Elements of Generative Manifold Learning for semi-supervised tasks

Raúl Cruz and Alfredo Vellido

Departament de Llenguatges i Sistemes Informatics,
Universitat Politècnica de Catalunya

Abstract
For many real-world application problems, the availability of data labels for supervised learning is rather limited. It is often the case that a limited number of labelled cases is accompanied by a larger number of unlabeled ones. This is the setting for semi-supervised learning, in which unsupervised approaches assist the supervised problem and viceversa. In this report, we outline some basic theoretical foundations of semi-supervised learning using models of the generative manifold-learning family.

1 Introduction

Labeling aspects of reality seems to be one of the most standard occupations of the human brain and, therefore, of natural learning. When dividing the existing reality into different categories, we are seamlessly performing a classification task that can be improved over time through learning. In the realm of non-natural, or machine learning, the task of unravelling the relationship between the observed data and their corresponding class labels can be seen as the modelling of the mapping between a set of data inputs and a set of data targets. This is understood as supervised learning.

Unfortunately, in many real applications class labels are either completely or partially unavailable. The first case scenario is that of unsupervised learning, where the most common task to be performed is that of data clustering, which aims to discover the "true" group structure of multivariate data [18].

The second case is less frequently considered but far more common than what one might expect: quite often, only a reduced number of class labels is readily available and even that can be difficult and/or expensive to obtain. In such context, unsupervised models are an adequate tool for a first exploratory approach. The available class labels can then be used to refine the unsupervised procedure. This becomes a task on the interface between supervised and unsupervised models: semi-supervised learning [10]. This type of learning is commonly understood as a way to improve supervised tasks (usually with few available labelled samples) with the use of unlabeled samples ([31, 6, 19, 14, 27]). One can take a less typical approach: improving and refining unsupervised learning by using class labeled data.

The baseline method we will resort to in order to illustrate this approach is a generative constrained mixture model of the manifold learning family: namely, Generative Topographic Mapping ([33]]). This model has been quoted to be "a very powerful architecture in such situations, obtaining the latent manifold as a smooth nonlinear mapping of a uniform distribution over a low-dimensional space, represented by a regular grid" ([31]). This regular grid low-dimensional representation allows GTM to be used for the intuitive visualization of both the multivariate data and the obtained clustering results.

2 The Semi-Supervised Learning Problem in Pattern Recognition

This section introduces some of the basic concepts underlying the field of semi-supervised learning, within the general framework of Machine Learning. It must be noted from the onset that this research area is still far from fully established and standardized, and that quite different approaches to deal with it can be found in the recent academic literature. In what follows, we shall stick to the view provided by the Statistical Machine Learning field.

Modern Pattern Recognition has for long been well served by Machine Learning techniques, many of them widely applied and accepted. There are many ways to categorize these techniques; amongst them, we are interested in that which divides them between supervised and unsupervised, according to the availability of data labels to accompany the data observations. It is common knowledge that, in supervised Machine Learning, the aim is to learn a mapping from the observed input data to an output whose correct values, or target labels, are provided by a supervisor. In unsupervised learning, instead, there is no such supervisor, and only unlabeled observed input data are available. The aim in this case is to find regularities that might exist in the input data.

Semi-Supervised Learning (SSL) is an emergent discipline that incorporates prior knowledge into supervised or unsupervised methods (classification and clustering, mainly). The need for SSL, understood as learning from a combination of both labelled and unlabeled data, rises naturally in cases for which there exists a large supply of unlabeled data but a limited one of labelled data (bearing in mind that in many practical domains it can be very difficult and/or expensive to generate the labelled data). When SSL is used for classification, the main goal is to improve the classification accuracy aided by unlabeled data.

SSL for classification has become popular over the past few years. Some of the proposed methods include: co-training [6] (in which there are two kinds of information - about examples and the availability of both labelled and unlabelled data); transductive Support Vector Machines [19] (in which transduction follows Vapnik's principle: when trying to solve some problem, one should not solve a more difficult problem as an intermediate step); and Expectation-Maximization (EM), within the Maximum Likelihood framework, to incorporate unlabeled data into the training processes [14, 27].

In [31] this task is defined as follows: Given an unknown probabilistic relationship $P(x,t)$ between input points x and class labels $t \in T = \{1,...,c\}$, the problem is to predict t from x, i.e. to find a *predictor* $\hat{t} = \hat{t}(x)$ such that the generalization error of $\hat{t}$,

$$P_{x,t}\{\hat{t}(x) \neq t\}, \tag{1}$$

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

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

In unsupervised learning, one of the most widely used methods for data analysis is clustering. Clustering tries to group a set of points into clusters such that points in the same cluster are more similar to each other than to points in different clusters, under a particular cluster distortion or distance measure [18].

Semi-supervised clustering (SSC) uses class labels or pairwise constraints (specifying wether two instances should be in same or different clusters) on some examples to aid unsupervised clustering. SSC is useful when knowledge of the relevant categories of a problem is incomplete. When it happens, SSC can group data using the categories in the initial labeled data as well as extend and modify the existing set of categories as needed to reflect other regularities in the data.

Two general approaches for SSC can be found in existing methods [1], namely: constraint-based and distance-based methods. In the former, the clustering algorithm itself is modified so that the available labels or constraints are used to bias the search for an appropiate clustering of the data. In the latter approaches, an existing clustering algorithm that uses a distance measure is employed; however, the distance measure is first trained to satisfy the labels or constraints in the supervised data.

At the present time, there is a tendency to consider as “standard” SSL methods [10] only those which use it for classification tasks (as it is defined in [31]). However, SSC should be considered a more general SSL setting when the number and nature of the classes are not known in advance but have to be inferred from the data.

A problem related to SSL is transductive learning. Here a (labeled) training set and an (unlabeled) test set are provided. The idea of transduction is to perform predictions only for the test data.

2.1 Semi-supervised learning categories

SSL methods work on the basis of some assumptions, which allow a general classification of the different techniques [10]:

- The semi-supervised smoothness assumption: if two points x_1, x_2 in a high density region are close, then so should be the corresponding outputs y_1, y_2. This assumption implies that if two points are separated by a low density region, then their outputs need not be close to each other.

- The cluster assumption: if points are in the same cluster, they are likely to be of the same class. This can be equivalently formulated as a low density separation criterion: the decision boundary should lie in a low-density region. Both formulations are conceptually equivalent but can inspire different algorithms.

- The manifold assumption: the (high-dimensional) data lie (roughly) on a low-dimensional manifold. This assumption allows to avoid the curse of dimensionality in the sense that when data happen to lie on a low-dimensional manifold, the learning algorithm can essentially operate in a space of corresponding dimension.

- Vapnik's principle: when trying to solve some problem, one should not solve a more difficult problem as an intermediate step. Transduction follows this principle, in this kind of problems as in supervised learning we want to predict a set of labels y corresponding to some objects x. Transduction consists of directly estimating the finite set of test labels (a function $f : X_u \rightarrow Y$ only defined on the test set) instead of inferring a function $f : X \rightarrow Y$ on the entire space X as in inductive methods.

Following the assumptions mentioned above, the SSL methods can be classified as [10]: generative models, low-density separation, graph-based methods and change of representation.

Inference in generative models involves the estimation of the conditional density $P(x|y)$. In this way, any additional information on $p(x)$ is useful. The cluster assumption is implemented using these models since a given cluster is assumed belong to only one class. Knowledge of the structure of the problem or the data can naturally be incorporated to the model [26]. It is important to note, though, that unlabeled data can decrease prediction accuracy, when modeling assumptions are not correct [11].

The algorithms which try to implement the low-density separation assumption push the decision boundary away from the unlabeled points. To achieve this goal the most common method is Transductive Support Vector Machine (TSVM)[19]. The TSVM method maximizes the margin for unlabeled as well as for labeled points. Some alternatives to TSVM have been formulated in a probabilistic and in an information theoretic framework [24, 15].

In graph-based methods, the data are represented by the nodes of a graph, the edges of which are labeled with the pairwise distances of the incident nodes (and a missing edge corresponds to infinite distance). The way the distance between two points is computed can be seen as an approximation of the geodesic distance of the two points with respect to the manifold of data points [3]. Thus, the manifold assumption is the appropriate base to build graph methods. Usually some graph methods are transductive because of the prediction consists of labels for the unlabeled nodes, although recent work has extended graph-based methods to produce inductive solutions [32]. Directed graphs used for information propagation have also been researched in this field [9].

Change of representation includes algorithms that are not intrinsically semi-supervised, but instead perform two-step learning:

1. Perform an unsupervised step on all data, labeled and unlabeled, but ignoring the available labels.
2. Ignore the unlabeled data and perform plain supervised learning using the new distance, representation, or kernel built in step 1.

The semi-supervised smoothness assumption is implemented here since the representation is changed in such a way that small distances in high-density regions are conserved. Some graph-based methods are related to these algorithms since the construction of the graph from the data can be seen as an unsupervised change of representation [36, 30].

2.2 Semi-supervised Generative Models

2.2.1 Generative Models

The main thrust of this report concerns generative baseline methods, which we now describe within the SSL framework.

The basic problem consists on modelling a probability density function $p(x)$, given a finite number of data points x^n, $n = 1, ..., N$ drawn from that density function. From the alternative approaches widely known to face this problem stand out the parametric, non-parametric and semi-parametric methods [4].

In parametric methods, a specific functional form for the density model is assumed. The drawback of such an approach is that the particular form of parametric function chosen might be incapable of providing a good representation of the true density (model).

Instead, in non-parametric methods no particular functional form is assumed, and the form of the density is determined entirely by the data. The problem in these methods is that the number of parameters in the model grows with the size of the data set.

The best of both worlds is merged in the semi-parametric approach. Here, a very general class of functional forms is allowed, in which the number of adaptive parameters can be increased in a systematic way to build ever more flexible models, but where the total number of parameters in the model can be varied independently from the size of the data set.

The last approach is the one we are interested in. In particular, we focus on mixture of distributions models. In these models, the density function is formed from a linear combination of basis functions, where the number M of basis functions is treated as a parameter of the model and is typically much less than the number N of data points. Thus, the model for the density can be written as a linear combination of component densities $p(x|j)$ in the form

$$p(x) = \sum_{j=1}^{M} p(x|j)P(j). \tag{2}$$

This representation is called a *mixture distribution* ([35], [25]) and the coefficients $P(j)$ are called the *mixing parameters*. The next constraints should be satisfied by $P(j)$ (which is the prior probability of the data point having been generated from component j of the mixture)

$$\sum_{j=1}^{M} P(j) = 1 \tag{3}$$

$$0 \leq P(j) \leq 1. \tag{4}$$

In the same way, the component density functions $p(x|j)$ are normalized so that

$$\int p(x|j)dx = 1. \tag{5}$$

To generate a data point from the probability distribution (2), one of the components j is first selected at random with probability $P(j)$, and then a data point is generated from the corresponding component density $p(x|j)$.

The way $p(x|j)$ is computed depends on the type of distributions chosen for the individual component densities. For example, if Gaussian distributions are selected, then we say we are working with a Gaussian mixture model and $p(x|j)$ is computed as (assuming the Gaussians each have a covariance matrix $\sum_j = \sigma_j^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix):

$$p(x|j) = \frac{1}{(2\pi\sigma_j^2)^{d/2}} \exp\left\{-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right\} \tag{6}$$

A Maximum Likelihood approach is often used to determine the parameters of a (Gaussian or other) mixture model from a set of data. An elegant, practical and iterative procedure for estimating the mixture parameters is the Expectation-Maximization or EM algorithm ([13]).

These kind of generative models are the background for posterior sections in which we will consider generative methods.

2.2.2 Semi-supervision in Generative Models

In this section we describe the way in which a generative model can be seen as a semi-supervised method.

We can find a description of how a generative method can be used for semi-supervised learning tasks in [31], specially for classification ones. Within this context the class distributions $P(\mathbf{x}|y)$[1] can be modeled using model families $\{P(\mathbf{x}|y,\theta)\}$, and the class priors $P(y)$ by $\pi_y = P(y|\pi)$, $\pi = (\pi_y)_y$. An architecture of this type is referred to as a joint density model, since the full joint density $P(\mathbf{x}, y)$ is modeled by $\pi_y P(\mathbf{x}|y,\theta)$. For any fixed $\hat{\theta}$, $\hat{\pi}$, an estimate of $P(y|\mathbf{x})$ can be computed by Bayes' formula:

$$P(y|\mathbf{x},\hat{\theta},\hat{\pi}) = \frac{\hat{\pi}_y P(\mathbf{x}|y,\hat{\theta})}{\sum_{y'=1}^{M} \hat{\pi}_{y'} P(\mathbf{x}|y',\hat{\theta})}. \tag{7}$$

A model for the marginal $P(x)$ is

$$P(x|\theta,\pi) = \sum_{y=1}^{M} \pi_y P(\mathbf{x}|y,\theta). \tag{8}$$

If labeled and unlabeled data are available, a natural criterion emerges as the *joint log likelihood* of both D_l and D_u[2],

$$\sum_{i=1}^{n} \log \pi_{y_i} P(\mathbf{x}_i|y_i,\theta) + \sum_{i=n+1}^{n+m} \log \sum_{y=1}^{M} \pi_y P(\mathbf{x}_i|y,\theta), \tag{9}$$

It is straightforward to consider this as an issue of Maximum Likelihood in the presence of missing data (treating y as a latent variable), which can in principle be tackled by the EM algorithm, or alternative methods such as direct gradient descent.

Limitations of generative techniques in SSL

In summary, generative techniques use a model family $\{P(\mathbf{x}, y|\theta,\pi)\}$ in order to model the joint data distribution $P(x, y)$. These techniques use a mixture density estimation method for $P(x)$ on $X_l \cup X_u$, treating y as a latent class variable, then using the labeled sample D_l in order to associate latent classes

---

[1] y plays the role of j as in section 2.2.1

[2] D_l and D_u follow the corresponding definitions on section 2

with actual ones. A problem with this approach is that the labeling provided by the unsupervised method may be inconsistent with D_l, in which case the clustering should be modified to achieve such consistency. Another problem when following the aforementioned strategy is that, for classification problems, generative methods might not always provide good solutions. That is, the maximization of the joint likelihood of a finite sample (for example) does not necessarily lead to a small classification error, because depending on the model it might be possible to make the likelihood increase more by improving the fit of $P(x)$ instead of that of $P(y|x)$. Some recent work describing these limitations can be found in [7], [23], [20], and [28].

3 Theoretical foundations of Generative Manifold Learning

3.1 Introduction

The manifold learning problem can be expressed as the recovery of meaningful low-dimensional structures hidden in high-dimensional data. An example might be a set of pixel images of an individual's face observed under different pose and lighting conditions; the manifold learning task would consist on the identification of the underlying variables (angle of elevation, direction of light, etc.) given only the high-dimensional observed pixel image data [34].

Recent years have witnessed the rapid development of nonlinear manifold methods. Four main approaches can be distinguished:

The first one, based on projection methods, aims to find principal surfaces covering data-populated areas, such as principal curves [16] [21].

The second entails local and global embedding algorithms. Amongst the former, Locally Linear Embedding (LLE) [29] and Laplacian Eigenmaps [2], which focus on the local data neighbouring structure. Amongst the later, ISOMAP [34].

The third resorts to mutual information, which is a measurement of the differences of probability distribution between the observed and embedded spaces. Examples of these are Stochastic Nearest Neighbor [17] and Manifold Charting [8].

The fourth concerns generative models (GTM: [5]), and hypothesizes that observed data are generated from a low-dimensional latent space.

Manifold learning models can also be considered according to the machine learning task they are fit for: supervised or unsupervised.

In recent times, semi-supervised learning methods have made use of manifolds for classification tasks. Here the fact that the data lie on a submanifold embedded in a high-dimensional space as commented in section 2.1, is assumed. In addition, learning algorithms developed under this assumption avoid the ubiquitous curse of dimensionality problem because they essentially operate in a space of corresponding (low) dimension.

In [10], it is shown how several graph-based methods can be built under the manifold assumption. The main idea stemming for these methods is that the data are represented by the nodes of a graph (forming a manifold of data points) and the edges are labeled with the pairwise distances of the incident nodes. For example, in [3] the approach is that classification functions are naturally defined only on the submanifold in question rather than the total ambient space. The problem with this approach is that a relatively small amount of noise or a few outliers can change the results dramatically. There are other approaches that take the problem in different directions (see [10]).

Not all generative models for manifold learning concern supervised learning (e.g. [5], [34], [12]). The unsupervised problem is stated as follows.

Let Y be a d-dimensional domain contained in the Euclidean space $\mathbb{R}^d$, and let $f : Y \to \mathbb{R}^N$ be a smooth embedding, for some $N > d$. Data points $\{y_i\} \subset Y$ are generated by some random process, and are mapped by f to give the data observed, $\{x_i = f(y_i)\} \subset \mathbb{R}^N$. Y is referred as the latent space and $\{y_i\}$ as the latent data.

The task is to reconstruct f and $\{y_i\}$ from the observed data $\{x_i\}$ alone. In the next section we describe the Generative Topographic Mapping, mentioned in previous sections, as a model of this kind.

3.2 Generative Topographic Mapping

In this section we describe the Generative Topographic Mapping (GTM: [33, 5]) model.

The GTM is a generative non-linear latent variable model that, in its original definition, was intended for modelling continuous, intrinsically low-dimensional data distributions, embedded in high-dimensional spaces. It also provides a principled alternative to the self-organizing map (SOM:[22]) algorithm, resolving many of its associated theoretical problems. Like SOM, GTM is used for unsupervised clustering and visualization.

3.2.1 The standard GTM model

The GTM is a non-linear latent variable model of the manifold learning family defined as a mapping from a low dimensional latent space onto the multivariate space where observed data reside. The mapping is carried through by a number of basis functions generating a constrained mixture density distribution. It is defined as a generalized linear regression model:

$$\mathbf{y} = \phi(\mathbf{u})\mathbf{W} \tag{10}$$

where ϕ are M basis functions $\phi(\mathbf{u}) = (\phi_{\mathbf{1}}(\mathbf{u}), ..., \phi_{\mathbf{M}}(\mathbf{u}))$. For continuous data of dimension D, spherically symmetric Gaussians

$$\phi_m(u) = \exp\left\{-1/2\sigma^2 \|u - \mu_m\|^2\right\} \tag{11}$$

are an obvious choice of basis function, with centres μ_m and common width σ; $\mathbf{W}$ is a matrix of adaptive weights w_{md} that defines the mapping, and $\mathbf{u}$ is a point in latent space. To avoid computational intractability a regular grid of K points $\mathbf{u}_k$ can be sampled from the latent space. Each of them, which can be considered as the representative of a data cluster, has a fixed prior probability $p(\mathbf{u}_k) = 1/K$ and is mapped, using Eq. 10, into a low dimensional manifold non-linearly embedded in the data space. This latent space grid is similar in design and purpose to that of the visualization space of the SOM. A probability distribution for the multivariate data $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$ can then be defined, leading to the following expression for the log-likelihood:

$$L(\mathbf{W}, \beta|\mathbf{X}) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{K} \sum_{k=1}^{K} \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\beta/2\|y_k - \mathbf{x}_n\|^2\right\} \right\} \tag{12}$$

where y_k, usually known as *reference* or *prototype vectors*, are obtained for each $\mathbf{u}_k$ using Eq. 10; and β is the inverse of the noise variance, which accounts for the fact that data points might not strictly lie on the low dimensional embedded manifold generated by the GTM. The EM algorithm is an straightforward alternative to obtain the Maximum Likelihood (ML) estimates of the adaptive parameters of the model, namely $\mathbf{W}$ and β.

3.2.2 Visualization using GTM

The interpretation of clustering results usually requires a drastic reduction of the dimensionality of the data. Latent variable models can provide such interpretation through visualization, as they describe the multivariate data in intrinsically low-dimensional spaces. The GTM was originally defined as an alternative to the SOM, defined within a probabilistic framework. As a result, the data visualization capabilities of the latter are fully preserved and even augmented by the former. The main advantage of GTM and any of its extensions over general finite mixture models consists precisely on the fact that both data and results can be intuitively visualized on a low dimensional representation space.

Each of the cluster representatives $\mathbf{u}_k$ in the latent visualization space is mapped, following Eq. 10, into a point y_k belonging to a manifold embedded in data space. Given that the posterior probability of every GTM cluster representative for being the generator of each data point $\mathbf{x}_n$ can be calculated, using Bayes' theorem, in the expectation step of the EM algorithm (as the expected value taken by an auxiliary term z_{kn} expressing our initial ignorance of which cluster k is responsible for generating each data point n), both data points and cluster prototypes can be visualized as a function of the latent point locations. The assignment of a probability of cluster membership to each data point n is a neat improvement on the SOM sharp map unit membership attribution for each data point, and leads to 2-dimensional representations of each multivariate data point in the form of the mean of the posterior distribution, or estimated responsibility $\hat{z}_{kn}$

$$u_n^{mean} = \sum_{k=1}^{K} \mathbf{u}_k \hat{z}_{kn}, \tag{13}$$

or in the form of attributions to the latent space locations bearing maximum responsibility:

$$u_n^{maxresp} = \arg\max_{u_k} \hat{z}_{kn}. \tag{14}$$

References

[1] S. Basu. *Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments.* PhD thesis, The University of Texas at Austin, 2005.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56:209–239, 2004.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[5] C. M. Bishop, M. Svensén, and C. K. I. Williams. The Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998.

[6] A. Blum and T. Mitchell. Learning from labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998.

[7] G. Bouchard and B. Triggs. The trade-off between generative and discriminative classifiers. In *IASC 16th International Symposium on Computational Statistics*, pages 721–728, 2004.

[8] M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems*, pages 857–864, 2003.

[9] C. J. C. Burges and J. C. Platt. Semi-supervised learning with conditional harmonic mixing. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*. The MIT Press, 2006.

[10] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006.

[11] F. Cozman and I. Cohen. Risks of semi-supervised learning. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*. The MIT Press, 2006.

[12] V. de Silva and J. Tenenbaum. Unsupervised learning of curved manifolds. In D. Denison, M. Hansen, C. Holmes, B. Mallick, and B. Yu, editors, *Nonlinear Estimation and Classification, Lecture Notes in Statistics*, volume 171, pages 453–466. Springer Verlag, New York, 2003.

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

[14] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via the EM approach. In *Advances in Neural Information Processing Systems*, number 6, pages 120–127, 1994.

[15] Y. Grandvalet and Y. Bengio. Entropy regularization. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*. The MIT Press, 2006.

[16] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1988.

[17] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, pages 857–864, 2003.

[18] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, New Jersey, 1998.

[19] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, pages 200–209, 1999.

[20] S. Kaski, J. Sinkkonen, and A. Klami. Discriminative clustering. *Neurocomputing*, 69:18–41, 2005.

[21] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):281–297, 2000.

[22] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[23] J. Lasserre, C. M. Bishop, and T. Minka. Principled hybrids of generative and discriminative models. In *Proceedings 2006 IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[24] N. D. Lawrence and M. I. Jordan. Gaussian processes and the null-category noise model. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning.* The MIT Press, 2006.

[25] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering.* New York: Marcel Dekker, 1988.

[26] K. Nigam, A. McCallum, and T. Mitchell. Semi-supervised text classification using EM. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning.* The MIT Press, 2006.

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

[28] J. Peltonen, A. Klami, and S. Kaski. Improved learning of Riemannian metrics for exploratory analysis. *Neural Networks*, 17:1087–1100, 2004. Invited paper.

[29] S. T. Roweis and K. S. Lawrence. Nonlinear dimensionality reduction by locally linear embedding. *Science*, (290):2323–2326, 2000.

[30] L. K. Saul, K. Q. Weinberger, F. Sha, J. Ham, and D. D. Lee. Spectral methods for dimensionality reduction. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning.* The MIT Press, 2006.

[31] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for ANC, Edinburgh, UK, 2000.

[32] V. Sindhwani, M. Belkin, and P. Niyogi. The geometric basis of semi-supervised learning. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*. The MIT Press, 2006.

[33] M. Svensén. *GTM: The Generative Topographic Mapping*. PhD thesis, Aston University, 1998.

[34] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[35] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. New York: John Wiley, 1985.

[36] X. Zhu, J. Kandola, J. Lafferty, and Z. Ghahramani. Graph kernels by spectral transforms. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*. The MIT Press, 2006.