymbol{\lambda}_l$.

Indeed, computing the second derivatives of L, it is easy to check that L is a concave function. Concerning W, all the terms are of the form $\frac{(T\boldsymbol{\lambda})_i^2}{\lambda_i(1-\lambda_i)}$, which can be shown to be convex thanks to the following lemma

Lemma 1 *If f is a convex non-negative function on $\mathbb{R}^n$ and g is a concave positive function on $\mathbb{R}^n$, then f^2/g is convex.*

Proof: The Hessian of f^2/g is

$$\nabla^2 \frac{f^2}{g} = \frac{2f}{g}\nabla^2 f - \frac{f^2}{g^2}\nabla^2 g + \frac{2}{g^3}(g\nabla f - f\nabla g)(g\nabla f - f\nabla g)^\top,$$

which is a sum of positive definite matrices □.

We then apply the previous lemma with the convex function $f(\boldsymbol{\lambda}) = |\sum_j S_{ij}\lambda_j|$ (it satisfies Jensen's inequality) and the concave function $g(\boldsymbol{\lambda}) = \lambda_i(1-\lambda_i)$.

Instead of optimizing on $\boldsymbol{\lambda} \in [0,1]^n$, in practice we optimize on $\alpha_i = \log \lambda_i/(1-\lambda_i)$ (see also (11). This leads to an unconstrained optimization problem which is easier to solve numerically.

Note that by doing this change of variable, the objective function is not convex anymore. However, since this is a monotonic transformation, it is easy to show that the function is quasiconvex (Boyd & Vandenberghe, 2003), and can thus be minimized efficiently (there is no local minima).

Acknowlegments

References

Boyd, S., & Vandenberghe, L. (2003). *Convex optimization*. Cambridge University Press.

Chapelle, O. (2003). *Support vector machines: Induction principle, adaptive tuning and prior knwoledge*. Doctoral dissertation, LIP6.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1995). Active learning with statistical models. *Advances in Neural Information Processing Systems* (pp. 705–712). The MIT Press.

Fedorov, V. (1972). *Theory of optimal experiments*. New York: Academic Press.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*, 133–168.

Lewis, D., & Gale, W. (1994). Training text classifiers by uncertainty sampling. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3–12).

MacKay, D. (1992a). The evidence framework applied to classification networks. *Neural Computation*, *4*, 720–736.

MacKay, D. (1992b). Information-based objective functions for active data selection. *Neural Computation*, *4*, 590–604.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. *Proceedings of the International Conference on Machine Learning*.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. *Proceedings of 17th International Conference on Machine Learning* (pp. 839–846). San Francisco, CA: Morgan Kaufmann.

Sugiyama, M., & Ogawa, H. (2000). Incremental active learning for optimal generalization. *Neural Computation*, *12*, 2909–2940.

Sung, K. K., & Niyogi, P. (1995). Active learning for function approximation. *Advances in Neural Information Processing Systems* (pp. 593–600). The MIT Press.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research* (pp. 45–66).

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Zhu, X., Lafferty, J., & Ghahramani, Z. (2003). Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. *ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*.