Mixture Models for Clustering and Dimension Reduction
Verbeek, J.J.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
CONCLUSION AND DISCUSSION

Clustering and dimension reduction methods can be used to find a more compact representation of high dimensional data. In this compact representation, the data are described on the basis of cluster membership and/or by a low dimensional vector obtained by projecting the data on a low dimensional manifold. These methods are useful tools for data visualization, and pre-processing data in classification or regression applications in order to avoid the curse of dimensionality.

In this thesis we studied probabilistic mixture models, which provide a versatile framework for clustering and dimension reduction. By using different component distributions, the framework can be applied to many different types of data. Furthermore, parameter estimation, missing values in the data, and model selection can all be treated within a single and formally sound framework. We presented several contributions which we hope increase the practical applicability of mixture models for clustering and dimension reduction problems. Below we summarize the conclusions drawn in the previous chapters and outline directions for further research.

6.1 Summary of conclusions

Many, if not all, mixture model based clustering and dimension reduction techniques suffer from the problem that the objective function that is optimized exhibits several local optima. Standard parameter estimation techniques are hindered by these local optima, since they may return parameter estimates which are locally optimal but far from globally optimal. In Section 3.2 we presented a greedy parameter estimation scheme, that iteratively (i) adds components to the mixture and (ii) re-estimates the parameters of the mixture obtained so far with the EM algorithm. In Section 3.3 we presented a similar greedy scheme for k-means clustering. Our experimental results, obtained using synthetic and natural data, show that the greedy approach yields equal or better clusterings (in terms of the objective function) than alternative techniques. An additional benefit of the greedy approach is that a sequence of mixture models with an increasing
number of components is generated, which is useful when the number of clusters has to be determined as well.

In Section 3.4 we presented a constrained EM algorithm to accelerate parameter estimation for Gaussian mixtures from large data sets. Other existing acceleration techniques allow a limited freedom in setting a trade-off between accuracy and speed-up of each EM step. For very large speed-up and small accuracy the algorithms are not guaranteed to converge to (in a particular sense) locally optimal parameter estimates. In contrast, our algorithm converges to (locally) optimal parameter values for any trade-off between accuracy and speed, and also allows other speed-up techniques (e.g. based on geometric reasoning) to be plugged-in.

In Chapter 4 we presented a self-organizing map approach based on mixture models; the parameters are estimated by a constrained EM algorithm. The advantage of our mixture model based approach over Kohonen’s original self-organizing map is that the learning algorithm is guaranteed to converge and can be interpreted as maximizing an objective function. Moreover, the objective function can be augmented such that data with missing values can also be used for parameter estimation. Since self-organization is achieved by a simple modification of the standard EM algorithm for mixture models, our approach is readily applied to any mixture model for which the standard EM algorithm is available. A priori domain knowledge can be used to select an appropriate class of component densities. Therefore, it is relatively easy to apply this method to data which is not given as a set of vectors of real numbers.

In Chapter 5 we considered the coordinated factor analysis (CFA) model which uses a mixture of linear Gaussian latent variable models for non-linear dimension reduction. We presented an improvement of the original parameter estimation algorithm that is faster and leads to more accurate parameter estimates. We experimentally compared the performance of this method with generative topographic mapping and the self-organizing map approach of Chapter 4. The experimental results show that the CFA model needs about half the number of mixture components needed by the other approaches to reconstruct the data from the latent representation with a given accuracy. This is due to the fact that the CFA approach use a continuous latent representation, where the other approaches use a discrete latent representation.

In Section 5.3 we applied the CFA model to predict high dimensional correspondences. In this setting two data sets are given, each sampled from a different high dimensional embedding of the same low dimensional manifold. In addition, some correspondences are given: for some data points in the first set it is known that they share the same low dimensional coordinate on the manifold as a point in the second set. The goal is to predict, for points without a given correspondence, the coordinates of the corresponding point in the other set. This problem can be regarded as a missing value problem, which can be solved without constructing a global low dimensional latent representation, e.g. using a mixture of factor analyzers (MFA) which uses several local linear low dimensional representations to predict the correspondences. With our experiments we
demonstrated that if only few correspondences are given, then the CFA models yield more accurate predictions than the MFA models. This is explained by the fact that the MFA models lack a global low dimensional representation as used by the CFA models.

6.2 Discussion of directions for further research

Combining spectral methods and parametric models. A drawback of mixture models is the fact that parameter estimation techniques may return estimates that are locally, but not globally optimal. In principle, sampling based global optimization methods can be used, like simulated annealing, but these require an impractically long sequence of samples. In Section 3.2 and Section 3.3 we presented algorithms that avoid the local optima better than alternative algorithms, but in general there is no efficient algorithm available which is guaranteed to identify the globally optimal parameters.

In recent years there has been considerable interest in spectral methods for clustering and dimension reduction. Spectral methods deliver a clustering or low dimensional representation of the given data, but not a parametric model that can be used to map new data to a cluster index or low dimensional coordinate. The main attraction of these methods is that they minimize a quadratic function with a single global minimum that can be efficiently identified. Most of these methods involve a small number of parameters which have to be set, such as the number of neighbors \( k \) in nearest neighbor based methods. Since they are few in number, global optimization of these parameters is easier than finding a parametric model with many parameters of the high dimensional data.

To generalize the clustering or low dimensional representation produced by spectral methods to new data, all training data has to be retained and often nearest neighbors of the new data have to be found in the original data. Both storing the original data and finding nearest neighbors is undesirable when dealing with large data sets. In Chapter 5 we initialized the estimates of hidden variables in the CFA model with the solutions of spectral methods. Further research is needed to explore further possibilities to combine the benefits of spectral methods and methods that deliver a parametric model that can be applied to new data. For example, spectral methods could be used to define a distribution over plausible clusterings or latent coordinates. Parametric model learning can then be biased toward such plausible solutions rather than just be initialized by the solution of the spectral method.

Robust dimension reduction. In Chapter 5 we used mixtures of linear Gaussian latent variable models for non-linear dimension reduction. Gaussian densities are attractive from a computational point of view, and are also preferred as a default density in the absence of prior knowledge which suggests another, more appropriate, class of densities. However, the short tails of the Gaussian density make parameter estimation relatively
sensitive to outliers in the data. Outliers are data points which are very different from all other data points and that would be very unlikely under a distribution with parameters estimated from the other data points. Outliers can be caused by noise in measurement systems. Mixtures of linear latent variable models based on t-distributions have been proposed (de Ridder and Franc, 2003) to obtain more robustness against outliers. It would be interesting to consider how such mixtures of t-distributions could be used to form global non-linear models as in the CFA model.

**Semi-supervised learning.** As discussed in the introduction, unsupervised clustering and dimension reduction techniques can be used to find compact data representations to overcome the curse of dimensionality in classification and regression problems. Data processing then proceeds in two steps. First, a suitable compact data representation is determined using unsupervised examples. Second, the supervised data is analyzed in the new representation to find an appropriate classification or regression function. Recently semi-supervised learning\(^1\) approaches have been introduced that integrate these two steps by directly learning a classification or regression function in the high dimensional space from both supervised and unsupervised data. Loosely speaking, these methods avoid the curse of dimensionality by using unsupervised data to reduce the set of possible functions. The power of the semi-supervised approach lies in the fact that both the supervised and unsupervised data are used to reduce the set of functions.

Various semi-supervised learning approaches have been proposed (Baluja, 1998; Blum and Mitchell, 1998; Nigam et al., 2000; Szummer and Jaakkola, 2002; Zhu et al., 2003). Some of these use nearest neighbor graphs, to encode the smoothness assumption the class label (or regression variable) tends to be the same or similar for nearby points in the high dimensional data space. These methods can be implemented efficiently, have few parameters that have to be estimated and perform very well in practice. However, to evaluate the predictive density all training data needs to be accessed and thus stored. Nearest neighbor based methods can also be used to define a distribution over the class labels of both the supervised and unsupervised data. By conditioning on the known class labels of the supervised data we obtain a distribution over class labels of the unsupervised data. It is an interesting possibility to define such a conditional distribution on the class labels of the unsupervised data, and then to learn a parametric model from supervised and unsupervised data that—in expectation with respect to the uncertain class labels of the unsupervised data—optimally predicts the class labels.

\(^1\) Unfortunately, the term semi-supervised learning is sometimes used to refer to reinforcement learning, e.g. in (Arbib, 1995), which is quite different from the setup considered here.