Parametric Embedding for Class Visualization

Tomoharu Iwata
Kazumi Saito
Naonori Ueda
NTT Communication Science Laboratories, Japan

Sean Stromsten
BAE Systems Advanced Information Technologies, USA

Thomas L. Griffiths
Department of Cognitive and Linguistic Sciences
Brown University, USA

Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology, USA

Abstract

We propose a new method, Parametric Embedding (PE), that embeds objects with the class structure into a low-dimensional visualization space. PE takes as input a set of class conditional probabilities for given data points, and tries to preserve the structure in an embedding space by minimizing a sum of Kullback-Leibler divergences, under the assumption that samples are generated by a Gaussian mixture with equal covariances in the embedding space. PE has many potential uses depending on the source of the input data, providing insight into the classifier's behavior in supervised, semi-supervised and unsupervised settings. The PE algorithm has a computational advantage over conventional embedding methods based on pairwise object relations since

its complexity scales with the product of the number of objects and the number of classes. We demonstrate PE by visualizing supervised categorization of web pages, semi-supervised categorization of digits, and the relations of words and latent topics found by an unsupervised algorithm, Latent Dirichlet Allocation.

1 Introduction

Recently there has been great interest in algorithms for constructing low-dimensional feature-space embeddings of high-dimensional data sets. These algorithms seek to capture some aspect of the data set's intrinsic structure in a low-dimensional representation that is easier to visualize or more efficient to process by other learning algorithms. Typical embedding algorithms take as input a matrix of data coordinates in a high-dimensional ambient space (e.g., PCA (Jolliffe, 1980)), or a matrix of metric relations between pairs of data points (MDS (Torgerson, 1958), Isomap (Tenenbaum, Silva, & Langford, 2000), SNE (Hinton & Roweis, 2002)). The algorithms generally attempt to map nearby input points onto nearby points in the output embedding.

Here we consider a different sort of embedding problem with two sets of points $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and $C = \{c_1, \ldots, c_K\}$, which we call "objects" (X) and "classes" (C). The input consists of conditional probabilities $p(c_k|\boldsymbol{x}_n)$ associating each object $\boldsymbol{x}_n$ with each class c_k. Many kinds of data take this form: for a classification problem, C may be the set of classes, and $p(c_k|\boldsymbol{x}_n)$ the posterior distribution over these classes for each object $\boldsymbol{x}_n$; in a marketing context, C might be a set of products and $p(c_k|\boldsymbol{x}_n)$ the probabilistic preferences of a consumer; or in language modeling, C might be a set of semantic topics, and $p(c_k|\boldsymbol{x}_n)$ the distribution over topics for a particular document, as produced by a method like Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003). Typically, the number of classes is much smaller than the number of objects, $K \ll N$.

We seek a low-dimensional embedding of both objects and classes such that the distance between object $\boldsymbol{x}_n$ and class c_k is monotonically related to the probability $p(c_k|\boldsymbol{x}_n)$. This embedding simultaneously represents not only the relations between objects and classes, but also the relations within the set of objects and within the set of classes – each defined in terms of relations to points in the other set. That is, objects that tend to be associated with the same classes should be embedded nearby, as should classes that tend to have

the same objects associated with them. Our primary goals are visualization and structure discovery, so we typically work with two- or three-dimensional embeddings.

Object-class embeddings have many potential uses, depending on the source of the input data. If $p(c_k|\boldsymbol{x}_n)$ represents the posterior probabilities from a supervised Bayesian classifier, an object-class embedding provides insight into the behavior of the classifier: how well separated the classes are, where the errors cluster, whether there are clusters of objects that "slip through a crack" between two classes, which objects are not well captured by any class, and which classes are intrinsically most confusable with each other. Answers to these questions could be useful for improved classifier design. The probabilities $p(c_k|\boldsymbol{x}_n)$ may also be the product of unsupervised or semi-supervised learning, where the classes c_k represent components in a generative mixture model. Then an object-class embedding shows how well the intrinsic structure of the objects (and, in a semi-supervised setting, any given labels) accords with the clustering assumptions of the mixture model.

Our specific formulation of the embedding problem assumes that each class c_k can be represented by a spherical Gaussian distribution in the embedding space, so that the embedding as a whole represents a simple Gaussian mixture model for each object $\boldsymbol{x}_n$. We seek an embedding that matches the conditional probabilities for each object under this Gaussian mixture model to the input probabilities $p(c_k|\boldsymbol{x}_n)$. Minimizing the Kullback-Leibler (KL) divergence between these two probability distributions leads to an efficient algorithm, which we call *Parametric Embedding* (PE).

The rest of this article is organized as follows. In the next section, we formulate PE, and in Section 3, we describe the optimization procedures. In Section 4, we briefly review related work. In Section 5, we compare PE with conventional methods by visualizing classified web pages. In Section 6, we visualize hand written digits with two classifiers, and show that PE can visualize the characteristics of assumed models as well as given data. In Section 7, we show that PE can visualize latent topic structure discovered by an unsupervised method. Finally, we present concluding remarks and discussion of future work in Section 8.

2 Parametric Embedding

Given as input conditional probabilities $p(c_k|\boldsymbol{x}_n)$, PE seeks an embedding of objects with coordinates $\boldsymbol{R} = \{\boldsymbol{r}_n\}_{n=1}^N$ and classes with coordinates $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_k\}_{k=1}^K$, such that $p(c_k|\boldsymbol{x}_n)$ is approximated as closely as possible by the conditional probabilities under the assumption of a unit-variance spherical Gaussian mixture model in the embedding space:

$$p(c_k|\boldsymbol{r}_n) = \frac{p(c_k)\exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2)}{\sum_{l=1}^K p(c_l)\exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_l\|^2)}, \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm in the embedding space. The dimension of the embedding space is D, and $\boldsymbol{r}_n \in \mathcal{R}^D$, $\boldsymbol{\phi}_k \in \mathcal{R}^D$. When the conditional probabilities $p(c_k|\boldsymbol{x}_n)$ arise as posterior probabilities from a mixture model, we will also typically be given priors $p(c_k)$ as input; otherwise the $p(c_k)$ terms above may be assumed equal. Assuming this model in the embedding space, if the Euclidean distance between object $\boldsymbol{r}_n$ and class $\boldsymbol{\phi}_k$ is small,the conditional probability $p(c_k|\boldsymbol{r}_n)$ becomes high. Therefore, we can understand the input conditional probabilities from the visualization result.

It is natural to measure the degree of correspondence between input probabilities and embedding-space probabilities using a sum of KL divergences for each object: $\sum_{n=1}^N \mathrm{KL}(p(c_k|\boldsymbol{x}_n)\|p(c_k|\boldsymbol{r}_n))$. Minimizing this sum w.r.t. $p(c_k|\boldsymbol{r}_n)$ is equivalent to minimizing the objective function

$$E(\boldsymbol{R}, \boldsymbol{\Phi}) = -\sum_{n=1}^N \sum_{k=1}^K p(c_k|\boldsymbol{x}_n)\log p(c_k|\boldsymbol{r}_n). \tag{2}$$

Gradients of E w.r.t. $\boldsymbol{r}_n$ and $\boldsymbol{\phi}_k$ are respectively (see Appendix A):

$$\frac{\partial E}{\partial \boldsymbol{r}_n} = \sum_{k=1}^K (p(c_k|\boldsymbol{x}_n) - p(c_k|\boldsymbol{r}_n))(\boldsymbol{r}_n - \boldsymbol{\phi}_k), \tag{3}$$

$$\frac{\partial E}{\partial \boldsymbol{\phi}_k} = \sum_{n=1}^N (p(c_k|\boldsymbol{x}_n) - p(c_k|\boldsymbol{r}_n))(\boldsymbol{\phi}_k - \boldsymbol{r}_n). \tag{4}$$

These learning rules have an intuitive interpretation (analogous to those in SNE) as a sum of forces pulling or pushing $\boldsymbol{r}_n$ ($\boldsymbol{\phi}_k$) depending on the difference of conditional probabilities.

Importantly, the Hessian of E w.r.t. $\boldsymbol{r}_n$ is a semi-positive definite matrix (see Appendix B):

$$\frac{\partial^2 E}{\partial \boldsymbol{r}_n \partial \boldsymbol{r}_n^T} = \sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n)\boldsymbol{\phi}_k\boldsymbol{\phi}_k^T - \left(\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n)\boldsymbol{\phi}_k\right)\left(\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n)\boldsymbol{\phi}_k\right)^T, \quad (5)$$

since the r.h.s. of (5) is exactly a covariance matrix, where T represents transpose. When we add regularization terms to the above objective function as follows:

$$J(\boldsymbol{R}, \boldsymbol{\Phi}) = E(\boldsymbol{R}, \boldsymbol{\Phi}) + \eta_r \sum_{n=1}^{N} \|\boldsymbol{r}_n\|^2 + \eta_\phi \sum_{k=1}^{K} \|\boldsymbol{\phi}_k\|^2, \eta_r, \eta_\phi > 0, \quad (6)$$

the Hessian of the objective function J w.r.t. $\boldsymbol{r}_n$ becomes positive definite. Thus we can find the globally optimal solution for $\boldsymbol{R}$ given $\boldsymbol{\Phi}$.

The visualization result depends on only initial coordinates of classes $\boldsymbol{\Phi}$, not initial coordinates of objects $\boldsymbol{R}$, since we can find the globally optimal solution for $\boldsymbol{R}$ given $\boldsymbol{\Phi}$. On the other hand, the result of conventional nonlinear embedding methods (e.g. SNE) depends on the initial coordinates of objects $\boldsymbol{R}$. Therefore, we can get more stable results than conventional nonlinear methods in the case that the number of classes is much smaller than the number of objects. The dependence of initial conditions was small in our experiments (see Section 5.4).

3 Algorithm

We minimize the objective function J by alternately optimizing $\boldsymbol{R}$ while fixing $\boldsymbol{\Phi}$, and optimizing $\boldsymbol{\Phi}$ while fixing $\boldsymbol{R}$, until J converges. The optimization procedure can be summarized as follows:

1. Initialize $\boldsymbol{R}$ and $\boldsymbol{\Phi}$ randomly
2. Calculate $\{\frac{\partial J}{\partial \boldsymbol{r}_n}\}_{n=1}^N$
3. If $\|\frac{\partial J}{\partial \boldsymbol{r}_n}\|^2 < \epsilon_r$ for all $n = 1, \ldots, N$, go to Step 6
4. Calculate modification vectors $\{\boldsymbol{\Delta} r_n\}_{n=1}^N$
5. Update $\boldsymbol{R}$ by $\boldsymbol{r}_n = \boldsymbol{r}_n + \boldsymbol{\Delta} r_n$, and go to Step 2

6. Calculate $\frac{\partial J}{\partial \boldsymbol{\phi}}$

7. If $\|\frac{\partial J}{\partial \boldsymbol{\phi}}\|^2 < \epsilon_\phi$, output $\boldsymbol{R}$, $\boldsymbol{\Phi}$ and terminate

8. Calculate the modification vector $\boldsymbol{\Delta\phi}$

9. Update $\boldsymbol{\Phi}$ by $\boldsymbol{\phi} = \boldsymbol{\phi} + \boldsymbol{\Delta\phi}$, and go to Step 2

where $\boldsymbol{\phi} = (\boldsymbol{\phi}_1^T, \ldots, \boldsymbol{\phi}_K^T)^T$, and ϵ_r and ϵ_ϕ are convergence precisions. In Step 1, if we have information of classes, as the initial values for $\boldsymbol{\Phi}$, we may use the result of other embedding methods such as MDS. From Step 2 to Step 5, $\boldsymbol{R}$ is moved to minimize J while $\boldsymbol{\Phi}$ is fixed. In Step 4, according to Newton methods, $\boldsymbol{\Delta}r_n$ is calculated using the Hessian w.r.t. $\boldsymbol{r}_n$ as follows:

$$\boldsymbol{\Delta}r_n = -\left(\frac{\partial^2 J}{\partial \boldsymbol{r}_n \partial \boldsymbol{r}_n^T}\right)^{-1} \frac{\partial J}{\partial \boldsymbol{r}_n}. \tag{7}$$

Since $\frac{\partial^2 J}{\partial \boldsymbol{r}_n \boldsymbol{r}_n^T}$ is positive definite as described above, the inverse always exists. From Step 6 to Step 9, $\boldsymbol{\Phi}$ is moved to minimize J while $\boldsymbol{R}$ is fixed. In Step 9, according to quasi-Newton methods, $\boldsymbol{\Delta}\phi$ is calculated as follows:

$$\boldsymbol{\Delta}\phi = -\lambda \boldsymbol{G}^{-1} \frac{\partial J}{\partial \boldsymbol{\phi}}, \tag{8}$$

where $\boldsymbol{G}^{-1}$ is the approximation of $\left(\frac{\partial^2 J}{\partial \boldsymbol{\phi} \partial \boldsymbol{\phi}^T}\right)^{-1}$ that is calculated by limited memory BFGS (Saito & Nakano, 1997) and the step length λ is calculated so as to minimize the objective function.

Step 2 and Step 6 can be calculated using $O(NKD)$ multiplications, and Step 4 can be calculated using $O(ND^3)$ multiplications. The complexity of Step 9 is $O(KDs)$, where s is the memory size in limited memory BFGS (Saito & Nakano, 1997). Thus, the complexity of a single iteration of PE is $O(NK)$, when we assume the dimension of the embedding space D and the memory size s are constant. We found experimentally that the number of iterations doesn't grow with N (see Fig.3).

4 Related work

MDS and PCA are representative linear embedding methods. MDS embeds objects so as to preserve objects' pair-wise distances, and PCA embeds

objects so as to maximize variance. These methods can find globally optimal embeddings and are computationally efficient, but they cannot represent nonlinear structure. Recently, therefore, a number of nonlinear embedding methods have been proposed, such as Isomap, local linear embedding (Roweis & Saul, 2000), SNE, and connectivity preserving embedding (Yamada, Saito, & Ueda, 2003). However, these nonlinear embedding methods are more computationally expensive than linear methods and PE. Furthermore, these embedding methods do not use any class information.

Fisher linear discriminant analysis (Fisher, 1950) and kernel discriminant analysis (KDA) (Baudat & Anouar, 2000) (Mika, Ratsch, Weston, Scholkopf, & Muller, 1999) are embedding methods that do use class information. FLDA embeds objects so as to maximize between-class variance and minimize within-class variance. KDA extends FLDA to nonlinear embedding by using the kernel method. FLDA and KDA are dimensionality reduction methods for data given as a set of class-object pairs $\{\boldsymbol{x}_n, c(n)\}_{n=1}^N$ ($c(n)$ is the class label of a object $\boldsymbol{x}_n$), PE, by contrast, uses conditional class probabilities rather than hard classifications.

PE can be seen as a generalization of stochastic neighbor embedding (SNE). SNE corresponds to a special case of PE where the objects and classes are identical sets. In SNE, the class conditional probabilities $p(c_k|\boldsymbol{x}_n)$ are replaced by the probability $p(\boldsymbol{x}_m|\boldsymbol{x}_n)$ of object $\boldsymbol{x}_n$ under a Gaussian distribution centered on $\boldsymbol{x}_m$. When the inputs (conditional probabilities) to PE come from an unsupervised mixture model, PE performs unsupervised dimensionality reduction just like SNE. However, it has several advantages over SNE and other methods for embedding a single set of data points based on their pairwise relations. When class labels are available, it can be applied in supervised or semi-supervised modes. Because its computational complexity scales with NK, the product of the number of objects and the number of classes, it can be applied efficiently to data sets with very many objects (as long as the number of classes remains small). In this sense, PE is closely related to landmark MDS (LMDS) (Silva & Tenenbaum, 2002), if we equate classes with landmarks, objects with data points, and $-\log p(c_k|\boldsymbol{x}_n)$ with the squared distances input to LMDS. However, LMDS lacks a probabilistic semantics and is only suitable for unsupervised settings. The formulation of co-occurrence data embedding (Globerson, Chechik, Pereira, & Tishby, 2005) is similar PE, but it embeds objects of different types based on their co-occurrence statistics. PE embeds objects and classes based on parametric models which describe their relationships. Mei and Shelton also proposed a

method to embed objects of different types, but they focused on visualizing collaborative filtering data with ratings (Mei & Shelton, 2006).

5 Visualization of labeled data

In this section, we show how PE helps visualize labeled data, and compare PE with conventional methods (MDS, Isomap, SNE, FLDA, KDA) in terms of visualization, conditional probability approximation, and computational complexity.

5.1 Experimental setting

We visualized 5000 Japanese web pages categorized into 10 topics by the Open Directory Project[1], where objects are web pages and classes are topic categories. We omitted pages with fewer than 50 words and those in multiple categories. Each page is represented as a word frequency vector, and the class prior and conditional probabilities are obtained from a naive Bayes model (McCallum & Nigam, 1998) trained on these data (see Appendix C). The dimension of the word frequency vector is 34,248. We used $\eta_r = 0.1$, $\eta_\phi = 50$ for parameters of PE.

5.2 Compared methods

We used 7 methods for comparison that are closely related to PE:

- MDS1 : the input is squared Euclidean distances between word frequency vectors divided by L_2 norm:

$$d_{ij}^{MDS1} = \left\| \frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|} - \frac{\boldsymbol{x}_j}{\|\boldsymbol{x}_i\|} \right\|^2. \tag{9}$$

- MDS2 : the input is squared Euclidean distances between conditional probabilities:

$$d_{ij}^{MDS2} = \sum_{k=1}^{K} (p(c_k|\boldsymbol{x}_i) - p(c_k|\boldsymbol{x}_j))^2. \tag{10}$$

[1] http://dmoz.org/

- Isomap1 : the input is squared Euclidean distances between word frequency vectors divided by L_2 norm as (9). We used 10-nearest neighbor approach to construct the graph.

- Isomap2 : the input is KL divergences between conditional probabilities:
$$d_{ij}^{Isomap2} = \mathrm{KL}(p(c_k|\boldsymbol{x}_i)\|p(c_k|\boldsymbol{x}_j)). \tag{11}$$
We used 10-nearest neighbor approach to construct the graph. The input distance of Isomap need not be symmetric, since the shortest path distances become symmetric even if the input is not symmetric distance.

- SNE : the input is KL divergences between conditional probabilities as (11).

- FLDA : the input is word frequency vectors that are reduced to dimension 2000 by PCA and their classes [2].

- KDA : the input is word frequency vectors and their classes. We used Gaussian kernels with variance 1, and regularization as in (Mika et al., 1999) with regularization parameter 10^{-3}.

5.3 Visualization results

Fig.1(a) is the result of PE. Each point represents a web page, and the shape represents the class. Pages from the same class cluster together, and closely related pairs of classes, such as sports and health, or computers and online-shopping, are located nearby. There are few objects near the sports cluster, so sports pages are easy to distinguish from others. On the other hand, the regional cluster is central and diffuse, and there are many objects from other classes mixed in with it; regional is apparently a vague topic. These can be confirmed by F-measure for each class (Table 1), which is the harmonic mean of precision and recall. The precision of class c_k is the ratio

[2] If the dimension of an object is higher than N-K, the between-class covariance matrix becomes singular, and FLDA is not applicable (small sample size problem (Fukunaga, 1990)). We avoided this problem using PCA as in (Belhumeur, Hespanha, & Kriegman, 1997).

Table 1: F-measure for each class

arts	sports	business	computers	health	home	recreation	regional	science	online-shop
0.973	0.978	0.929	0.924	0.967	0.957	0.958	0.909	0.964	0.941

of the number of objects correctly estimated at class c_k compared to the total number of objects estimated at class c_k, and the recall of class c_k is the ratio of the number of objects correctly estimated at class c_k compared to the total number of objects classified into class c_k. The estimated class is the class that has the highest conditional probability. The high F-measure of sports reflects the easiness of the classification, and the low F-measure of regional reflects the difficulty of the classification. Furthermore, we can visualize not only the relations among classes, but also how pages relate to their classes. For example, pages that are located at the center of cluster are typical pages for the class, and pages that are located between clusters have multiple topics. Some pages are located in the cluster of different classes; these may be misclassified pages.

MDS1 and Isomap1 do not use class information, therefore they yields no clear class structure (Fig.1(b)(d)). Fig.1(c) is the result of MDS2. Pages from the same class are embedded closely, but some classes are overlapping, so we do not see the class structure as clearly as we do with PE. In the result of Isomap2 and SNE (Fig.1(e)(f)), we can clearly see the class structure as in PE. Fig.1(g) is the result of FLDA. Since FLDA use the class information, pages are more class-clustered than in MDS1. However many clusters are overlapping and it is difficult to understand the relationships among classes. Linear embedding methods, cannot, in general, separate all the classes. Fig.1(h) is the result of KDA. All clusters are separated perfectly, and we can understand the relationships among classes. However, little within-class structure is visible.

5.4 Comparison on conditional probability approximation

We evaluate the degree of conditional probability approximation quantitatively. Since conditional probabilities are not given as inputs in MDS1, Isomap1, FLDA and KDA, we compare PE, MDS2, Isomap2 and SNE.

Let $\boldsymbol{X}_k^{high}(h) = \{\boldsymbol{x}_{k,1}^{high}, \ldots, \boldsymbol{x}_{k,h}^{high}\}$ be the set of h objects with the high-

est conditional probabilities $p(c_k|\boldsymbol{x}_n)$ in the class c_k, and let $\boldsymbol{X}_k^{close}(h) = \{\boldsymbol{x}_{k,1}^{close}, \ldots, \boldsymbol{x}_{k,h}^{close}\}$ be the set of h objects closest to the class center $\boldsymbol{\phi}_k$ in the embedding space. If conditional probabilities are approximated perfectly, $\boldsymbol{X}_k^{high}(h)$ and $\boldsymbol{X}_k^{close}(h)$ should be identical in each class c_k, since high posterior objects should be embedded close to the class. As a measure of conditional probability approximation, we use the precision between $\boldsymbol{X}_k^{high}(h)$ and $\boldsymbol{X}_k^{close}(h)$ as follows:

$$prec(h) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{h}|\boldsymbol{X}_k^{high}(h) \cap \boldsymbol{X}_k^{close}(h)| \tag{12}$$

where $|\cdot|$ is the number of elements. MDS2, Isomap2 and SNE do not output $\boldsymbol{\Phi}$. Here, we take the object that has the highest conditional probability as a representative point of the class, and define the coordinates of the class c_k to be the coordinates of this object. (i.e. $\boldsymbol{\phi}_k = \boldsymbol{r}_{\arg_n \max p(c_k|\boldsymbol{x}_n)}$).

Fig.2 shows precisions of PE, MDS2, Isomap2 and SNE as h goes from 10 to 500. Each error bar of PE represents the standard deviation of 100 results with different initial conditions. By this measure, the conditional probability approximation of PE is clearly better than those of the other methods. In Isomap2 and SNE, precisions are low in small h. This is because Isomap2 and SNE preserve not object-class relationships but objects' pairwise neighborhood relationships. In MDS2, precision goes down as h increase. This is because classes are overlapping as in Fig.1(c).

5.5 Comparison on computational time

One of the main advantages of PE is its efficiency. As described in Section 3, the computational complexity of a single iteration of PE is $O(NK)$. That is, the computational time increases linearly with the number of objects. We evaluated the number of iterations of PE experimentally. Fig.3 shows that the number of iterations does not depend on the number of objects, where each error bar represents the standard deviation of 1000 results with different initial conditions.

MDS computes eigenvectors of the matrix $\boldsymbol{B} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{A}^2\boldsymbol{H}$, where $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T$ is the centering matrix and $\boldsymbol{A}$ is the distance matrix. If the input of MDS is squared Euclidean distances ($A_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$), the complexity of MDS increases linearly with the number of objects by Lanczos methods (Golub & Van Loan, 1996), since the matrix-vector product of

Table 2: The slopes of regression lines in Fig.4

PE	MDS1	MDS2	Isomap1	Isomap2	SNE	FLDA	KDA
0.749	0.770	0.834	2.722	2.824	2.232	2.898	2.998

$\boldsymbol{B} = \boldsymbol{H}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{H}$ can be calculated using $O(NV)$ multiplications, where V is the number of non-zero elements in each row of $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. Isomap has $O(N^3)$ complexity (Silva & Tenenbaum, 2002). SNE has $O(N^2)$ complexity since it uses objects' pair-wise relationships. FLDA and KDA lead to generalized eigenvalue problems, whose complexity is the order of the cube of matrix size.

We measured computational time experimentally, varying the number of objects from 500 to 5000, on a Xeon 3.2GHz CPU, 2GB memory PC. Fig.4 shows the result [3]. The x-axis and the y-axis shows the logarithm of number of objects and the logarithm of computational time (sec), respectively. The dotted line is the regression line in the log-log plot. Note that preprocessing time (conditional probability estimation in PE, MDS2, Isomap2 and SNE, and dimensionality reduction by PCA in FLDA) is omitted. Table 2 shows the slopes of regression lines. The slopes represent how computational time increases with the number of objects. The results are consistent with the theoretical computational complexities as described above, even though they are not same since iterative methods are used in all methods.

In the case of 5000 objects, the computational time of PE is 3.13 sec. Even taking into account the preprocessing time of PE (2.33sec), PE is more efficient than Isomap1, Isomap2, SNE, FLDA and KDA (with computational times of 1593sec, 704sec, 1869sec, 211sec and 6752sec, respectively).

5.6 Summary of comparison

In our experiments, we showed that PE approximates conditional probabilities well, and is quite efficient compared to conventional methods. MDS is also efficient, but does not extract the class structure. SNE and Isomap2 achieves results similar to those of PE, but take more time. FLDA and KDE are different from PE in input information, and also take more time.

[3] As a preprocessing step of FLDA, input vectors are reduced by PCA to $N - K$ dimensions if $N \leq 2000$.

6 Visualization of classifiers

The utility of PE for analyzing classifier performance may best be illustrated in a semi-supervised setting, with a large unlabeled set of objects and a smaller set of labeled objects. We fit a probabilistic classifier based on the labeled objects, and we would like to visualize the behavior of the classifier applied to the unlabeled objects, in a way that suggests how accurate the classifier is likely to be and what kinds of errors it is likely to make.

We constructed a simple probabilistic classifier for 2558 handwritten digits (classes 0-4) from the MNIST database. The classifier was based on a mixture model for the density of each class, defined by selecting either 10 or 100 digits uniformly at random from each class and centering a fixed-covariance Gaussian (in pixel space) on each of these examples – essentially a soft nearest-neighbor method (see Appendix D). The posterior distribution over this classifier for all 2558 digits was submitted as input to PE.

The resulting embeddings allow us to predict the classifiers' patterns of confusions, calculated based on the true labels for all 2558 objects. Fig.5 shows embeddings for both 10 labels/class and 100 labels/class. In both cases we see five clouds of points corresponding to the five classes. The clouds are elongated and oriented roughly towards a common center, forming a star shape (also seen to some extent in our other applications). Objects that concentrate their probability on only one class will lie as far from the center of the plot as possible – ideally, even farther than the mean of their class, because this maximizes their posterior probability on that class. Moving towards the center of the plot, objects become increasingly confused with other classes.

Relative to using only 10 labels/class, using 100 labels yields clusters that are more distinct, reflecting better between-class discrimination. Also, the labeled examples are more evenly spread through each cluster, reflecting more faithful within-class models and less overfitting. In both cases, the '1' class is much closer than any other to the center of the plot, reflecting the fact that instances of other classes tend to be mistaken for '1's. Instances of other classes near the '1' center also tend to look rather "one-like" – thinner and more elongated. The dense cluster of points just outside the mean for '1' reflects the fact that '1's are rarely mistaken for other digits. In Fig.5(a), the '0' and '3' distributions are particularly overlapping, reflecting that those two digits are most readily confused with each other (apart from 1). The 'webbing' between the diffuse '2' arm and the tighter '3' arm reflects the large

number of '2's taken for '3's. In Fig.5(b), that 'webbing' persists, consistent with the observation that (again, apart from many mistaken responses of 1) the confusion of '2's for '3's is the only large-scale error these larger data permit.

7 Visualization of latent structure of unlabeled data

In the applications above, PE was applied to visualize the structure of classes based at least to some degree on labeled examples. The algorithm can also be used in a completely unsupervised setting, to visualize the structure of a probabilistic generative model based on latent classes. Here we illustrate this application of PE by visualizing a semantic space of word meanings: objects correspond to words, and classes correspond to topics in a latent Dirichlet allocation (LDA) model (Blei et al., 2003) fit to a large (>37,000 documents, >12,000,000 word tokens) corpus of educational materials for first grade to college (TASA). The problem of mapping a large vocabulary is particularly challenging, and, with over 26,000 objects (word types), prohibitively expensive for pairwise methods. Again, PE solves for the configuration shown in about a minute.

In LDA (not to be confused with FLDA above), each topic defines a probability distribution over word types that can occur in a document. This model can be inverted to give the probability that topic c_k was responsible for generating word x_n; these probabilities $p(c_k|\boldsymbol{x}_n)$ provide the input needed to construct a space of word and topic meanings in PE (see Appendix E).

More specifically, we fit a 50-topic LDA model to the TASA corpus. Then, for each word type, we computed its posterior distribution restricted to a subset of 5 topics, and input these conditional probabilities to PE (with $N = 26,243$, $K = 5$). Fig.6 shows the resulting embedding. As with the embeddings in Figs. 1 and 2, the topics are arranged roughly in a star shape, with a tight cluster of points at each corner of the star corresponding to words that place almost all of their probability mass on that topic. Semantically, the words in these extreme clusters often (though not always) have a fairly specialized meaning particular to the nearest topic. Moving towards the center of the plot, words take on increasingly general meanings.

This embedding shows other structures not visible in previous figures:

in particular, dense curves of points connecting every pair of clusters. This pattern reflects the characteristic probabilistic structure of topic models of semantics: in addition to the clusters of words that associate with just one topic, there are many words that associate with just two topics, or just three, and so on. The dense curves in Fig.6 show that for any pair of topics in this corpus, there exists a substantial subset of words that associate with just those topics. For words with probability sharply concentrated on two topics, points along these curves minimize the sum of the KL and regularization terms. This kind of distribution is intrinsically high-dimensional and cannot be captured with complete fidelity in any 2-dimensional embedding.

As shown by the examples labeled in Fig.6, points along the curves connecting two apparently unrelated topics often have multiple meanings or senses that join them to each topic: 'deposit' has both a geological and a financial sense, 'phase' has both an everyday and a chemical sense, and so on.

8 Conclusions and future works

We have proposed a probabilistic embedding method, PE, that embeds objects and classes simultaneously. PE takes as input a probability distribution for objects over classes, or more generally of one set of points over another set, and attempts to fit that distribution with a simple class-conditional parametric mixture in the embedding space. Computationally, PE is inexpensive relative to methods based on similarities or distances between all pairs of objects, and converges quickly on many thousands of data points.

The visualization results of PE shed light on features of both the data set and the classification model used to generate the input conditional probabilities, as shown in applications to classified web pages, partially classified digits, and the latent topics discovered by an unsupervised method, LDA. PE may also prove useful for similarity-preserving dimension reduction, where the high-dimensional model is not of primary interest, or more generally, in analysis of large conditional probability tables that arise in a range of applied domains.

As an example of an application we have not yet explored, purchases, web-surfing histories, and other preference data naturally form distributions over items or categories of items. Conversely, items define distributions over people or categories thereof. Instances of such dyadic data abound–

restaurants and patrons, readers and books, authors and publications, species and foods...–with patterns that might be visualized. PE provides a tractable, principled, and effective visualization method for large volumes of such data, for which pairwise methods are not appropriate.

A Gradients

This appendix describes gradients of objective function E w.r.t. $\boldsymbol{r}_n$ and $\boldsymbol{\phi}_k$. We rewrite the objective function (2) as follows:

$$\begin{aligned}
& E(\boldsymbol{R}, \boldsymbol{\Phi}) \\
= & -\sum_{n=1}^{N}\sum_{k=1}^{K} p(c_k|\boldsymbol{x}_n) \log p(c_k|\boldsymbol{r}_n) \\
= & -\sum_{n=1}^{N}\sum_{k=1}^{K} p(c_k|\boldsymbol{x}_n) \left(\log p(c_k) - \frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2 - \log \sum_{l=1}^{K} p(c_l) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_l\|) \right) \\
= & -\sum_{n=1}^{N} \left(\sum_{k=1}^{K} p(c_k|\boldsymbol{x}_n) \log p(c_k) - \frac{1}{2}\sum_{k=1}^{K} p(c_k|\boldsymbol{x}_n)\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2 - \log \sum_{k=1}^{K} p(c_k) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2) \right) \quad (13)
\end{aligned}$$

Differentiating (13) w.r.t. $\boldsymbol{r}_n$, we obtain:

$$\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{r}_n} &= \sum_{k=1}^{K} p(c_k|\boldsymbol{x}_n)(\boldsymbol{r}_n - \boldsymbol{\phi}_k) - \frac{\sum_{k=1}^{K}(\boldsymbol{r}_n - \boldsymbol{\phi}_k)p(c_k)\exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2)}{\sum_{k=1}^{K} p(c_k)\exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2)} \\
&= \sum_{k=1}^{K} (p(c_k|\boldsymbol{x}_n) - p(c_k|\boldsymbol{r}_n))(\boldsymbol{r}_n - \boldsymbol{\phi}_k) \\
&= \sum_{k=1}^{K} (p(c_k|\boldsymbol{r}_n) - p(c_k|\boldsymbol{x}_n))\boldsymbol{\phi}_k. \quad (14)
\end{aligned}$$

Differentiating (13) w.r.t. $\boldsymbol{\phi}_k$, we obtain:

$$\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{\phi}_k} &= -\sum_{n=1}^{N} p(c_k|\boldsymbol{x}_n)(\boldsymbol{r}_n - \boldsymbol{\phi}_k) + \sum_{n=1}^{N} \frac{(\boldsymbol{r}_n - \boldsymbol{\phi}_k)p(c_k)\exp(\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2)}{\sum_{k=1}^{K} p(c_k)\exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2)} \\
&= \sum_{n=1}^{N} (p(c_k|\boldsymbol{x}_n) - p(c_k|\boldsymbol{r}_n))(\boldsymbol{\phi}_k - \boldsymbol{r}_n). \quad (15)
\end{aligned}$$

B Hessian

This appendix describe the Hessian of the objective function E w.r.t. $\boldsymbol{r}_n$. Differentiating (14) w.r.t. $\boldsymbol{r}_n^T$, we obtain:

$$\begin{aligned}
\frac{\partial^2 E}{\partial \boldsymbol{r}_n \partial \boldsymbol{r}_n^T} &= -\sum_{k=1}^{K} \boldsymbol{\phi}_k \left(\frac{(\boldsymbol{r}_n - \boldsymbol{\phi}_k)^T p(c_k) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2)}{\sum_{l=1}^{K} p(c_l) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_l\|^2)} \right. \\
&\quad \left. + \frac{p(c_k) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_k\|^2) \sum_{l=1}^{K} (\boldsymbol{r}_n - \boldsymbol{\phi}_l)^T p(c_l) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_l\|^2)}{\left(\sum_{l=1}^{K} p(c_l) \exp(-\frac{1}{2}\|\boldsymbol{r}_n - \boldsymbol{\phi}_l\|^2)\right)^2} \right) \\
&= -\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \boldsymbol{r}_n^T + \sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T \\
&\quad + \sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \boldsymbol{r}_n^T - \left(\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \right) \left(\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \right)^T \\
&= \sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T - \left(\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \right) \left(\sum_{k=1}^{K} p(c_k|\boldsymbol{r}_n) \boldsymbol{\phi}_k \right)^T . \qquad (16)
\end{aligned}$$

C Web page classifier

We explain here the naive Bayes model used for the estimation of conditional probabilities in Section 5. We assume that a web page (considered as a bag of words) $\boldsymbol{x}_n$ in class c_k is generated from a multinomial distribution as follows:

$$p(\boldsymbol{x}_n|c_k) \propto \prod_{j=1}^{V} {\theta_{kj}}^{x_{nj}}, \qquad (17)$$

where V is the number of word types, x_{nj} is the number of tokens of word type w_j in a page $\boldsymbol{x}_n$, θ_{kj} is the probability that a word token is of type w_j in a page of a class c_k ($\theta_{kj} > 0, \sum_{j=1}^{V} \theta_{kj} = 1$). We approximate θ_{kj} by its maximum a posteriori (MAP) estimate. The estimated θ_{kj} is

$$\hat{\theta}_{kj} = \frac{\sum_{n \in \boldsymbol{C}_k} x_{nj} + \lambda_k}{N_k + \lambda_k V}, \qquad (18)$$

where, N_k is the number of pages in a class c_k, $\boldsymbol{C}_k$ is a set of pages in a class c_k, λ_k is a hyper-parameter. We estimated λ_k by a leave-one-out cross validation method.

D Hand written digits classifier

The hand-written digits classifier discussed in Section 6 represents each class as a mixture of Gaussians. The mean of each component is a random sample from the class, and the covariance for each is the covariance for the entire data set:

$$p(\boldsymbol{x}_n|c_k) \propto \sum_{m \in \boldsymbol{C}_k} \exp\{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{x}_m)^T \Sigma^{-1} (\boldsymbol{x}_n - \boldsymbol{x}_m)\}, \tag{19}$$

where $\boldsymbol{x}_n$ is a 256 dimensional vector of pixel grey levels for a hand written digit, and $\boldsymbol{C}_k$ is the set of samples defining the model for class c_k.

E Latent topic model

The conditional probabilities of topics in Section 7 are from a Latent Dirichlet Allocation (LDA) model of a text corpus. In LDA, each document is assumed to be generated by a mixture of latent topics. The topic proportion vector for each document is drawn from a Dirichlet distribution. Each word is generated by first drawing a topic from this distribution, and then drawing a word from a topic-specific multinomial distribution. Let $\boldsymbol{x}$ be a document, w_m be the m-th word in the document, M be the number of words in the document, z_k be a latent topic, and K be the number of latent topics. The generative model of a document is as follows:

$$p(\boldsymbol{x}) = \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\psi}} \left(\prod_{m=1}^{M} \sum_{k=1}^{K} p(w_m|z_k, \boldsymbol{\psi}) p(z_k|\boldsymbol{\theta}) \right) p(\boldsymbol{\psi}|\boldsymbol{\beta}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) d\boldsymbol{\psi} d\boldsymbol{\theta}, \tag{20}$$

where $p(w_m|z_k, \boldsymbol{\psi})$, $p(z_k|\boldsymbol{\theta})$ are multinomial distributions and $p(\boldsymbol{\psi}|\boldsymbol{\beta})$, $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$ are Dirichlet distributions. We estimated ψ_{km} (the probability of word w_m given latent topic z_k) by Gibbs sampling (Gilks, Richardson, & Spiegelhalter, 1996), (Griffiths & Steyvers, 2004), and obtained conditional probabilities (probabilities of latent topics given word) by Bayesian inversion.

References

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, 2385–2404.

Belhumeur, P., Hespanha, P., & Kriegman, D. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 711–720.

Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 993–1022.

Fisher, R. (1950). The use of multiple measurements in taxonomic problem. *Annuals of Eugenics*, 179–188.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition.* New York: Academic Press.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov chain monte carlo in practice.* New York: Chapman & Hall.

Globerson, A., Chechik, G., Pereira, F., & Tishby, N. (2005). Euclidean embedding of co-occurrence data. *Advances in Neural Information Processing Systems 17*, 497–504.

Golub, G., & Van Loan, C. (1996). *Matrix computation 3rd edition.* Baltimore, Maryland: John Hopkins University Press.

Griffiths, T., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 5228–5235.

Hinton, G., & Roweis, S. (2002). Stochastic neighbor embedding. *Advances in Neural Information Processing Systems 15*, 833–840.

Jolliffe, I. (1980). *Principal component analysis.* New York: Springer-Verlag.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Proceedings of aaai workshop on learning for text categorization* (pp. 41–48). Madison: AAAI Press.

Mei, G., & Shelton, C. (2006). Visualization of collaborative data. In *Proceedings of international conference on uncertainty in artificial intelligence* (pp. 341–348).

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., & Muller, K. (1999). Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX*, 41–48.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by local linear embedding. *Science*, 2323–2326.

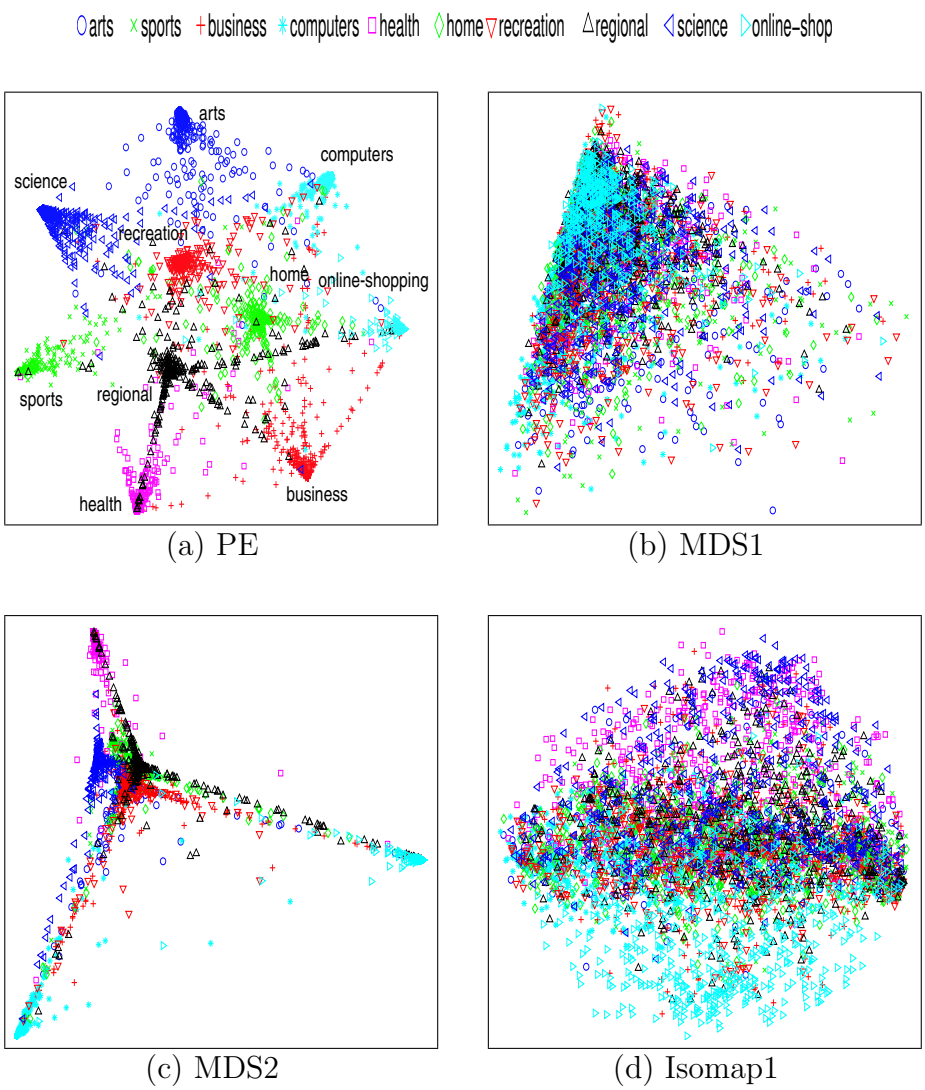
Saito, K., & Nakano, R. (1997). Partial bfgs update and efficient step-length calculation for three-layer neural networks. *Neural Computation*, 123–141.

Silva, V. de, & Tenenbaum, J. (2002). Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15*, 705–712.

Tenenbaum, J., Silva, V. de, & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 2319–2323.

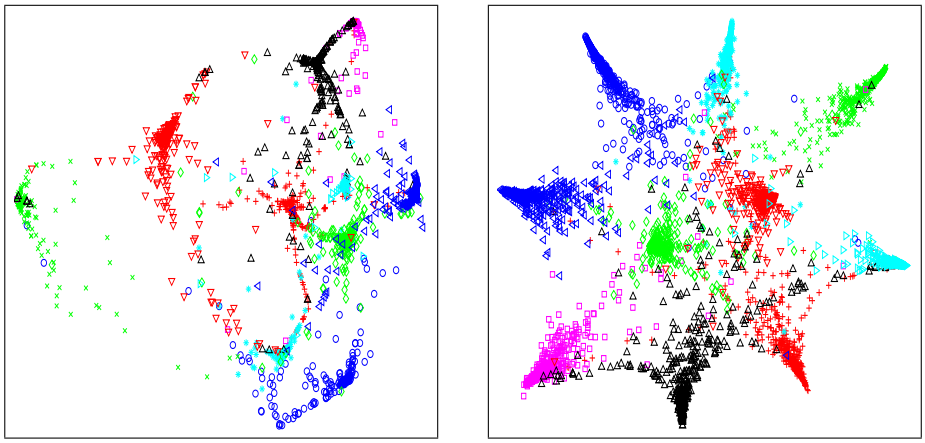
Torgerson, W. (1958). *Theory and methods of scaling*. New York: Wiley.

Yamada, T., Saito, K., & Ueda, N. (2003). Cross-entropy directed embedding of network data. In *Proceedings of international conference on machine learning* (pp. 832–839). Washington: AAAI Press.

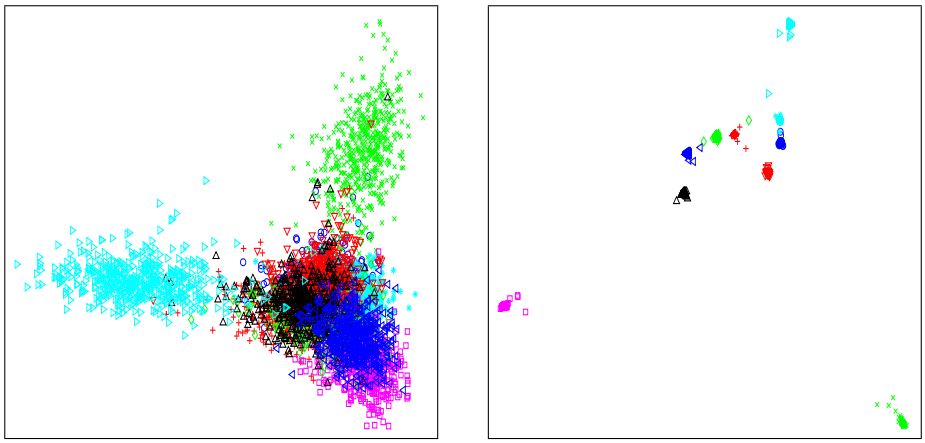


(a) PE (b) MDS1

(c) MDS2 (d) Isomap1



(e) Isomap2 (f) SNE



(g) FLDA (h) KDA

Figure 1: The visualizations of categorized web pages. Each of the 5000 web pages is show by a particle with shape indicating the page's class.

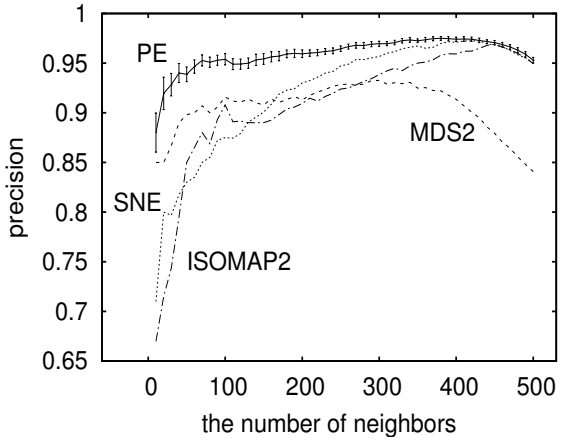


Figure 2: Experimental comparisons of the degree of conditional probability approximation. Each error bar of PE represents the standard deviation of 100 results with different initial conditions.

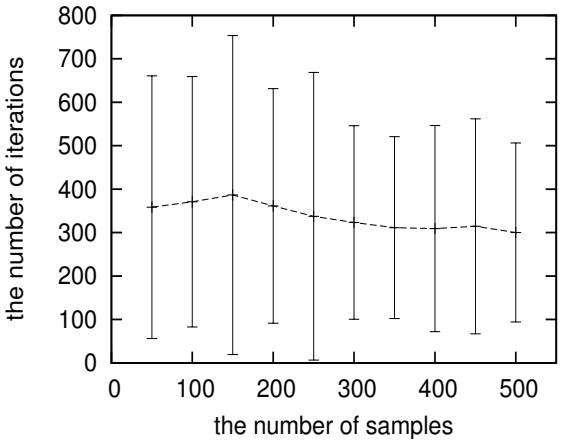


Figure 3: The number of iterations of PE with 10 classes. Each error bar represents the standard deviation of 1000 results with different initial conditions.

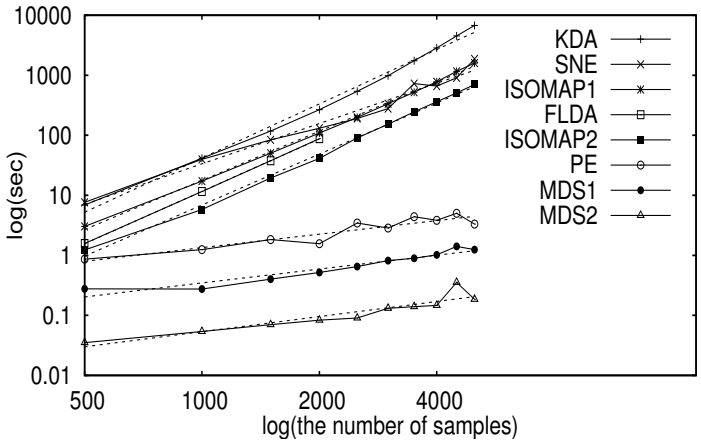
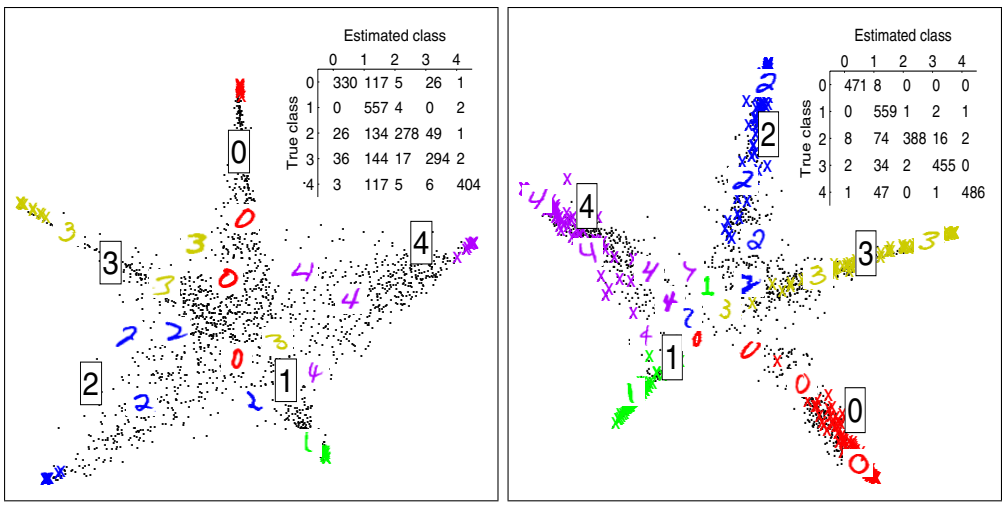


Figure 4: Experimental comparisons of the computational complexity



(a) 10 labels/class (b) 100 labels/class

Figure 5: Parametric embeddings for handwritten digit classification. Each dot represents the coordinates $\boldsymbol{r}_n$ of one image. Boxed numbers represent the class means $\boldsymbol{\phi}_k$. $\times$'s show labeled examples used to train the classifier. Images of several unlabeled digits are shown for each class.

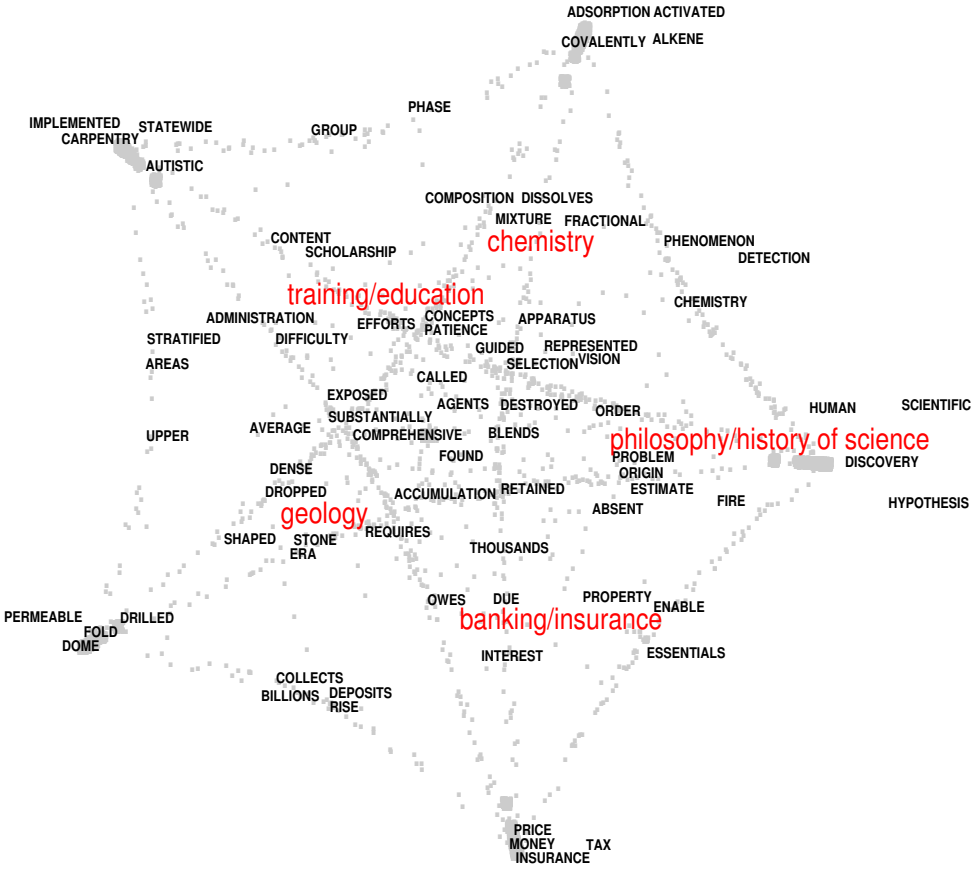


Figure 6: Parametric embedding for word meanings and topics based on posterior distributions from an LDA model. Each dot represents the coordinates $\boldsymbol{r}_n$ of one word. Large phrases indicate the positions of topic means $\boldsymbol{\phi}_k$ (with topics labeled intuitively). Examples of words that belong to one or more topics are also shown.