Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning

Xiaojin Zhu ZHUXJ@CS.CMU.EDU
John Lafferty LAFFERTY@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract

Graph-based methods for semi-supervised learning have recently been shown to be promising for combining labeled and unlabeled data in classification problems. However, inference for graph-based methods often does not scale well to very large data sets, since it requires inversion of a large matrix or solution of a large linear program. Moreover, such approaches are inherently transductive, giving predictions for only those points in the unlabeled set, and not for an arbitrary test point. In this paper a new approach is presented that preserves the strengths of graph-based semi-supervised learning while overcoming the limitations of scalability and non-inductive inference, through a combination of generative mixture models and discriminative regularization using the graph Laplacian. Experimental results show that this approach preserves the accuracy of purely graph-based transductive methods when the data has "manifold structure," and at the same time achieves inductive learning with significantly reduced computational cost.

1. Introduction

The availability of large data collections, with only limited human annotation, has turned the attention of a growing community of machine learning researchers to the problem of semi-supervised learning. The broad research agenda of semi-supervised learning is to develop methods that can leverage a large amount of unlabeled data to build more accurate classification algorithms than can be achieved using purely supervised learning. An attractive new family of semi-supervised methods is based on the use of a graphical representation of the unlabeled data—examples of this paradigm include the work of Blum and Chawla (2001); Zhu et al. (2003); Zhou et al. (2004); Belkin et al. (2004a).

Many graph-based methods are inherently *transductive* in nature: a graph is formed with vertices representing the labeled and unlabeled data, and a graph algorithm is used to somehow separate the nodes according to the predicted class labels. However, when a new data point is presented, it is unclear how to make a prediction—other than to rebuild the graph with the new test point and rerun the graph algorithm from scratch. Since this may involve solving a large linear program or inverting a huge matrix, these procedures have limited generalization ability. Yet semi-supervised methods should be most attractive when the unlabeled data set is extremely large, and thus scalability becomes a central issue. In this paper we address the problems of scalability and non-inductive inference by combining parametric mixture models with graph-based methods. Our approach is related to, but different from, the recent work of Delalleau et al. (2005) and Belkin et al. (2004b).

The mixture model has long been recognized as a natural approach to modeling unannotated data; indeed, some of the earliest studies of the semi-supervised learning problem investigated the statistical or learning-theoretic efficiency of estimating mixture models through a combination of labeled and unlabeled data (Castelli & Cover, 1996; Ratsaby & Venkatesh, 1995). As a generative model, a mixture is naturally inductive, and typically has a relatively small number of parameters. Various applied studies suggested that multinomial mixtures can be effective at using unlabeled data for classifying text documents (Nigam et al., 2000), where the learning is typically carried out using the EM algorithm to estimate the MAP model over the unlabeled set. However, the anecdotal evidence is that many more studies were not published because they obtained *negative* results, showing that learning a mixture model will often degrade the performance of a model fit using only the labeled data; one published study with these conclusions is (Cozman et al., 2003). One of the reasons for this phenomenon is that the data may have a "manifold structure" that is incompatible with the generative mixture

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

model; thus EM may have difficulty in making the *labels* follow the manifold. An illustrative example is given in the left plot of Figure 1. The desired behavior is shown in the right plot, which is achieved using the *harmonic mixture* model presented in this paper.

Mixture models and graph-based semi-supervised learning methods make different assumptions about the relation between the data and the labels—but these assumptions are not mutually exclusive. It is possible that the data fits the component model (a Gaussian, for example) *locally*, while the manifold structure appears *globally*. The present work attempts to combine the strengths of both approaches. We show how the mixture model can be combined with the graph to yield a much smaller "backbone graph" with nodes induced from the mixture components. The number of mixture components controls the size of the backbone graph, leading to computationally efficient algorithms. The harmonic mixture is a special case of our general framework, where a harmonic function (Zhu et al., 2003) is induced over the backbone graph to specify the class membership of the mixture model. Since the mixture model is generative it handles new points, while the graph allows the labels to follow the data manifold. Importantly, our procedure for combining the mixture model with the backbone graph involves a convex optimization problem.

After a brief overview of mixture models and previous work on graph-based semi-supervised learning in Section 2, we detail our combined approach in Section 3. Experimental results for synthetic data, handwritten digits recognition, image analysis and text categorization are given in Section 4. The paper concludes with a discussion and summary of the results.

2. Background and Notation

Let $(\mathbf{x}_L, \mathbf{y}_L) = \{(x_1, y_1) \ldots (x_l, y_l)\}$ be the labeled data. For simplicity we consider binary classification, with $y \in \{-1, 1\}$. Let $\mathbf{x}_U = \{x_{l+1} \ldots x_n\}$ be the unlabeled data, and $u = n - l$. The letters L and U will be used to represent the labeled and unlabeled data, respectively. In semi-supervised learning the goal is to learn a classifier from both $(\mathbf{x}_L, \mathbf{y}_L)$ and $\mathbf{x}_U$.

2.1. Mixture models

In the standard view of a mixture model for classification, the generative process is to sample $m \sim \text{Mult}(\lambda_y)$ from a multinomial depending on the class y, and to then sample $x \sim G(\eta_m)$ for some generative model G. We will work with a different, but equivalent view where a mixture component $m \sim \text{Mult}(\gamma)$ is first sampled from a multinomial model γ over M outcomes, where M is the number of mixture components. Then, the label $y \sim \text{Mult}(\lambda_m)$ and features $x \sim G(\eta_m)$ are generated (conditionally independently) given m. Note that $p(y \,|\, m)$ can take 'soft' values between 0 and 1, enabling classes to share a mixture component. In unlabeled data, both the mixing component m and class label y is latent for each example. The parameters of the mixture model are $\theta = \{(\gamma_m, \lambda_m, \eta_m)\}_{m=1}^M$. The EM algorithm is the standard procedure for estimating the parameters to maximize the incomplete likelihood $\mathcal{L}(\theta) = \prod_i \sum_{m,y} p(x_i \,|\, \eta_m)\, p(y \,|\, \lambda_m)\, \gamma_m$. Combining the labeled and unlabeled data together, the log likelihood is

$$\begin{aligned} \ell(\theta) \;\; = \;\; & \sum_{i \in L} \log \sum_m \gamma_m \, \lambda_{my_i} \, p(x_i \,|\, \eta_m) + \\ & \sum_{i \in U} \log \sum_m \gamma_m \, p(x_i \,|\, \eta_m) \end{aligned} \qquad (1)$$

2.2. Label smoothness on the data graph

Graph-based semi-supervised learning methods are based on the principle that the label probability should vary smoothly over the data graph. The graph has n nodes, with two nodes connected by a (weighted) edge if they are deemed similar according to some similarity function, chosen by prior knowledge. The graph is thus represented by the $n \times n$ symmetric weight matrix W; the combinatorial Laplacian is $\Delta = D - W$, where the diagonal degree matrix satisfies $D_{ii} = \sum_j w_{ij}$.

Label smoothness can be expressed in different ways. We adopt the *energy* used in semi-supervised learning by Zhu et al. (2003), given by

$$\mathcal{E}(\boldsymbol{f}) \;\; = \;\; \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \, (f_i - f_j)^2 = \boldsymbol{f}^\top \Delta \boldsymbol{f} \qquad (2)$$

where $\boldsymbol{f}$ is the label posterior vector of a mixture model, defined as

$$f_i = \begin{cases} \delta(y_i, 1) & i \in L \\ p(y_i = 1 \,|\, x_i, \theta) & i \in U \end{cases}$$

That is, f_i is the probability that point i has label $y = 1$ under the mixture model θ; since this is a function of the parameters of the mixture, we will write it also as $\mathcal{E}(\theta)$. The energy is small when $\boldsymbol{f}$ varies smoothly over the graph. Zhu et al. (2003) proposed the use of the harmonic solution $\Delta f = 0$ subject to the constraints specified by the labeled data. Since Δ is the combinatorial Laplacian, this implies that for $i \in U$, f_i is the average of the values of f at neighboring nodes (the *harmonic* property), which is consistent with the smoothness assumption. Alternative measures of smoothness are based on the normalized Laplacian (Zhou et al., 2004) or spectral transforms (Zhu et al., 2005), and work similarly in the framework below.

Note that by extending the notion of a weighted graph to allow self-transitions and possibly negative edge weights, any symmetric matrix with zero row sums can be considered to be a graph Laplacian. We will make use of this notion of a generalized Laplacian in the following section.

3. Combining Mixture Models and Graph Regularization

Our goal is to combine the mixture model and graph-based learning algorithms. Intuitively, to enforce the smoothness assumption encoded in the weight matrix W, we might regularize the mixture model by the energy of the posterior with respect to the graph. This leads naturally to minimization of the objective function

$$\mathcal{O}_\alpha(\theta) = -\alpha\, \ell(\theta) + (1-\alpha)\, \mathcal{E}(\theta) \qquad (3)$$

This objective function makes explicit the tension between maximizing the data likelihood and minimizing the graph energy. The most direct way of proceeding is to estimate parameters $\theta = (\gamma, \lambda, \eta)$ to minimize the objective $\mathcal{O}_\alpha(\theta)$ where $\alpha \in [0,1]$ is a coefficient that controls the relative strength of the two terms.

Note that while the energy $\mathcal{E}(\boldsymbol{f})$ may appear to be the logarithm of a prior $p(\boldsymbol{f}) \propto \exp(-\boldsymbol{f}^\top \Delta \boldsymbol{f})$, it in fact involves the observed labels $\mathbf{y}_L$, since $\boldsymbol{f}$ is fixed on the labeled data. Thus, it is perhaps best thought of as a discriminative component of the objective function, while $\ell(\theta)$ is the generative component. In other words, optimizing $\mathcal{O}_\alpha$ will carry out a combination of discriminative and generative learning. This is closely related to, but different from, the graph regularization framework of Belkin et al. (2004b).

Unfortunately, learning all of the parameters together is difficult—since the energy $\mathcal{E}(\theta)$ is discriminative, EM training is computationally demanding as the M-step does not have a closed-form solution; moreover, it has the usual drawback of local minima. We propose instead the following two-step approach.

Generative/Graph-Discriminative Training

Select the number of mixture components M, and initialize the parameters $\theta = (\gamma, \lambda, \eta)$.

1. Train the mixture model θ using the objective function $\mathcal{O}_1(\theta) = -\ell(\theta)$ with standard EM.
2. Fix γ and η, and reestimate the multinomial λ to minimize $\mathcal{O}_\alpha(\theta)$.

Figure 2. Training combining $\ell(\theta)$ and $\mathcal{E}(\theta)$.

Clearly this algorithm is suboptimal in terms of optimizing the objective function. However it has two important advantages. One advantage is that the first step is standard EM estimation. The second advantage is that, as we will show below, $\mathcal{O}_\alpha(\gamma, \lambda, \eta)$ is a *convex* function of λ.

3.1. Convexity of $\mathcal{O}_\alpha$

$\mathcal{O}_\alpha$ involves $\ell(\theta)$ and $\mathcal{E}(\theta)$. First let us consider $\ell(\theta)$. In (1) the sum over $i \in U$ is constant w.r.t. λ. The sum over $i \in L$ can be written as

$$\begin{aligned} \ell_L(\theta) \;=\; & \sum_{i \in L,\, y_i = 1} \log \sum_{m=1}^{M} \gamma_m\, \lambda_m\, p(x_i \,|\, \eta_m) \;+ \\ & \sum_{i \in L,\, y_i = -1} \log \sum_{m=1}^{M} \gamma_m\, (1-\lambda_m)\, p(x_i \,|\, \eta_m) \end{aligned}$$

Since we fix γ and η, the term within the first sum has the form $\log \sum_m a_m \lambda_m$. It can be directly verified that its Hessian is negative-definite:

$$\left[\frac{\partial \log \sum_m a_m \lambda_m}{\partial \lambda_i \partial \lambda_j} \right] \;=\; \frac{-aa^\top}{(\sum_m a_m \lambda_m)^2} \;\preceq\; 0$$

A similar calculation shows that the Hessian for the second term is negative-definite as well. Thus $\ell_L(\theta)$ is concave in λ, and $-\alpha\ell(\theta)$ is convex.

Next let us consider $\mathcal{E}(\theta)$. Define a $u \times M$ *responsibility matrix* R by $R_{im} = p(m \,|\, x_i)$, depending on γ and η, with R_m denoting the m-th column. We can write $\boldsymbol{f}_U = R\,\lambda$. We partition Δ into labeled and unlabeled parts, with Δ_{UU} being the submatrix on unlabeled points, Δ_{LL} on labeled points and so on. The graph energy is written as

$$\begin{aligned} \mathcal{E}(\theta) &= \boldsymbol{f}^\top \Delta \boldsymbol{f} \\ &= \boldsymbol{f}_L^\top \Delta_{LL} \boldsymbol{f}_L + 2\boldsymbol{f}_L^\top \Delta_{LU} \boldsymbol{f}_U + \boldsymbol{f}_U^\top \Delta_{UU} \boldsymbol{f}_U \\ &= \boldsymbol{f}_L^\top \Delta_{LL} \boldsymbol{f}_L + 2\boldsymbol{f}_L^\top \Delta_{LU} R\lambda + \lambda^\top R^\top \Delta_{UU} R\lambda \end{aligned}$$

Since $\Delta_{UU} \succeq 0$, the Hessian $2R^\top \Delta_{UU} R \succeq 0$ is positive semi-definite in λ. Thus $(1-\alpha)\mathcal{E}(\theta)$ is convex in λ. Putting it together, $\mathcal{O}_\alpha$ is convex in λ.

3.2. Special case: $\alpha = 0$

The graph-discriminative training in step 2 has a very special structure, which we now explain. We first consider the special case $\alpha = 0$ and then the general case $\alpha \in [0,1]$.

The case $\alpha = 0$ has a particularly simple closed form solution and interpretation. Notice that although $\alpha = 0$, the solution depends on the incomplete log-likelihood $\ell(\theta)$ through the choice of γ and η learned in step 1.

The parameters λ are constrained within $[0,1]^M$. However first let us consider λ that minimize $\mathcal{E}$ in the unconstrained problem. The solution to the linear system

$$\nabla_\lambda \mathcal{E} \;=\; R^\top \left(2\Delta_{UU} R\lambda + 2\Delta_{UL} \boldsymbol{f}_L\right) = \mathbf{0} \qquad (4)$$

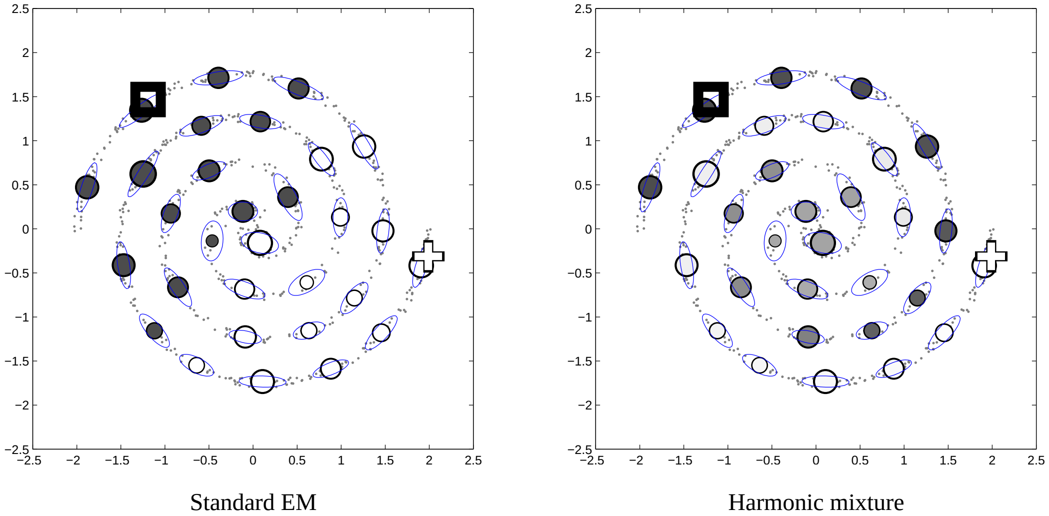


Figure 1. Predictions of Gaussian mixture models learned with the standard EM algorithm do not follow the manifold structure. Small dots are unlabeled data. Two labeled points, marked with a white $+$ and a black box, are at roughly the ends of the spirals. Each plot shows a Gaussian mixture model with $M = 36$ components, with the ellipses showing contours of the covariance matrices. The central dots have sizes proportional to the component weight $p(m) = \gamma_m$ (tiny components are not plotted), and its brightness indicates the strength of class membership, given by $\lambda_m \equiv p(y = 1 \,|\, m)$: white denotes $\lambda_m = 1$, black denotes $\lambda_m = 0$, and intermediate gray denotes values in between. Although the density $p(x)$ is estimated well by the standard mixture fit using EM (left), λ does not follow the data manifold. The right plot shows the harmonic mixture, where λ is refit to be harmonic on the backbone graph.

is given by

$$\lambda^\star = -\left(R^\top \Delta_{UU} R\right)^{-1} R^\top \Delta_{UL} \boldsymbol{f}_L \qquad (5)$$

Note that, in general, the constraints $0 \le \lambda_m \le 1$ must also be explicitly enforced. If the above solution lies in the interior of the hypercube $[0,1]^M$ then it must also be the solution of the constrained problem.[1] In this case, (5) determines the class membership probabilities for each mixture component — the soft label for the unlabeled data is given by $\boldsymbol{f}_U = R\lambda$. Previously unseen test points can be classified similarly.

Compare the solution (5), which we will refer to as the *harmonic mixture*, with the completely graph-based harmonic function solution (Zhu et al., 2003):

$$\begin{array}{ll} \text{harmonic:} & \boldsymbol{f}_U = -\Delta_{UU}^{-1} \Delta_{UL} \boldsymbol{f}_L \\ \text{harmonic mixture:} & \boldsymbol{f}_U = -R\left(R^\top \Delta_{UU} R\right)^{-1} R^\top \Delta_{UL} \boldsymbol{f}_L \end{array}$$

Computationally, obtaining the harmonic mixture requires the inversion of an $M \times M$ matrix, or if the solution lies on the boundary solving the associated constrained optimization problem. Solving the system (4) will be much less computationally expensive than the $u \times u$ matrix inversion required by harmonic solution, when the number of mixture components M is much smaller than the number of unlabeled points u. This reduction is possible because the $\boldsymbol{f}_U$ are now obtained by marginalizing the mixture model.

3.3. Graphical interpretation

The procedure just described can be interpreted graphically in terms of a much smaller *backbone graph* with *supernodes* induced by the mixture components. The backbone graph has the same l labeled nodes as in the original graph, but only M unlabeled supernodes.

By rearranging terms it is not hard to show that in the backbone graph, the generalized Laplacian is

$$\widetilde{\Delta} \;=\; \begin{pmatrix} I & 0 \\ 0 & R \end{pmatrix}^\top \begin{pmatrix} \Delta_{LL} & \Delta_{LU} \\ \Delta_{UL} & \Delta_{UU} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & R \end{pmatrix} \qquad (6)$$

Note that $\widetilde{\Delta}$ has zero row sums. The harmonic mixture parameter λ is then a harmonic function on the generalized Laplacian. The harmonic mixture algorithm is summarized in Figure 3.

Perhaps the best intuition for the backbone graph comes from considering hard clustering. In this case $R_{im} = 1$ if m is the cluster to which point i belongs, and $R_{im} = 0$

[1] More generally, the Karush-Kuhn-Tucker optimality conditions imply that the harmonic mixture can be expressed as $\lambda^\star = -\left(R^\top \Delta_{UU} R\right)^{-1} \left(R^\top \Delta_{UL} \boldsymbol{f}_L + \mu\right)$, where μ is a vector of Lagrange multipliers. Geometrically, this can be viewed as the solution of an inhomogeneous Dirichlet boundary value problem for a generalized Laplacian. Computationally if some λ_m are out of bounds, we clip them as the starting point for constrained convex optimization, which converges quickly. Pseudo inverse is used if R is rank deficient.

Harmonic Mixture Training: $\alpha = 0$

Select the number of mixture components M, initialize the parameters $\theta = (\gamma, \lambda, \eta)$, and form the graph Laplacian Δ.

1. Run standard EM to obtain γ, λ, and η.
2. Form the generalized Laplacian $\widetilde{\Delta}$, from (6).
3. Fixing γ and η, compute

$$\lambda^\star = \arg\min_\lambda \lambda^\top \widetilde{\Delta}_{UU} \lambda + 2\lambda^\top \widetilde{\Delta}_{UL} f_L$$

Output: Harmonic mixture model $\theta = (\gamma, \lambda^\star, \eta)$

Figure 3. The harmonic mixture algorithm, $\alpha = 0$

otherwise. Let $c(m) = \{i \,|\, R_{im} = 1\}$ denote cluster m. In this case the supernodes are the clusters themselves. Let w_{ij} be the weight between nodes i, j in the original graph. The equivalent weight between supernodes (s, t) reduces to $\tilde{w}_{st} = \sum_{i \in c(s),\, j \in c(t)} w_{ij}$; and the equivalent weight between a supernode s and a labeled node $j \in L$ is $\tilde{w}_{sj} = \sum_{i \in c(s)} w_{ij}$. In this case the solution (5) is also guaranteed to satisfy the constraints. One can create such a backbone graph using, for instance, k-means clustering.

3.4. General case: $\alpha > 0$

In the general case of $\alpha > 0$ step 2 does not have a closed-form solution. As λ_m must lie in the interval $[0, 1]$, we perform constrained convex optimization in this case. The gradient of the objective function is easily computed. Note that

$$\frac{\partial \ell(\theta)}{\partial \lambda_m} = \sum_{i \in L,\, y_i = 1} \frac{\gamma_m \, p(x_i \,|\, \eta_m)}{\sum_{k=1}^{M} \gamma_k \, p(x_i \,|\, \eta_k) \lambda_k} - \sum_{i \in L,\, y_i = -1} \frac{\gamma_m \, p(x_i \,|\, \eta_m)}{\sum_{k=1}^{M} \gamma_k \, p(x_i \,|\, \eta_k)\,(1 - \lambda_k)}$$

and $\partial \mathcal{E} / \partial \lambda$ was given in (4). One can also use sigmoid function to transform it into an unconstrained optimization problem with $\lambda_m = \sigma(\beta_m) = 1/\left(\exp(-\beta_m) + 1\right)$ and optimize the β parameters.

Although the objective function is convex, a good starting point for λ is important for fast convergence. We select an initial value for λ by solving a one dimensional problem first. We have two parameters at hand: λ_{EM}, the solution from the standard EM algorithm in step 1, and λ_{HM}, the harmonic mixture solution from the special case $\alpha = 0$. We find the optimal interpolated coefficient $\epsilon \in [0, 1]$ $\lambda_0 = \epsilon\, \lambda_{\text{EM}} + (1 - \epsilon)\, \lambda_{\text{HM}}$ that minimizes the objective function. Then, we start from λ_0 and use a quasi-Newton algorithm to find the global optimum for λ.

4. Experiments

We test harmonic mixtures on synthetic data, handwritten digits, image analysis and text categorization tasks. The emphases are on how the harmonic mixtures (denoted 'HM' below) perform against several baseline methods on unlabeled data; how they handle unseen data; and whether they can reduce the problem size. Unless otherwise noted, the harmonic mixtures are computed with $\alpha = 0$.

We use three baseline methods: sampling unlabeled data to create a smaller graph ('sample'), mixture models constructed with the standard EM algorithm ('EM'), and harmonic functions on the original large graphs ('graph'). In 'sample', we randomly draw M unlabeled points from U. We create a small (size $l + M$) graph with these and the labeled points, and compute the harmonic function f_i on the small graph first. The graph computation cost is thus the same as 'HM'. Then as in (Delalleau et al., 2005), we compute the labels for other points j by $f_j = (\sum_{i=1}^{l+M} w_{ij} f_i)/(\sum_{i=1}^{l+M} w_{ij})$.

4.1. Synthetic Data

First let us look at the synthetic dataset in Figure 1. It has a Swiss roll structure, and we hope the labels can follow the spiral arms. There is one positive and one negative labeled point, at roughly the opposite ends. We use $u = 766$ unlabeled points and an additional 384 points as unseen test data.

The mixture model and standard EM. To illustrate the idea, consider a Gaussian mixture model (GMM) with $M = 36$ components, each with full covariance. The left panel shows the converged GMM after running EM. The GMM models the manifold density $p(x)$ well. However the component class membership $\lambda_m \equiv p(y = 1 \,|\, m)$ (brightness of the central dots) does not follow the manifold. In fact λ takes the extreme values of 0 or 1 along a somewhat linear boundary instead of following the spiral arms, which is undesirable. The classification of data points will not follow the manifold either.

The graph and harmonic mixtures. Next we combine the mixture model with a graph to compute the harmonic mixtures, as in the special case $\alpha = 0$. We construct a fully connected graph on the $L \cup U$ data points with weighted edges $w_{ij} = \exp\left(-||x_i - x_j||^2/0.01\right)$. The weight parameters in all experiments are selected with 5-fold cross validation. We then reestimate λ, which are shown in the right panel of Figure 1. Note λ now follow the manifold as it changes from 0 (black) to approximately 0.5 (gray) and finally 1 (white). This is the desired behavior.

The particular graph-based method we use needs extra care. The harmonic function solution f is known to sometimes skew toward 0 or 1. This problem is easily corrected if we know or have an estimate of the proportion of positive and negative points, with the Class Mass Normalization heuristic (Zhu et al., 2003). In this paper we use a similar but simpler heuristic. Assuming the two classes are about equal in size, we simply set the decision boundary at $median(f)$.

Sensitivity to M. If the number of mixture components M is too small, the GMM is unable to model $p(x)$ well, let alone λ. In other words, the harmonic mixture is sensitive to M. M has to be larger than a certain threshold so that the manifold structure can appear. In fact M may need to be larger than the number of labeled points l, which is unusual in traditional mixture model methods for semi-supervised learning. But once M is over the threshold, further increase should not dramatically change the solution. In the end the harmonic mixture may approach the harmonic function solution when $M = u$.

Figure 4(top left) shows the classification accuracies on U as we change M. 'graph' is the ideal performance. We find that 'HM' threshold is around $M = 35$, at which point the accuracy jumps up and stabilizes thereafter. This is the number of mixture components needed for 'HM' to capture the manifold structure. 'sample' needs far more samples ($M > 400$, not shown) to reach 95% accuracy. 'EM' fails to make the labels to follow the manifold structure regardless of the number of mixtures.

Computational savings. 'HM' performs almost as good as 'graph' but with a much smaller problem size. As Figure 4(left) shows we only need to invert a 35×35 matrix, instead of a 766×766 one as required by 'graph'. The difference can be significant if U is even larger. There is of course the overhead of EM training.

Handling unseen data. Because 'HM' is a mixture model, it naturally handles unseen points. On 384 new test points 'HM' performs well, with accuracy 95.3% after $M \geq 35$ as shown in Figure 4(bottom left). Note 'graph' cannot handle unseen data and is therefore not shown in the plot.

4.2. Handwritten Digits Recognition

We use the '1vs2' dataset which contains handwritten digits of 1s and 2s. Each gray scale image is 8×8, which is represented by a 64 dimensional vector of pixel values. We use $l + u = 1600$ images as the labeled and unlabeled set, and 600 additional images as unseen new data to test induction. The total numbers of 1s and 2s are the same.

The mixture model. We use Gaussian mixture models. To avoid data sparseness problem, we model each Gaussian component with a spherical covariance, i.e. diagonal covariance matrix with the same variance in all dimensions. Different components may have different variances. We set the initial means and variances of the GMM with k-means algorithm before running EM.

The graph. We use a symmetrized 10-nearest-neighbor weighted graph on the 1600 images. That is, images i, j are connected if i is within j's 10NN or vice versa, as measured by Euclidean distance. The weights are $w_{ij} = \exp\left(-||x_i - x_j||^2/140^2\right)$.

Sensitivity to M. As illustrated in the synthetic data, the number of mixture components M needs to be large enough for harmonic mixture to work. We vary M and observe the classification accuracies on the unlabeled data with different methods. For each M we perform 20 trials with random L/U split, and plot the mean of classification accuracies on U in Figure 4(top center). The experiments were performed with labeled set size fixed at $l = 10$. We conclude that 'HM' needs only $M \approx 150$ components to match the performance of 'graph'. 'HM' outperforms both 'sample' and 'EM'.

Computational savings. In terms of graph method computation, we invert a 150×150 matrix instead of the original 1590×1590 matrix for harmonic function. This is good saving with little sacrifice in accuracy.

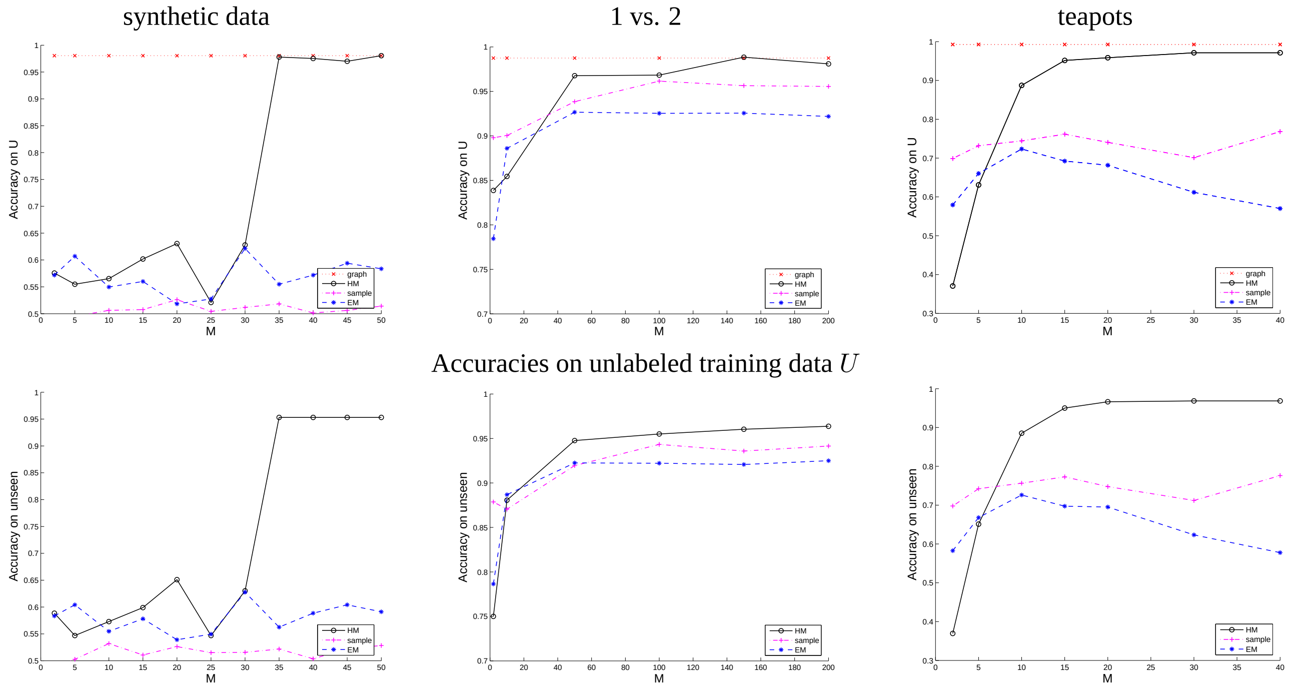
Handling unseen data. On 600 unseen data points (Figure 4 bottom center), 'HM' is better than 'sample' and 'EM' too.

The general case $\alpha > 0$. We also vary the parameter α between 0 and 1, which balances the generative and discriminative objectives. In our experiments $\alpha = 0$ always gives the best accuracies.

4.3. Teapots Image Analysis

We perform binary classification on the Teapots dataset, which was previously used for dimensionality reduction. See (Weinberger et al., 2004) for details. The dataset consists of a series of teapot photos, each rotated by a small angles. Our task is to identify whether the spout points to the left or the right. Excluding the few images from the original dataset in which the spout is roughly in the middle, we arrive at 365 images. We process each image by converting it to gray scale and down-sizing it to 12×16. Each image is thus represented by a 192-dimensional vector of pixel values. Nonetheless we believe the dataset resides on a much lower dimensional manifold, since image pairs in which the teapot rotates by a small angle are close to each other. Therefore we expect graph-based semi-supervised learning methods to perform well on the dataset. We use 273 images as $L \cup U$, and the remaining 92 as unseen test data.

The mixture model. We again use Gaussian mixture models with spherical covariances. We initialize the models with k-means before running EM.


synthetic data

1 vs. 2

teapots

Accuracies on unlabeled training data U

Accuracies on unseen test data (induction)

Figure 4. Sensitivity to M in the synthetic data (left), 1 vs. 2 (center) and teapots (right). Shown are the classification accuracies on U (top row) and unseen new data (bottom row) as M changes. 'graph' is the harmonic function on the complete $L \cup U$ graph; 'HM' is the harmonic mixture, 'sample' is the smaller graph with sampled unlabeled data, and 'EM' is the standard EM algorithm. Note y-axes are not on the same scale.

The graph. We use a symmetrized 10NN weighted graph on the 273 images, with weights $w_{ij} = \exp\left(-||x_i - x_j||^2/100^2\right)$.

Sensitivity to M. The classification accuracies on U with different number of components M is shown in Figure 4 (top right). Each curve is the average of 20 trials. l is fixed at (merely) 2. With $M > 15$ components, 'HM' approaches the 'graph' performance. 'sample' and 'EM' are clearly worse.

Computational savings. The graph computation for 'HM' inverts a 15×15 matrix, which is much cheaper than 271×271 for 'graph'.

Handling unseen data. 'HM' performs similarly on the 92 unseen images (Figure 4 bottom right), achieving high accuracy with small M, and outperforms 'sample' and 'EM'.

4.4. Text Categorization: A Discussion

We also perform text categorization on the PC vs. Mac groups from the 20-newsgroups data. Of the 1943 documents, we use 1600 as $L \cup U$ and the rest as unseen test data. We use a symmetrized 10NN weighted graph on the 1600 documents with weight $w_{uv} = \exp\left(-(1 - c_{uv})/0.03\right)$, where c_{uv} is the cosine between the $tf.idf$ document vectors u, v. With $l = 10$, 'graph' accuracy is around 90%. We use multinomial mixture models on documents. However unlike other tasks, 'HM' suffers from a loss of transductive accuracy, and only reaches 83% accuracy on U and unseen data. It does so with an undesirably large M around 600. Furthermore 'HM' and 'sample' perform about the same (though both are better than 'EM').

Why does 'HM' perform well on other tasks but not on text categorization? We think the reasons are: 1) The manifold assumption needs to hold strongly. For instance in the synthetic and the Teapots data the manifolds are evident, and 'HM' achieved close approximations to 'graph'. The text data seems to have a weaker manifold structure. 2) On top of that, the text data has a very high dimensionality ($D = 12008$). The curse of dimensionality may prevent a generative mixture model from fitting the manifold well. In addition the multinomial model may not be appropriate for creating localized supernodes.

Interestingly we do not have to use generative models. If we work with $\alpha = 0$, all we need from the mixture model is the responsibility R. One can instead first use simple prototype methods like k-means to cluster the data, and then train discriminative models to obtain R. This remains a future research direction.

5. Related Work

Recently Delalleau et al. (2005) used a small subset of the unlabeled data to create a small graph for semi-supervised learning. This is related to the Nyström method in spectral clustering (Fowlkes et al., 2004), and to the random 'landmarks' in dimensionality reduction (Weinberger et al., 2005). Our method is different in that it incorporates a generative mixture model, which is a second knowledge source besides the original graph. Our method outperforms random subset selection, and can be viewed as a principled way to carry out the elaborate subset selection heuristics in (Delalleau et al., 2005).

In terms of handling unseen data, our approach is closely related to the regularization framework of Belkin et al. (2004b); Krishnapuram et al. (2005) as graph regularization on mixture models. But instead of a regularization term we used a discriminative term, which allows for the closed form solution in the special case.

6. Summary

To summarize, our proposed harmonic mixture method reduces the graph problem size, and handles unseen test points. It achieves comparable accuracy as the harmonic function on complete graph for semi-supervised learning.

There are some open questions. One is when $\alpha > 0$ would be useful in practice. Another is whether we can use fast prototype methods instead of EM. Finally, we want to automatically select the appropriate number of mixture components M.

Acknowledgments

We thank the reviewers for their useful comments, and Guy Lebanon and Lillian Lee for interesting discussions.

References

Belkin, M., Matveeva, I., & Niyogi, P. (2004a). Regularization and semi-supervised learning on large graphs. *COLT*.

Belkin, M., Niyogi, P., & Sindhwani, V. (2004b). *Manifold regularization: A geometric framework for learning from examples* (Technical Report TR-2004-06). University of Chicago.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proc. 18th International Conf. on Machine Learning*.

Castelli, V., & Cover, T. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, *42*, 2101–2117.

Cozman, F., Cohen, I., & Cirelo, M. (2003). Semi-supervised learning of mixture models. *ICML-03, 20th International Conference on Machine Learning*.

Delalleau, O., Bengio, Y., & Roux, N. L. (2005). Efficient non-parametric function induction in semi-supervised learning. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*.

Fowlkes, C., Belongie, S., Chung, F., & Malik, J. (2004). Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 214–225.

Krishnapuram, B., Williams, D., Xue, Y., Hartemink, A., Carin, L., & Figueiredo, M. (2005). On semi-supervised classification. In L. K. Saul, Y. Weiss and L. Bottou (Eds.), *Advances in neural information processing systems 17*. Cambridge, MA: MIT Press.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Ratsaby, J., & Venkatesh, S. (1995). Learning from a mixture of labeled and unlabeled examples with parametric side information. *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, 412–417.

Weinberger, K. Q., Packer, B. D., & Saul, L. K. (2005). Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*.

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. *Proceedings of ICML-04* (pp. 839–846).

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schlkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing System 16*.

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

Zhu, X., Kandola, J., Ghahramani, Z., & Lafferty, J. (2005). Nonparametric transforms of graph kernels for semi-supervised learning. In L. K. Saul, Y. Weiss and L. Bottou (Eds.), *Advances in neural information processing systems 17*. Cambridge, MA: MIT Press.