Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions

Xiaojin Zhu* ZHUXJ@CS.CMU.EDU
John Lafferty* LAFFERTY@CS.CMU.EDU
Zoubin Ghahramani†* ZOUBIN@GATSBY.UCL.AC.UK
*School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213, USA
†Gatsby Computational Neuroscience Unit, University College London, London WC1N 3AR, UK

Abstract

Active and semi-supervised learning are important techniques when labeled data are scarce. We combine the two under a Gaussian random field model. Labeled and unlabeled data are represented as vertices in a weighted graph, with edge weights encoding the similarity between instances. The semi-supervised learning problem is then formulated in terms of a Gaussian random field on this graph, the mean of which is characterized in terms of harmonic functions. Active learning is performed on top of the semi-supervised learning scheme by greedily selecting queries from the unlabeled data to minimize the estimated expected classification error (risk); in the case of Gaussian fields the risk is efficiently computed using matrix methods. We present experimental results on synthetic data, handwritten digit recognition, and text classification tasks. The active learning scheme requires a much smaller number of queries to achieve high accuracy compared with random query selection.

1. Introduction

Semi-supervised learning targets the common situation where labeled data are scarce but unlabeled data are abundant. Under suitable assumptions, it uses unlabeled data to help supervised learning tasks. Various semi-supervised learning methods have been proposed and show promising results; Seeger (2001) gives a survey. These methods typically assume that the labeled data set is given and fixed.

In practice, it may make sense to utilize *active learning* in conjunction with semi-supervised learning. That is, we might allow the learning algorithm to pick a set of unlabeled instances to be labeled by a domain expert, which will then be used as (or to augment) the labeled data set. In other words, if we have to label a few instances for semi-supervised learning, it may be attractive to let the learning algorithm tell us which instances to label, rather than selecting them randomly. We will limit the range of query selection to the unlabeled data set, a practice known as pool-based active learning or selective sampling.

There has been a great deal of research in active learning. For example, Tong and Koller (2000) select queries to minimize the version space size for support vector machines; Cohn et al. (1996) minimize the variance component of the estimated generalization error; Freund et al. (1997) employ a committee of classifiers, and query a point whenever the committee members disagree. Most of the active learning methods do not take further advantage of the large amount of unlabeled data once the queries are selected. Exceptions include McCallum and Nigam (1998) who use EM with unlabeled data integrated into the active learning, and Muslea et al. (2002) who use a semi-supervised learning method during training. In addition to this body of work from the machine learning community, there is a large literature on the closely related topic of experimental design in statistics; Chaloner and Verdinelli (1995) give a survey of experimental design from a Bayesian perspective.

Recently Zhu et al. (2003) introduced a semi-supervised learning framework which is based on Gaussian random fields and harmonic functions. In this paper we demonstrate how this framework allows a combination of active learning and semi-supervised learning. In brief, the framework allows one to efficiently estimate the expected generalization error after querying a point, which leads to a better query selection criterion than naively selecting the point with maximum label ambiguity. Then, once the queries are selected and added to the labeled data set, the classifier can be trained using both the labeled and remaining unlabeled data. We present results on synthetic data, text classifica-

Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington DC, 2003.

tion and image classification tasks that indicate the combination of these techniques can result in highly effective learning schemes.

2. Gaussian random fields and harmonic energy minimizing functions

We begin by briefly describing the semi-supervised learning framework of Zhu et al. (2003). There are l labeled points $(x_1, y_1), \ldots, (x_l, y_l)$, and u unlabeled points $x_{l+1}, \ldots, x_{l+u}$; usually $l \ll u$. We will use L, U to denote the labeled and unlabeled set, and $n = l + u$ the total number of points. We assume the labels are binary: $y_L \in \{0, 1\}$. We assume a connected graph $G = (V, E)$ is given with nodes V corresponding to the n data points, of which the nodes in L are labeled with the corresponding y's. The edges E are represented by an $n \times n$ weight matrix W which is given. For example if $x \in \mathbb{R}^m$, W can be the radial basis function (RBF):

$$w_{ij} = \exp\left(-\frac{1}{\sigma^2}\sum_{d=1}^{m}(x_{id} - x_{jd})^2\right) \tag{1}$$

so that nearby points in Euclidean space are assigned large edge weights. Other weightings are possible, of course, and may be more appropriate when x is discrete or symbolic. For our purposes the matrix W fully specifies the data manifold structure. We note that a method for learning the scale parameter σ is proposed in (Zhu et al., 2003).

The semi-supervised algorithm in this paper is based on a relaxation of the requirement that labels are binary, resulting in a simple and tractable family of algorithms. We allow continuous labels on unlabeled nodes $y_U : V \to \mathbb{R}$. The labels on labeled data are still constrained to be 0 or 1: $y(i) = y_L(i) \in \{0, 1\}$ for $i = 1, \ldots, l$. We also denote the constraint by $y|_{L=y_L}$. Since we want unlabeled points that are nearby in the graph to have similar labels, we define the *energy* to be

$$E(y) = \frac{1}{2}\sum_{i,j} w_{ij}\left(y(i) - y(j)\right)^2 \tag{2}$$

so that low energy corresponds to a slowly varying function over the graph. Define the diagonal matrix $D = diag(d_i)$ whose entries $d_i = \sum_j w_{ij}$ are the weighted degrees of the nodes in G. The *combinatorial Laplacian* is the $n \times n$ matrix $\Delta = D - W$. We can rewrite the energy function in matrix form as $E(y) = y^\top \Delta y$. We consider the *Gaussian random field*

$$p(y) = \frac{1}{Z_\beta}\exp\left(-\beta E(y)\right)$$

where β is an "inverse temperature" parameter, and Z_β is the partition function $Z_\beta = \int_{y|_{L=y_L}} \exp\left(-\beta E(y)\right) dy$. The Gaussian random field differs from the "standard" discrete Markov random field in that the field configurations are over a continuous state space. Moreover, for a Gaussian field the joint probability distribution over unlabeled nodes is Gaussian with covariance matrix $\frac{1}{\beta}\Delta_{uu}^{-1}$. Δ_{uu} is the submatrix of Δ corresponding to unlabeled data.

The minimum energy function $f = \arg\min_{y|_{L=y_L}} E(y)$ of the Gaussian field is *harmonic*; that is, it satisfies $\Delta f = 0$ on unlabeled data points U, and is equal to y_L on the labeled data points L. The harmonic property means that the value at each unlabeled node is the average of neighboring nodes:

$$f(j) = \frac{1}{d_j}\sum_{i \sim j} w_{ij} f(i), \text{ for } j = l+1, \ldots, l+u$$

which is consistent with our prior notion of smoothness with respect to the graph. Because of the maximum principle of harmonic functions (Doyle & Snell, 1984), f is unique. Furthermore, f satisfies $0 < f(j) < 1$ for $j \in U$ when U is connected and labeled nodes from both classes are present at the boundary (the usual case; otherwise f takes on the extremum 0 or 1). By definition f is the *mode* of the Gaussian random field, but since the joint distribution is Gaussian, f is also the *mean* of the field.

The harmonic energy minimizing function f can be computed with matrix methods. We partition the Laplacian matrix Δ into blocks for labeled and unlabeled nodes,

$$\Delta = \begin{bmatrix} \Delta_{ll} & \Delta_{lu} \\ \Delta_{ul} & \Delta_{uu} \end{bmatrix}$$

and let $f = \begin{bmatrix} f_l \\ f_u \end{bmatrix}$ where $f_l = y_L$, and f_u denotes the mean values on the unlabeled data points. The solution is given by

$$f_u = -\Delta_{uu}^{-1}\Delta_{ul} f_l \tag{3}$$

It is not hard to show that the Gaussian field, conditioned on labeled data, is a multivariate normal: $y_u \sim \mathcal{N}(f_u, \Delta_{uu}^{-1})$. To carry out classification with a Gaussian field, we note that the harmonic energy minimizing function f is the mean of the field. Therefore the Bayes classification rule is to label node i as class 1 in case $f(i) > 0.5$, and to label node i class 0 otherwise.

The harmonic function f has many nice interpretations, of which the random walk view is particularly relevant here. Define the transition matrix $P = D^{-1}W$. Consider the random walk on graph G by a particle. Starting from an unlabeled node i, it moves to a node j with probability P_{ij} after one step. The walk continues until the particle reaches a labeled node. Then f_i is the probability that the particle, starting from node i, reaches a labeled node with label 1.

The labeled data are the absorbing boundary for the random walk. More on this semi-supervised learning framework can be found in (Zhu et al., 2003).

3. Active learning

We propose to perform active learning with the Gaussian random field model by greedily selecting queries from the unlabeled data to minimize the risk of the harmonic energy minimization function. The risk is the estimated generalization error of the Bayes classifier, and can be efficiently computed with matrix methods.

We define the *true* risk $\mathcal{R}(f)$ of the Bayes classifier based on the harmonic function f to be

$$\mathcal{R}(f) \quad = \quad \sum_{i=1}^{n} \sum_{y_i=0,1} [\mathrm{sgn}(f_i) \neq y_i] \; p^*(y_i|L)$$

where $\mathrm{sgn}(f_i)$ is the Bayes decision rule, where (with a slight abuse of notation) $\mathrm{sgn}(f_i) = 1$ if $f_i > 0.5$ and $\mathrm{sgn}(f_i) = 0$ otherwise. Here $p^*(y_i|L)$ is the unknown true label distribution at node i, given the labeled data. Because of this $\mathcal{R}(f)$ is not computable. In order to proceed, it is necessary to make assumptions. We begin by assuming that we can estimate the unknown distribution $p^*(y_i|L)$ with the mean of the Gaussian field model:

$$p^*(y_i = 1|L) \approx f_i$$

Intuitively, recalling f_i is the probability of reaching 1 in a random walk on the graph, our assumption is that we can approximate the distribution using a biased coin at each node, whose probability of heads is f_i. With this assumption we can compute the *estimated risk* $\widehat{\mathcal{R}}(f)$ as

$$\begin{aligned} \widehat{\mathcal{R}}(f) &= \sum_{i=1}^{n} [\mathrm{sgn}(f_i) \neq 0]\,(1-f_i) + [\mathrm{sgn}(f_i) \neq 1]\, f_i \\ &= \sum_{i=1}^{n} \min(f_i, 1-f_i) \end{aligned} \tag{4}$$

If we perform active learning and query an unlabeled node k, we will receive an answer y_k (0 or 1). Adding this point to the training set and retraining, the Gaussian field and its mean function will of course change. We denote the new harmonic function by $f^{+(x_k,y_k)}$. The estimated risk will also change:

$$\widehat{\mathcal{R}}(f^{+(x_k,y_k)}) \quad = \quad \sum_{i=1}^{n} \min(f_i^{+(x_k,y_k)}, 1 - f_i^{+(x_k,y_k)})$$

Since we do not know what answer y_k we will receive, we again assume the answer is approximated with $f_k \approx p^*(y_k = 1|L)$. The *expected* estimated risk after querying node k is therefore

$$\widehat{\mathcal{R}}(f^{+x_k}) \quad = \quad (1-f_k)\,\widehat{\mathcal{R}}(f^{+(x_k,0)}) + f_k\,\widehat{\mathcal{R}}(f^{+(x_k,1)})$$

Input: L, U, weight matrix W
While more labeled data required:
  Compute harmonic f using (3)
  Find best query k using (5)
  Query point x_k, receive answer y_k
  Add (x_k, y_k) to L, remove x_k from U
end
Output: L and classifier f.

Figure 1. The active learning algorithm

The active learning criterion we use in this paper is the greedy procedure of choosing the next query k that minimizes the expected estimated risk:

$$k \quad = \quad \arg\min_{k'} \widehat{\mathcal{R}}(f^{+x_{k'}}) \tag{5}$$

To carry out this procedure, we need to compute the harmonic function $f^{+(x_k,y_k)}$ after adding (x_k, y_k) to the current labeled training set. This is the retraining problem and is computationally intensive in general. However for Gaussian fields and harmonic functions, there is an efficient way to retrain. Recall that the harmonic function solution is

$$f_u = -\Delta_{uu}^{-1} \Delta_{ul} f_l$$

What is the solution if we fix the value y_k for node k? This is the same as finding the conditional distribution of all unlabeled nodes, given the value of y_k. Noting that $y_u \sim \mathcal{N}(f_u, \Delta_{uu}^{-1})$, a multivariate normal distribution, a standard result (a derivation is given in Appendix A) gives the conditional once we fix y_k:

$$f_u^{+(x_k,y_k)} \quad = \quad f_u + (y_k - f_k) \frac{(\Delta_{uu}^{-1})_{\cdot k}}{(\Delta_{uu}^{-1})_{kk}}$$

where $(\Delta_{uu}^{-1})_{\cdot k}$ is the k-th column of the inverse Laplacian on unlabeled data, and $(\Delta_{uu}^{-1})_{kk}$ is the k-th diagonal element of the same matrix. Both are already computed when we compute the harmonic function f. This is a linear computation and therefore can be carried out efficiently.

To summarize, the active learning algorithm is shown in Figure 1. The time complexity to find the best query is $O(n^2)$. As a final word on computational efficiency, we note that after adding query x_k and its answer to L, in the next iteration we will need to compute $((\Delta_{uu})_{\neg k})^{-1}$, the inverse of the Laplacian on unlabeled data, with the row/column for x_k removed. Instead of naively taking the inverse, there are efficient algorithms to compute it from $(\Delta_{uu})^{-1}$; a derivation is given in Appendix B.

4. Experiments

Figure 2 shows (top left) a synthetic dataset with two labeled data (marked '1', '0'), an unlabeled point 'a' in the

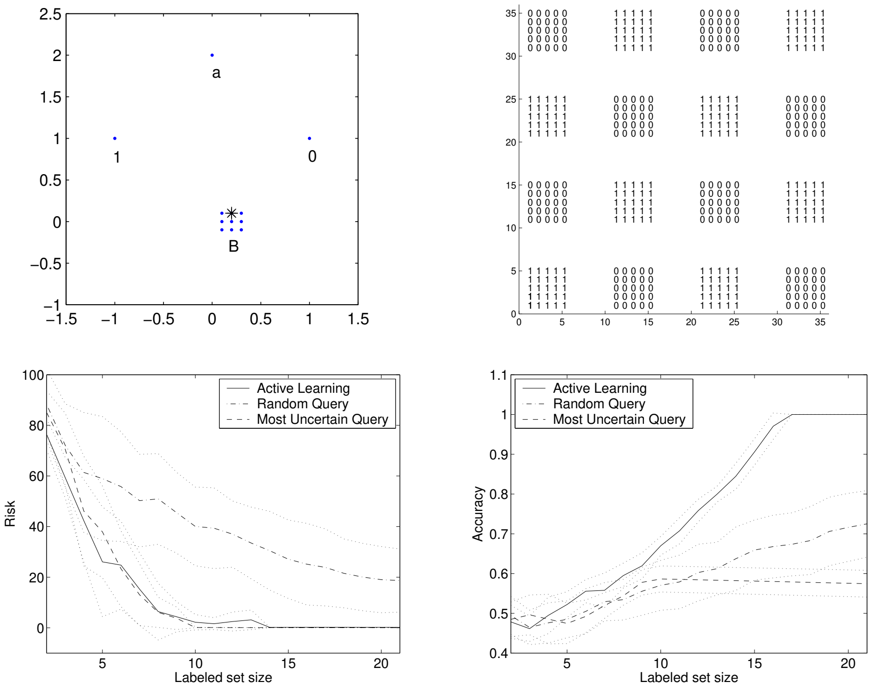

Figure 2. Synthetic data experiments. Top left: synthetic data 1; top right: synthetic data 2. Bottom left: risk on synthetic data 2; bottom right: classification accuracy on synthetic data 2. Standard errors are shown as dotted lines.

center above and a cluster of 9 unlabeled points 'B' below. 'B' is slighted shifted to the right. The graph is fully connected with weight $w_{ij} = \exp(-d_{ij}^2)$, where d_{ij} is the Euclidean distance between i, j. In this configuration, we have the most uncertainty in 'a': $f(a) = 0.43$. Points in 'B' have f around 0.32, so we are more certain about the labels of 'B' than 'a'. However, the risk minimization criterion picks the upper center point (marked with a star) in 'B' to query, instead of 'a'. In fact the estimated risk is $\widehat{\mathcal{R}}(a) = 2.9$, and $\widehat{\mathcal{R}}(b \in B) \approx 1.1$. This shows that the active learning algorithm is not simply picking the most uncertain point for query, but rather it "thinks globally."

Figure 2 also shows (top right) another synthetic dataset, where the true labels for the 400 points form a chess-board pattern. We expect active learning to discover the pattern and query a small number of representatives from each cluster. On the other hand, we expect a much larger number of queries if queries are randomly selected. We use a fully connected graph with weight $w_{ij} = \exp(-d_{ij}^2/4)$. We perform 20 random trials. At the beginning of each trial we randomly select a positive example and a negative example as the initial training set. We then run active learning and compare it to two baselines: (1) "Random Query": randomly selecting the next query from U; (2) "Most Uncertain Query": selecting the most uncertain instance in U, i.e. the one with f closest to 0.5. In each case we run for 20 iterations (queries). At each iteration we plot the estimated risk (4) of the selected query (lower left), and the classification accuracy on U (lower right). The error bars are ± 1 standard deviation, averaged over the random trials. As expected, with active learning we reduce risk more quickly than random queries or the most uncertain queries. In fact, active learning with about 15 queries (plus 2 initial random points) learns the correct concept, which is nearly optimal given that there are 16 clusters. Looking at the queries, we find that active learning mostly selects the central points of the clusters.

Next we report the results of document categorization experiments using the 20 newsgroups dataset[1]. We evaluate the following binary classification tasks: rec.sport.baseball (994 documents) vs. rec.sport.hockey (999); comp.sys.-ibm.pc.hardware (982) vs. comp.sys.mac.hardware (961); talk.religion.misc (628) vs. alt.atheism (799). They represent easy, moderate and hard problems respectively. Each document is minimally processed into a "tf.idf" vector, without applying header removal, frequency cutoff, stemming, or a stopword list. Two documents u, v are connected by an edge if u is among v's 10 nearest neighbors or if v is among u's 10 nearest neighbors, as measured by their cosine similarity $cs_{uv} = \frac{u^\top v}{|u||v|}$. We use the following weight function on the edges: $w_{uv} = \exp\left(-\left(1 - cs_{uv}\right)/0.03\right)$ As before, we perform 20 trials, and randomly pick two

[1] http://www.ai.mit.edu/people/jrennie/20Newsgroups/

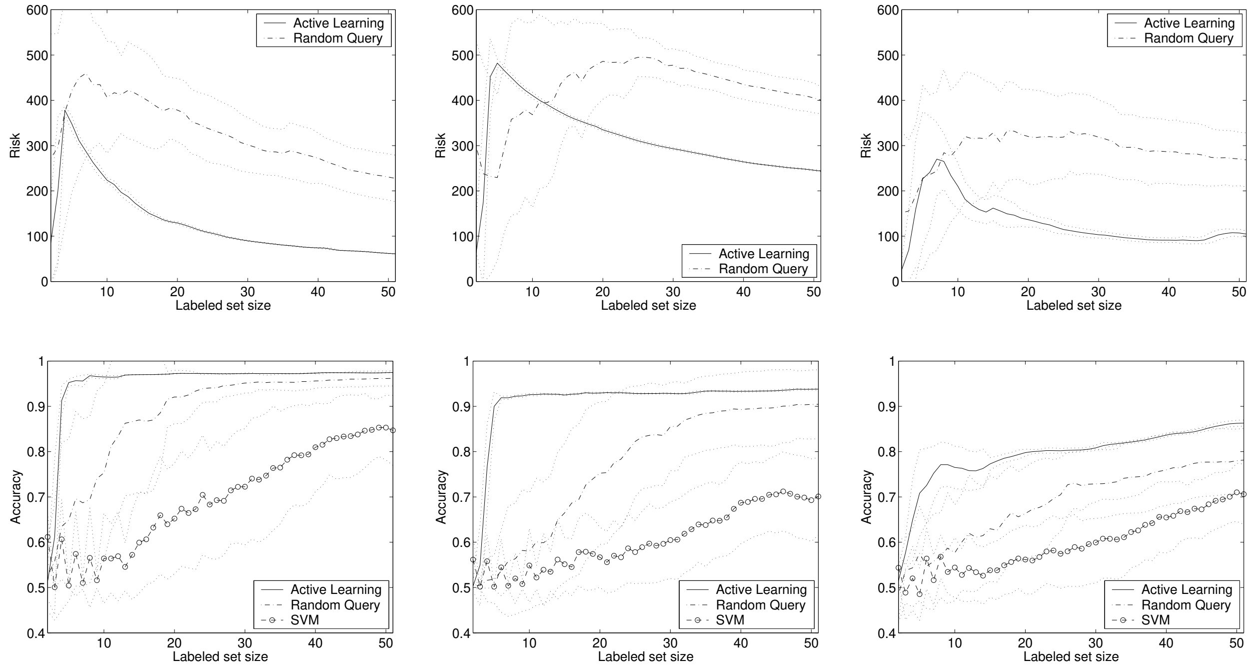

Figure 3. Risk (top) and classification accuracy (bottom) on 20 newsgroups, *compared to random queries*; Baseball vs. Hockey (left), PC vs. MAC (center), and Religion vs. Atheism (right).

initial training examples to start each trial. The rest of the documents are treated as unlabeled data. In each trial we answer 50 queries. Again we compare active learning with two baselines:

1. Figure 3 compares it with random queries. Active learning reduces risk faster, and achieves high classification accuracy more rapidly than random queries. For easy datasets only a handful of queries are necessary for active learning to achieve 90% accuracy. We observe that active learning tends to query the same small set of documents in different trials. We also trained a SVM classifier on the random queries, with the cosine similarity kernel and $C = 5$ (which was the best setting among several different kernels and a wide range of C values). Note that the SVM does not utilize unlabeled data; on these datasets the semi-supervised method outperforms the SVM.

2. Figure 4 compares it with the most uncertain queries. "Most Uncertain Query" selects the instance whose f value is closest to 0.5 as the next query. "SVM Most Uncertain" selects the instance whose margin is closest to 0 (closest to the decision boundary). As before the SVM does not utilize unlabeled data. The harmonic function classification is worse with the most uncertain queries than with random queries, while the SVM improves with the most uncertain queries.

In all cases our proposed active learning scheme clearly outperforms both baselines for the 20 newsgroups datasets.

We then evaluate active learning on a handwritten digits dataset, originally from the Cedar Buffalo binary digits database (Hull, 1994). The digits were preprocessed to reduce the size of each image down to a 16×16 grid by down-sampling and Gaussian smoothing, with pixel values ranging from 0 to 255 (Le Cun et al., 1990). They are further scaled down to 8×8 by averaging 2×2 pixel bins. Each image is thus represented by a 64-dimensional vector. We consider the binary problem of classifying digits "1" vs. "2," with 1100 images in each class. We create a graph on the 2200 images, with an edge between images i, j iff i is in j's 10 nearest neighbors (in Euclidean distance) or vice versa. The weights on edges are $w_{ij} = \exp\left(-d_{ij}^2/380^2\right)$ where d_{ij} is the pixel-wise Euclidean distance between images i, j. In each of 20 trials we randomly pick one example from each class to form the initial training set. We then compare active learning with random queries and the most uncertain queries for 50 iterations. Figure 5 (left) shows the risk. With active learning the risk decreases faster than with the baselines. Figure 5 (center) shows the classification accuracy where active learning is seen to outperform the baselines. Again only a handful of examples are needed for active learning to reach high accuracy. The SVM uses kernel $(1 + x_i^\top x_j)^2$ and $C = 1$, which is one of the best among polynomial kernels with order 1 to 5, and a wide range of C values. We also observe that there are certain

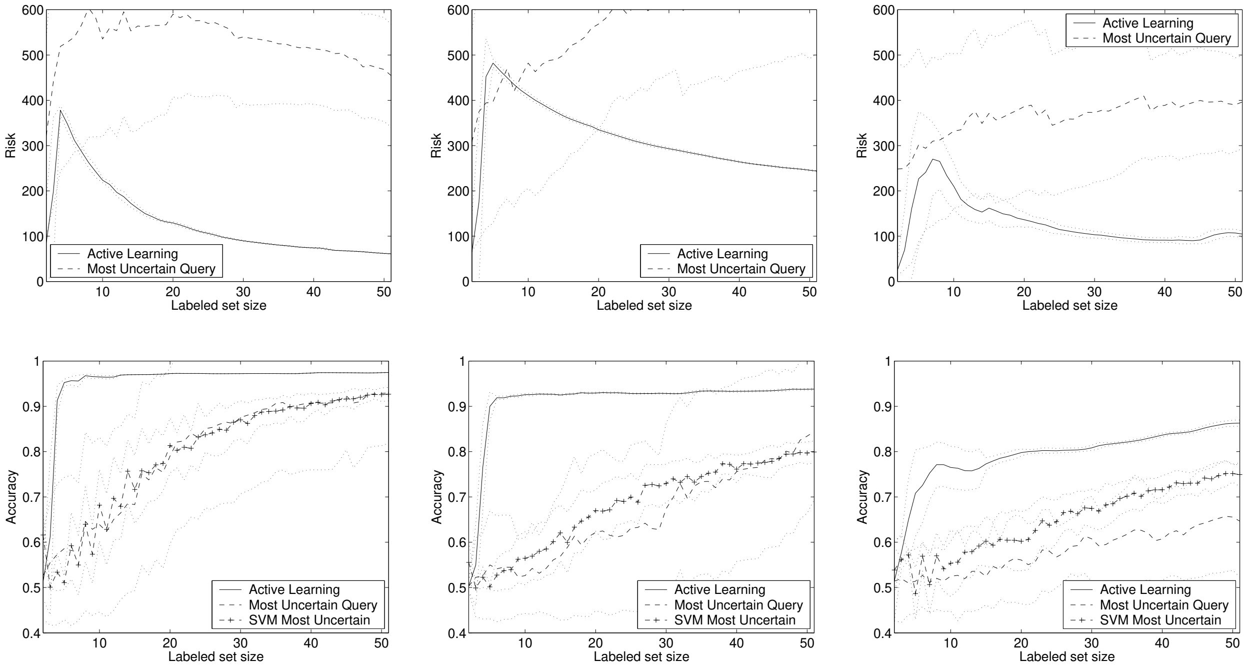

Figure 4. Risk (top) and classification accuracy (bottom) on 20 newsgroups, *compared to the most uncertain queries*; Baseball vs. Hockey (left), PC vs. MAC (center), and Religion vs. Atheism (right).

images that active learning frequently queries in different trials. Figure 5 (right) shows some of the most frequently queried images; we believe these images are representative of the variations in the dataset.

We also evaluate on the difficult binary problem of classifying odd digits vs. even digits. That is, we group "1,3,5,7,9" and "2,4,6,8,0" into two classes. There are 400 images per digit (2000 per class). It is a difficult dataset because the target concept is rather artificial; on the other hand this dataset resembles synthetic data 2 (Figure 2) where each class has several internal clusters. The experimental setup is the same as above except that we run for 100 iterations. Figure 6 shows the results. Again active learning is superior than the baselines. We also see that odd vs. even is a harder concept which takes active learning about 50 queries to approximately learn. The digits shown are the 25 most frequently queried instances in the first 30 iterations across all trials. With 4000 instances this dataset is also the largest and slowest. With a naive Matlab implementation, the calculations take roughly 50 seconds per iteration on a 1GHz Linux machine. In contrast, the "1" vs. "2" dataset requires five seconds per iteration, while all of the 20 newsgroups datasets take less than two seconds per iteration.

We finally note that if we train an SVM on the active queries chosen from the harmonic function risk minimization procedure, the accuracy is always worse than our proposed active learning method, and often even worse than the SVM on random queries.

5. Summary

We have proposed an approach to active learning which is tightly coupled with semi-supervised learning using Gaussian fields and harmonic functions. The algorithm selects queries to minimize an approximation to the expected generalization error. Experiments on text categorization and handwritten digit recognition indicate that the active learning algorithm can be highly effective.

References

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*, 237–304.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Doyle, P., & Snell, J. (1984). *Random walks and electric networks.* Mathematical Assoc. of America.

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

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Howard, W., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems, 2*.

McCallum, A. K., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. *Proceedings of ICML-98, 15th International Conference on Machine Learning* (pp. 350–358). Madison, US: Morgan Kaufmann Publishers, San Francisco, US.

Muslea, I., Minton, S., & Knoblock, C. (2002). Active + semi-supervised learning = robust multi-view learning. *Proceedings of ICML-02, 19th International Conference on Machine Learning* (pp. 435–442).

Seeger, M. (2001). *Learning with labeled and unlabeled data* (Technical Report). University of Edinburgh.

Tong, S., & Koller, D. (2000). Support vector machine active learning with applications to text classification. *Proceedings of ICML-00, 17th International Conference on Machine Learning* (pp. 999–1006). Stanford, US: Morgan Kaufmann Publishers, San Francisco, US.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *ICML-03, 20th International Conference on Machine Learning*.

Appendix A: The harmonic function after knowing one more label

Construct the graph as usual. The random walk solution is $f_u = -\Delta_{uu}^{-1}\Delta_{ul} f_l = \Delta_{uu}^{-1} W_{ul} f_l$. There are u unlabeled nodes. We ask the question: what is the solution if we add a node with value f_0 to the graph, and connect the new node to unlabeled node i with weight w_0? The new node is a "dongle" attached to node i. Besides the usage here, dongle nodes can be useful for handling noisy labels where one would put the observed labels on the dongles, and infer the hidden true labels for the nodes attached to dongles. Note that when $w_0 \to \infty$, we effectively assign label f_0 to node i.

Since the dongle is a labeled node in the augmented graph,

$$\begin{aligned} f_u^+ &= {\Delta_{uu}^+}^{-1} W_{ul}^+ f_l^+ = (D_{uu}^+ - W_{uu})^{-1} W_{ul}^+ f_l^+ \\ &= (w_0 e e^\top + D_{uu} - W_{uu})^{-1}(w_0 f_0 e + W_{ul} f_l) \\ &= (w_0 e e^\top + \Delta_{uu})^{-1}(w_0 f_0 e + W_{ul} f_l) \end{aligned}$$

where e is a column vector of length u with 1 in position i and 0 elsewhere. Note that we can use the matrix inversion lemma here, to obtain

$$\begin{aligned} (w_0 e e^\top + \Delta_{uu})^{-1} &= \Delta_{uu}^{-1} - \frac{\Delta_{uu}^{-1}(\sqrt{w_0}e)(\sqrt{w_0}e)^\top \Delta_{uu}^{-1}}{1 + (\sqrt{w_0}e)^\top \Delta_{uu}^{-1}(\sqrt{w_0}e)} \\ &= G - \frac{1}{1 + w_0 G_{ii}} w_0 G_{|i} G \end{aligned}$$

where we use the shorthand $G = \Delta_{uu}^{-1}$ (the Green's function); G_{ii} is the i-th row, i-th column element in G; $G_{|i}$ is a square matrix with G's i-th column and 0 elsewhere. Some calculation gives

$$f_u^+ = f_u + \frac{w_0 f_0 - w_0 f_i}{1 + w_0 G_{ii}} G_{\cdot i}$$

where f_i is the unlabeled node's original solution, and $G_{\cdot i}$ is the i-th column vector in G. If we want to pin down the unlabeled node to value f_0, we can let $w_0 \to \infty$ to obtain

$$f_u^+ = f_u + \frac{f_0 - f_i}{G_{ii}} G_{\cdot i}$$

Appendix B: The inverse of a matrix with one row/column removed

Let A be an $n \times n$ non-singular matrix. Given A^{-1}, we would like a fast algorithm to compute $A_{\neg i}^{-1}$, where $A_{\neg i}$ is the $(n-1) \times (n-1)$ matrix obtained by removing the i-th row and column from A.

Let $B = \text{perm}(A, i)$ be the matrix created by moving the i-th row in front of the 1st row, and the i-th column in front of the 1st column of A. Then

$$A_{\neg i}^{-1} = (\text{perm}(A,i)_{\neg 1})^{-1} = (B_{\neg 1})^{-1}$$

Also note $B^{-1} = \text{perm}(A^{-1}, i)$. So we only need to consider the special case of removing the first row/column of a matrix. Write B out as $B = \begin{bmatrix} b_{11} & B_{1*} \\ B_{*1} & B_{\neg 1} \end{bmatrix}$, where $B_{1*} = (b_{12} \dots b_{1n})$ and $B_{*1} = (b_{21} \dots b_{n1})^\top$. We will transform B into a block diagonal form in two steps. First, let $B' = \begin{bmatrix} 1 & 0 \\ B_{*1} & B_{\neg 1} \end{bmatrix} = B + uv^\top$ where $u = (-1, 0, \dots, 0)^\top$ and $v = (b_{11} - 1, B_{1*})^\top$. We are interested in $(B')^{-1}$ which will be used in the next step. By the matrix inversion lemma (Sherman-Morrison-Woodbury formula),

$$(B')^{-1} = (B + uv^\top)^{-1} = B^{-1} - \frac{B^{-1}uv^\top B^{-1}}{1 + v^\top B^{-1} u}$$

Next let $B'' = \begin{bmatrix} 1 & 0 \\ 0 & B_{\neg 1} \end{bmatrix} = B' + wu^\top$ where $w = (0, B_{*1})^\top$. Applying the matrix inversion lemma again,

$$(B'')^{-1} = (B' + wu^\top)^{-1} = (B')^{-1} - \frac{(B')^{-1} wu^\top (B')^{-1}}{1 + u^\top (B')^{-1} w}$$

But since B'' is block diagonal, we know $(B'')^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & (B_{\neg 1})^{-1} \end{bmatrix}$. Therefore $(B_{\neg 1})^{-1} = ((B'')^{-1})_{\neg 1}$.

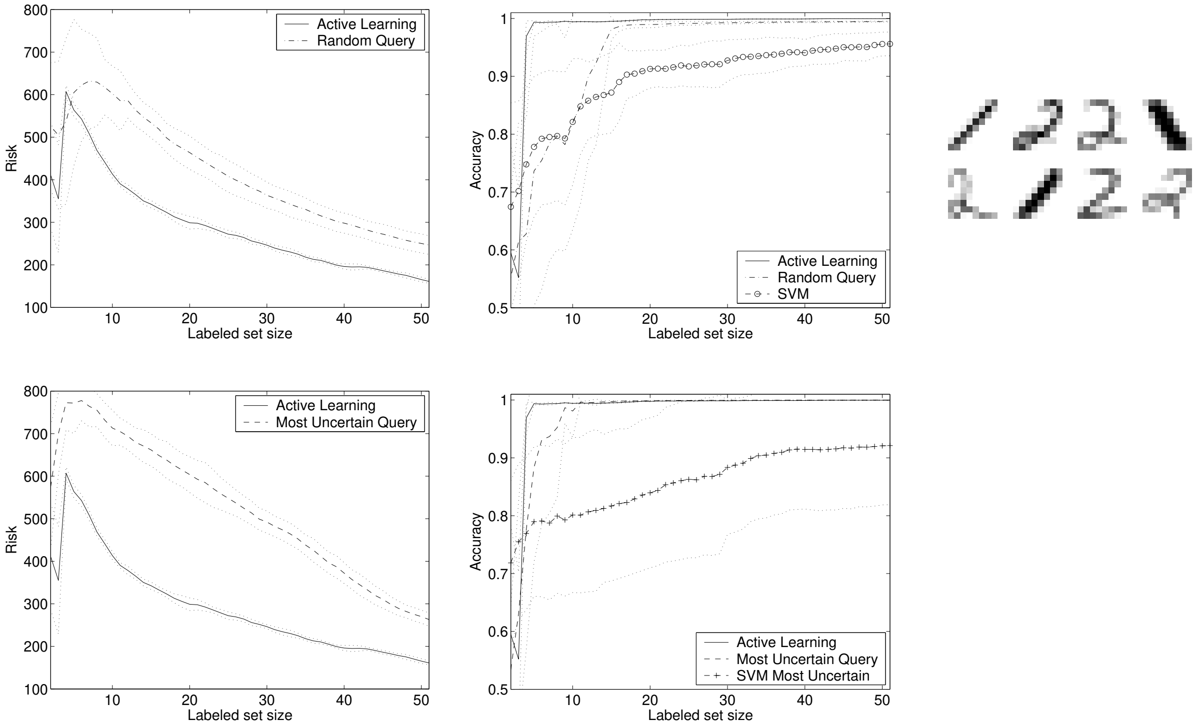

Figure 5. Handwritten digits "1" vs. "2", compared with random queries (top) and the most uncertain queries (bottom); Risk (left), classification accuracy (center) and frequently queried images (right).

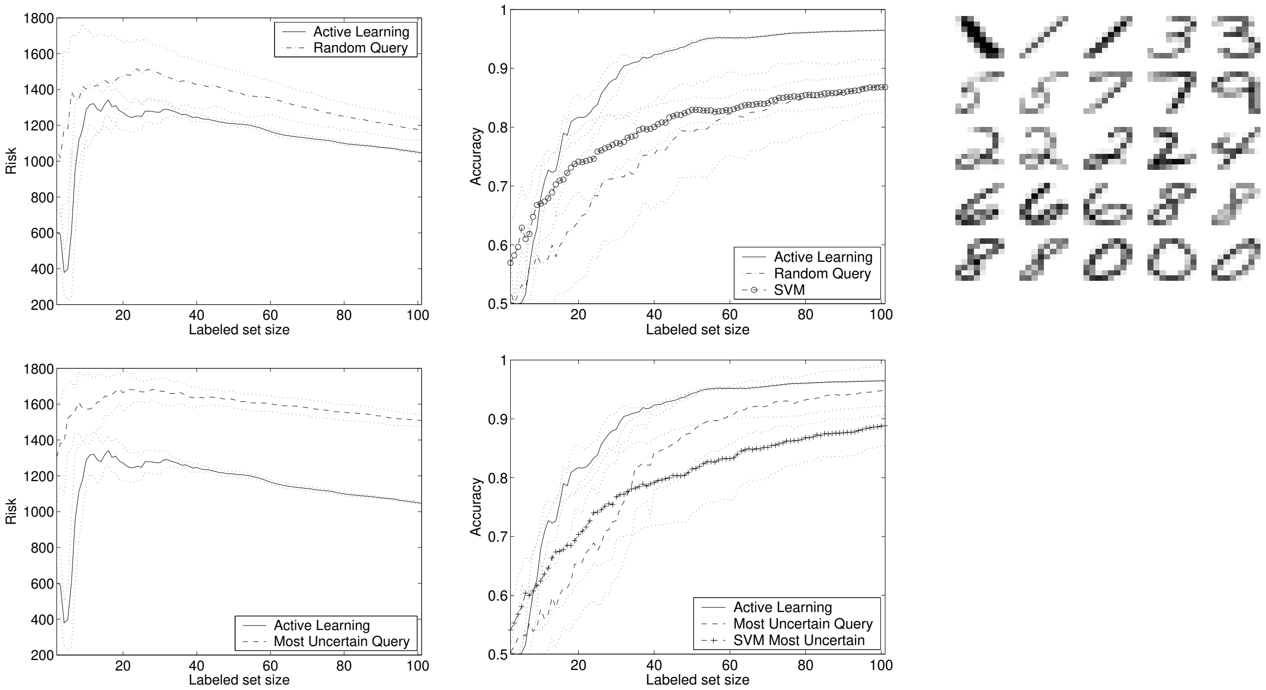

Figure 6. Handwritten digits odd vs. even, compared with random queries (top) and the most uncertain queries (bottom); Risk (left), classification accuracy (center) and frequently queried images (right).