Mixtures in Supervised Classification

Brian D. Ripley

Professor of Applied Statistics
University of Oxford

ripley@stats.ox.ac.uk
http://stats.ox.ac.uk/∼ripley

Overview

Report of work in progress in my group, with contributions from Rafael Perera-Salazar and (especially) Matthew Stephens.

- Why this is much suggested but little-used
- Ideas on approximations to the Bayesian ideal
- Connections to smoothing
- Light shed on unsupervised problems?

The message is a series of unexpected connections.

Main background reference is:

Ripley, B.D. (1996) *Pattern Recognition and Neural Networks.* Cambridge University Press.

Classification Setup

Suppose we have K predetermined classes and a set $\mathcal{T}$ of some examples x_i with (correctly) labelled classes. The task is to label future examples x, or to give the posterior probabilities

$$p(c \mid x), c = 1, \ldots, K$$

In the mixture approach we model the distribution $p_c(x)$ within each class by a finite mixture,

$$p_c(x) = \sum_{j=1}^{k_c} \omega_{cj} \; \phi(x; \theta_{cj})$$

and also give the prior probabilities of each class π_c. Then the distribution $p(x)$ of a future example is a mixture of mixtures, and

$$p(c \mid x) = \frac{\sum_{j=1}^{k_c} \omega_{cj} \; \phi(x; \theta_{cj})}{\sum_i \sum_{j=1}^{k_i} \omega_{ij} \; \phi(x; \theta_{ij})} \qquad (1)$$

Assume that the components are (multivariate) normal distributions; t's has been tried.

Then we have as special cases

- Linear discriminant analysis
  (one component per class,
  common variance matrix)
- Quadratic discriminant analysis
  (one component per class)
- 'Mixture discriminant analysis'
  (common variance for *all* components:
  Hastie & Tibshirani, JRSSB 1996)
- Normalized Radial Basis Function 'neural networks'.

Fitting

We do need to fit the model (1) by choosing the parameters

$$\theta = \big((k_c), (\pi_c), (\omega_{cj}, \theta_{cj})\big)$$

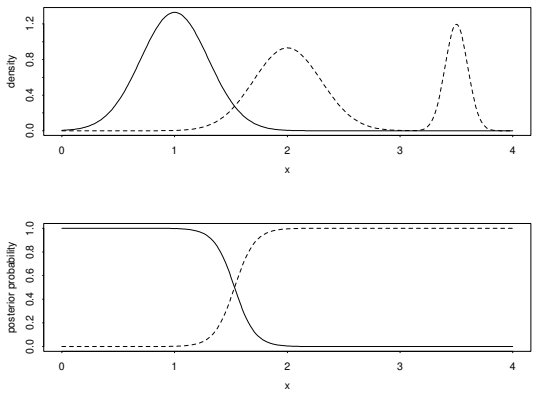
The parameters are not identifiable.

The first three approaches use the joint likelihood

$$\prod_s \pi_{c_s} p_{c_s}(x_s; \theta) = \prod_s p(c_s \mid x_s; \theta)\, p(x_s; \theta)$$

Sometimes known as the *sampling paradigm*. The alternative (normalized RBFs) is to condition on (x_i) and fit $p(c \mid x)$ directly using a conditional likelihood $\prod_j p(c_j \mid x_j; \theta)$, the *diagnostic paradigm*.

The difference can be dramatic! In the sampling paradigm *un*labelled observations contribute to the information about $p(c \mid x)$. This includes future examples presented to be classified. However, the sampling paradigm has many almost ineffective parameters.



It is not always necessary to model the class-conditional densities (upper figure) accurately, as the posterior probabilities in the lower figure are effectively unchanged by most aspects of modelling the right peak of the class-conditional density shown dashed. Only the densities in the interval $[1, 2]$ matter.

Maximum likelihood fitting

There are many problems with using (1) by 'pluging-in' MLEs.

- The likelihood is usually unbounded.
- There are many local maxima,
  and the usual EM 'algorithm' normally gets trapped far from a global maximum.
- Number of components?

A comparison using maximum likelihood fitting by Mette Langaas (Dr.Ing. thesis, NTH Trondheim) found the conditional approach more workable.

Bayesian approaches

One of the difficulties in the ML approach is that the components are unconstrained. I would expect components for one class to be similar, and those between classes to be similar (but less so than within a class) .... This suggests a hierarchical specification of a prior. Then the predictive approach is to use

$$\widetilde{p}(c \mid x) = \int p(c \mid x; \theta)\, p(\theta \mid \mathcal{T}, x)\, \mathrm{d}\theta \qquad (2)$$

The difficulty is to do the integration in (2). Note that θ includes the numbers (k_c) of components.

Approaches:

- MAP: use the peak of the integrand.
  (Also penalized likelihood approach.)
- MCMC: replace the integration by an average over samples from $p(\theta \mid \mathcal{T}, x)$.
- Approximations based on replacing $p(\theta \mid \mathcal{T})$ by a Gaussian mixture.

The MCMC approach has been taken by Diebolt & Robert (e.g. JRSSB 1994), Richardson & Green (JRSSB 1997) and others. It is not easy to find a representative set of samples from $p(\theta \mid \mathcal{T})$. We want $p(\theta \mid \mathcal{T}, x)$, so write

$$\widetilde{p}(c \mid x) \propto \int p(c \mid x; \theta)\, p_c(x; \theta) p(\theta \mid \mathcal{T})\, \mathrm{d}\theta$$

and sum over c to find the normalizing constant.

We have also found problems with averaging over the number of components.

Updating schemes

The difference between $p(\theta \mid \mathcal{T})$ and $p(\theta \mid \mathcal{T}, x)$ is probably negligible for one example. Suppose we originally had 30 classified examples and have now seen 3 000 examples to classify? The MCMC approach has to be re-run as each new example is presented, and all the data must be retained.

Matthew Stephens has worked on sequential updating schemes (with N) based on mixture approximations to

$$p(\theta \mid \mathcal{T}, x_i, i = 1, \ldots, N)$$

which seem very accurate. These could also be used to approximate $p(\theta \mid \mathcal{T})$.

The essence is that each additional observation makes the posterior into a mixture with an increased number of components, and the mixture is carefully collapsed to about 30 components at each stage (by minimizing KL divergence).

Connections to smoothing

Suppose we replace the K classes by a continuum. That is, we want to model a density $p(y \mid x)$ at all points of a space $\mathcal{X}$. Again we might use a mixture, now

$$p(y \mid x) = \sum_{j=1}^{k} \pi_j(x)\, \phi(y; \theta(x))$$

This has been suggested several times in the neural network literature, and we used it for multiple imputation in missing data problems, so y are the missing features, and x are the observed features.

Obviously we will want the conditional density to vary smoothly with x, which we do by taking $\pi_j(x)$ and $\theta(x)$ to be smooth functions. We need a high-dimensional smooth function of a high-dimensional argument, and used a neural network, with MAP fitting. (The prior imposes the smoothness.)

Connections to clustering

Let us revert to the classification problem, but take one component per class, and suppose we have a very small training set $\mathcal{T}$ and a very large set of unclassified observations. We will then have very precise knowledge of $p(x)$, but rather imprecise knowledge of $p(c \mid x)$.

If we had no training examples, $p(c \mid x) \equiv 1/K$. We have a complete lack of identifiability. However, this is not realistic; at least a few examples are labelled in most real clustering problems.

We may have very vague knowledge about $p(c \mid x)$, but we often have quite precise information about $p(c_1 = c_2 \mid x_1, x_2)$, ... Matthew Stephens' idea is to break the (near-)symmetry just as we did for the posterior $p(\theta \mid \mathcal{T})$, to summarize the joint distribution of the labels about just one of the many symmetries.