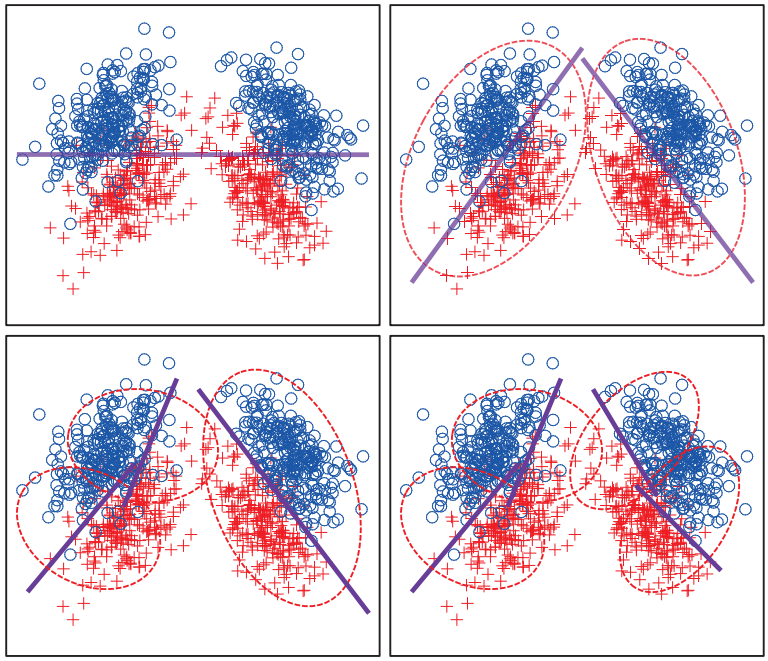
University of Cambridge Engineering Part IIB

Module 4F10: Statistical Pattern Processing

Handout 5: Mixtures and Products of Experts



Mark Gales
mjfg@eng.cam.ac.uk
Michaelmas 2015

Introduction

In the previous lectures generative models with Gaussian and Gaussian mixture model class-conditional PDFs were discussed. For mixtures models a linear combination of multivariate Gaussian distributions were used, where the weights (priors) used to combine the likelihoods were trained and fixed. In this lecture more general forms of combining information (distributions/classifiers) together will be discussed. Rather than referring to component distributions (as in the GMM) more general experts will be discussed. Two variations this form of model are possible: mixtures and products of experts.

In mixtures of experts (MoEs) the weights used to combine the experts is a function, the gating function, of the observation. Thus, the weight can vary from observation to observation. It is possible to combine classifier outputs from discriminative and generative models.

The second alternative is a product of experts (PoEs). Here rather than adding likelihood values, the values of the likelihoods are producted together and then normalised to yield the final values. This can be related to taking a geometric mean rather than an arithmetic mean.

Both these approaches will be discussed in this lecture.

Mixture of Experts

From the previous lecture the general form for the Mixture Model was

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} p(\boldsymbol{x}, \tilde{\omega}_m|\boldsymbol{\theta}_m) = \sum_{m=1}^{M} c_m p(\boldsymbol{x}|\tilde{\omega}_m, \boldsymbol{\theta}_m)$$

where $\tilde{\omega}_m$ indicates the component

$$c_m = P(\tilde{\omega}_m)$$

The equivalent general form for the Mixture of Experts for classification is

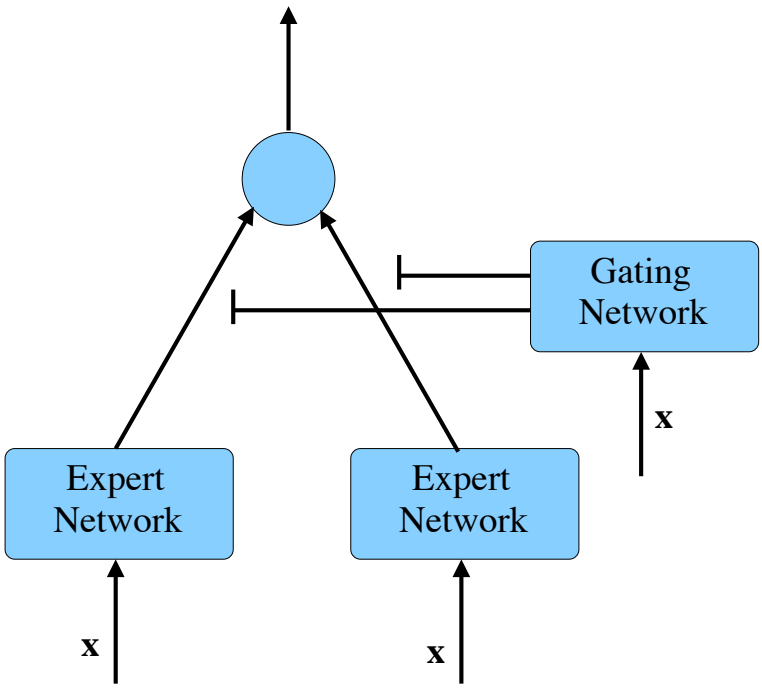
$$\begin{aligned} P(\omega|\boldsymbol{x}, \boldsymbol{\theta}) &= \sum_{m=1}^{M} P(\tilde{\omega}_m, \omega|\boldsymbol{x}, \boldsymbol{\theta}) \\ &= \sum_{m=1}^{M} P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta}) P(\omega|\boldsymbol{x}, \boldsymbol{\theta}_m) \end{aligned}$$

where $\tilde{\omega}_m$ indicates the expert and ω indicates the class.

The gating function, that gives $P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta})$ must yield a valid PMF for all observations $\boldsymbol{x}$

$$\sum_{m=1}^{M} P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta}) = 1; \quad P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta}) \geq 0$$

Mixture of Experts Structure



The diagram shows a simple two expert mixture of experts (MoEs). The gating function effectively determines the contribution that each of the experts should make, given knowledge of the input vector $\boldsymbol{x}$.

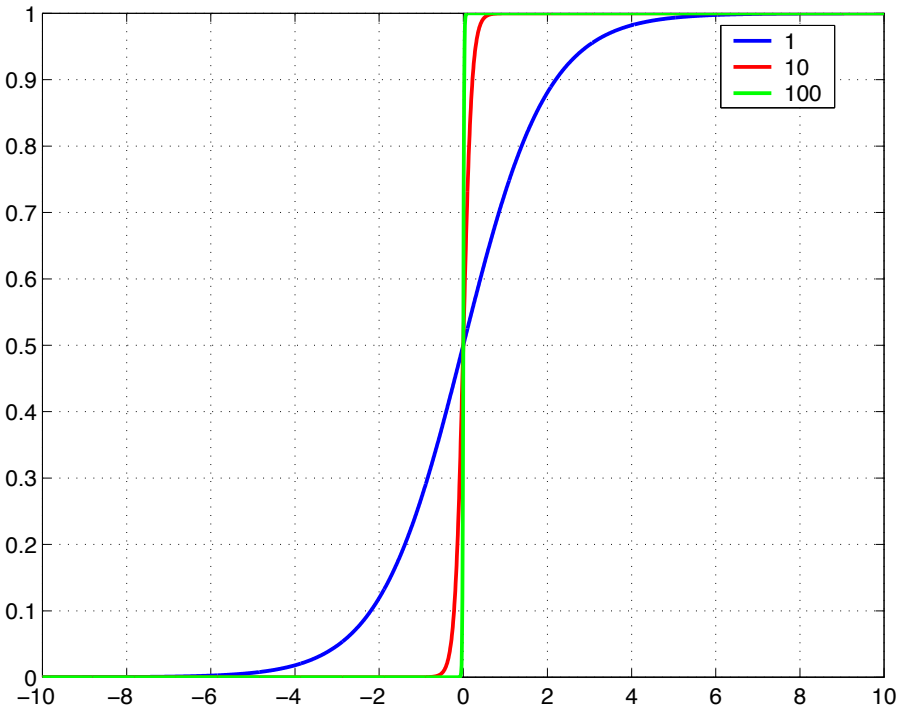
When specifying an MoE it is necessary to define the form and be able to train the:

- gating function
- expert

The exact form will vary from task to task.

Softmax Gating Functions

We have already come across a form of function that is appropriate as a gating function. It is guaranteed to yield a valid PMF for all observations.
For the two class problem - the sigmoid is one option.



For multiple class, a softmax function can be used. This can be written as

$$P(\tilde{\omega}_m|\boldsymbol{x},\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{b}'_m\boldsymbol{x} + c_m)}{\Sigma_{m=1}^{M}\exp(\boldsymbol{b}'_m\boldsymbol{x} + c_m)}$$

It is simple to see that for this case when $\boldsymbol{b}_m = \mathbf{0}$ for all components the gating function is independent of the observations. It then becomes the equivalent of the standard mixture model.

Mixture of Experts for Regression

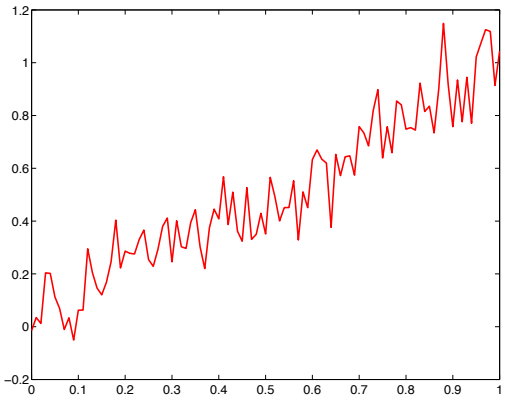
A standard example of the use MoEs is regression. Here given some observation $\boldsymbol{x}$ we need to predict a continuous valued output y. If the regression task can be split into distinct regions it may be better to use multiple experts, rather than one complicated one.

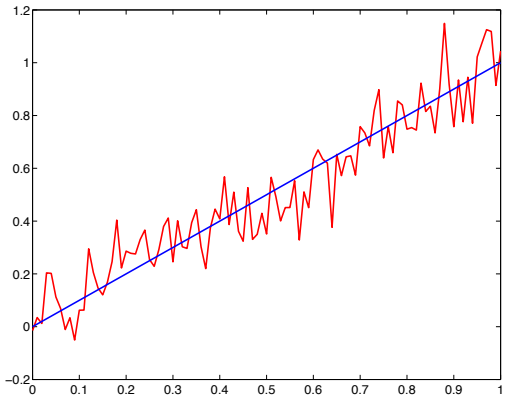
Consider the simple linear regression case an expert predicts the output y given an observation $\boldsymbol{x}$

$$\hat{y} = \boldsymbol{\mu}'\boldsymbol{x}$$

If the error on this predictor is assumed to be Gaussian distributed with variance σ^2 then

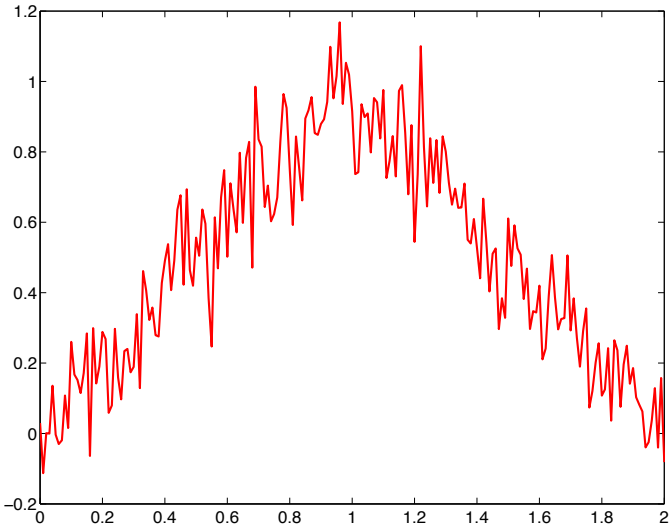
$$p(y|\boldsymbol{x}, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}(y; \boldsymbol{\mu}'\boldsymbol{x}, \sigma^2)$$





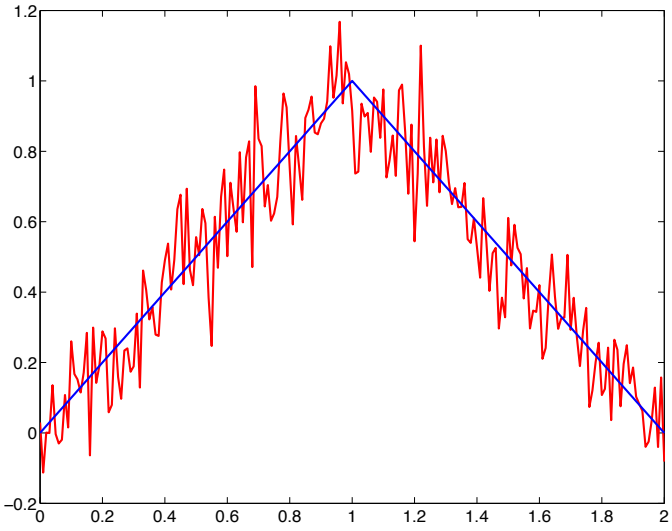
Mixture of Experts for Regression

Now consider the more complicated form



Using a MoE for prediction with a gating function

$$P(\tilde{\omega}_1|x,\boldsymbol{\theta}) = \begin{cases} 1; & x \leq 1 \\ 0; & \text{otherwise} \end{cases}$$



Training MoEs

To train a MoE for regression two forms and sets of parameters need to be determined:

1. Gating function: how to partition/smooth the space given the observation $\boldsymbol{x}$;
2. Expert: for each partition what predictor should be used.

For the simple linear predictor with Gaussian noise (the same variance is shared over all predictors) with a softmax gating function, the overall likelihood can be expressed as

$$\begin{aligned} p(y|\boldsymbol{x}, \boldsymbol{\theta}) &= \sum_{m=1}^{M} P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta}) p(y|\boldsymbol{x}, \boldsymbol{\mu}_m) \\ &= \sum_{m=1}^{M} \left(\frac{\exp(\boldsymbol{b}'_m \boldsymbol{x} + c_m)}{\sum_{m=1}^{M} \exp(\boldsymbol{b}'_m \boldsymbol{x} + c_m)} \right) \mathcal{N}(y; \boldsymbol{\mu}'_m \boldsymbol{x}, \sigma^2 \mathbf{I}) \end{aligned}$$

where

$$\mathcal{N}(y; \boldsymbol{\mu}'_m \boldsymbol{x}, \sigma^2 \mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(\frac{-(y - \boldsymbol{\mu}'_m \boldsymbol{x})^2}{2\sigma^2} \right)$$

This has similar form to the standard mixture model, except that the prior (component weight) is a function (softmax) of the observation.

Maximum Likelihood Training

To train the predictor/regression supervised, N data samples are assumed to be available of the form

$$\mathcal{D} = \{\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}, \ldots, \{\boldsymbol{x}_N, y_N\}\}$$

Maximum likelihood training can be used to estimate the parameters of the MoE model. Again the log-likelihood is usually optimised:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log(p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}))$$

The model parameters for each expert m are

1. expert: comprises the predictor $\boldsymbol{\mu}_m$ and the (shared) variance term σ^2
2. gating function: $\{\boldsymbol{b}_m, c_m\}$

One option is to use gradient descent based approaches - require:

$$\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta}) = \mathbf{0}$$

In the same fashion as standard mixture models this has no closed-from solution. Alternative is to use EM in the same fashion as the training of mixture models.

Expectation Maximisation

The auxiliary function has the same form as the standard mixture case discussed in the previous lecture. Thus

$$\mathcal{Q}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(k+1)}) = \sum_{i=1}^{N} \sum_{m=1}^{M} P(\tilde{\omega}_m|y_i, \boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) \left[\log(P(\tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k+1)})) + \log(p(y_i|\boldsymbol{x}_i, \tilde{\omega}_m, \boldsymbol{\theta}^{(k+1)}))\right]$$

This has split the optimisation into two distinct parts

- Expert: this is the standard optimisation discussed in the previous lecture. For each expert m need to maximise:

  $$\sum_{i=1}^{N} P(\tilde{\omega}_m|y_i, \boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) \log(p(y_i|\boldsymbol{x}_i, \tilde{\omega}_m, \boldsymbol{\theta}^{(k+1)}))$$

  This is exactly the same as the standard optimisation problem for EM.

- Gating function: Need to maximise:

  $$\sum_{i=1}^{N} \sum_{m=1}^{M} P(\tilde{\omega}_m|y_i, \boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) \log(P(\tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k+1)}))$$

  where for the sigmoidal case

  $$P(\tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{b}'_m \boldsymbol{x}_i + c_m)}{\Sigma_{m=1}^{M} \exp(\boldsymbol{b}'_m \boldsymbol{x}_i + c_m)}$$

  Unlike the expert this cannot be split into individual component optimisations. However it looks like a weighted version of logistic regression optimisation from 3F3. This will be further examined later in the course.

Expert Posteriors

When we previously examined EM for Gaussian mixture models the posterior probabilities were defined as $P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta})$ - this is now given by the gating function! The important difference for training MoEs for regression is that the posterior of interest is based on both the input observation and the output regression value.

When training MoEs the posterior we are interested in is given by

$$P(\tilde{\omega}_m|y_i, \boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) = \frac{p(y_i, \tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k)})}{\Sigma_{j=1}^{M} p(y_i, \tilde{\omega}_j|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k)})}$$

Consider just the numerator terms

$$\begin{aligned} p(\tilde{\omega}_m, y_i|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) &= P(\tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) P(y_i|\tilde{\omega}_m, \boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) \\ &= P(\tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k)}) \mathcal{N}(y_i, \boldsymbol{x}_i'\boldsymbol{\mu}_m, \sigma^2) \end{aligned}$$

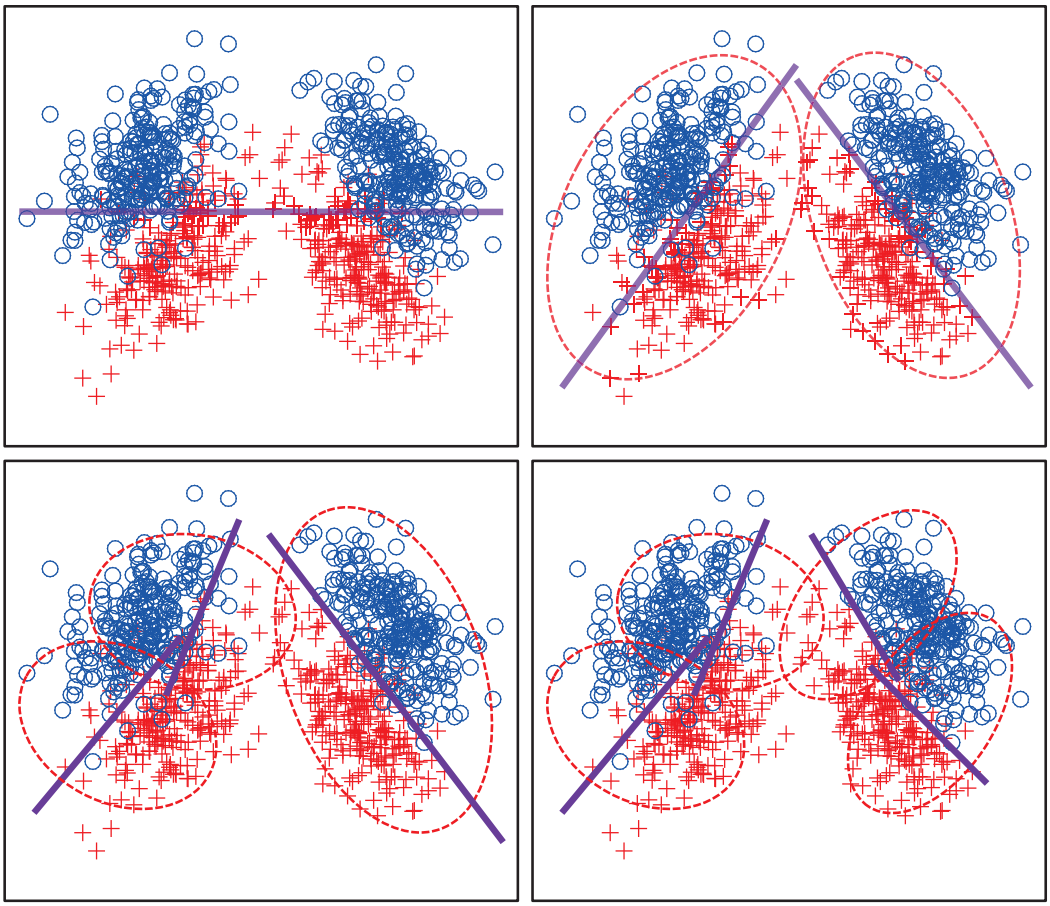
Two different posteriors for the expert

1. training (EM): $P(\tilde{\omega}_m|y_i, \boldsymbol{x}_i, \boldsymbol{\theta}^{(k)})$
2. gating: $P(\tilde{\omega}_m|\boldsymbol{x}_i, \boldsymbol{\theta}^{(k)})$

To emphasise this difference the posterior from the gating function is sometimes referred to as the prior.

Mixture of Experts for Classification

Instead of using a mixture of expert framework for prediction it can also be used to combine multiple classifiers.



Here the gating function varies the contribution of the classifier to decision boundary

$$P(\omega|\boldsymbol{x},\boldsymbol{\theta}) = \sum_{m=1}^{M} P(\tilde{\omega}_m|\boldsymbol{x},\boldsymbol{\theta})P(\omega|\boldsymbol{x},\boldsymbol{\theta}_m)$$

where $\tilde{\omega}_m$ indicates the expert and ω indicates the class.

What About Gaussian Experts?

The original example of a mixture model was the Gaussian mixture model which was a valid probability density function. Rather than using a fixed prior what happens if a gating function is used?

For the case of likelihood experts - applying this form yields

$$p(\boldsymbol{x}|\boldsymbol{\theta}) \propto \sum_{m=1}^{M} P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{x}|\tilde{\omega}_m, \boldsymbol{\theta}_m)$$

The likelihood is only proportional to mixture of experts as generally the product of two probabilities does not yield a valid distribution (without appropriate normalisation).

This is simple to see from the terms in the summation

$$P(\tilde{\omega}_m|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{x}|\tilde{\omega}_m, \boldsymbol{\theta}_m)$$

This cannot be expressed as a joint distribution.

The value of the normalisation term can be found by integrating over all possible values of $\boldsymbol{x}$ as usual. Unfortunately for many situations this is not simple.

Product of Experts

Rather than using a mixture, sum, of the likelihood the product can also be used. This has exactly the same problem as the previous slide. Thus

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\theta}) &= \frac{1}{Z}\prod_{m=1}^{M} p(\boldsymbol{x}|\boldsymbol{\theta}_m)^{\lambda_m} \\ &= \frac{1}{Z}\exp\left(\sum_{m=1}^{M}\lambda_m \log(p(\boldsymbol{x}|\boldsymbol{\theta}_m))\right) \end{aligned}$$

where

$$Z = \int \exp\left(\sum_{m=1}^{M}\lambda_m \log(p(\boldsymbol{x}|\boldsymbol{\theta}_m))\right) d\boldsymbol{x}$$

to ensure that this is a valid PDF.
There are again two forms of parameter associated with this model

1. λ_m: the weight associated to the expert
2. $p(\boldsymbol{x}|\boldsymbol{\theta}_m)$: nature of the expert

It is also possible to use product of experts for classifiers

$$P(\omega|\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{Z}\prod_{m=1}^{M} P(\omega|\boldsymbol{x},\boldsymbol{\theta}_m)^{\lambda_m}$$

Here the normalisation term is required to yield a valid PMF

$$Z = \sum_{\omega}\left(\prod_{m=1}^{M} P(\omega|\boldsymbol{x},\boldsymbol{\theta}_m)^{\lambda_m}\right)$$

Gaussian Distributions

Conditional Gaussian distribution:
Consider the joint distribution between vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right)$$

We are interested in the conditional distribution, which itself is Gaussian

$$\boldsymbol{x}_1 | \boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})$$

where

$$\begin{aligned} \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned}$$

Product of Gaussian distributions:
Consider the two distributions

$$p_1(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \quad p_2(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

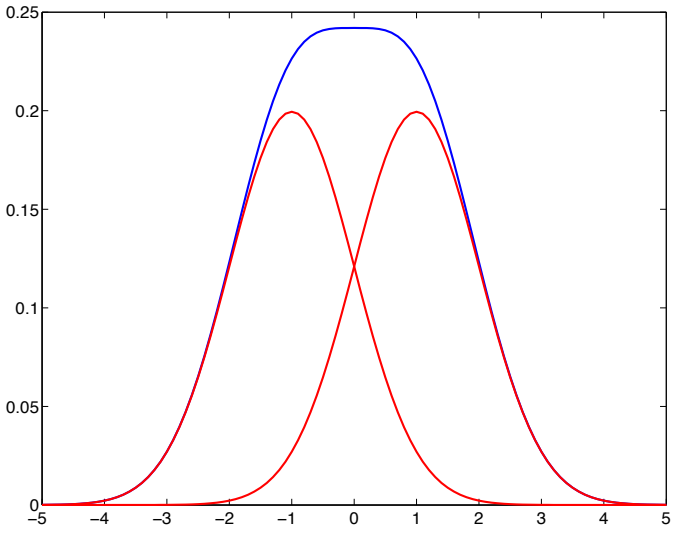
The product is an un-normalised Gaussian

$$p_1(\boldsymbol{x})p_2(\boldsymbol{x}) \propto \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

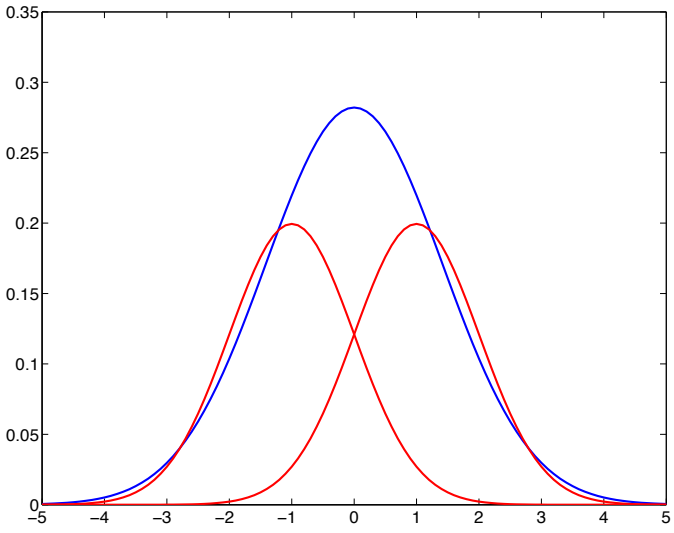
where

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\Sigma}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) \\ \boldsymbol{\Sigma} &= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \end{aligned}$$

Mixtures and Product of Gaussians



$$p(x|\boldsymbol{\theta}) = 0.5 \times \mathcal{N}(x; 1, 1) + 0.5 \times \mathcal{N}(x; -1, 1)$$



$$p(x|\boldsymbol{\theta}) = \frac{1}{Z}\left(\mathcal{N}(x; 1, 1)\mathcal{N}(x; -1, 1)\right)$$

Product of Gaussians

From the previous slides producting Gaussians together yields a Gaussian. For the general case of producting M multivariate Gaussian distributions together (setting $\lambda_m = 1$)

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{Z} \prod_{m=1}^{M} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

Using the equalities from the previous slide it is possible to find the parameters of the producted Gaussian

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\Sigma} \left(\sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\mu}_m \right) \\ \boldsymbol{\Sigma} &= \left(\sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{-1} \right)^{-1} \end{aligned}$$

It is then easy to show that the appropriate normalisation term is (see examples paper)

$$Z = \frac{\Pi_{m=1}^{M} (2\pi)^{d/2} |\boldsymbol{\Sigma}_m|^{1/2}}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}}$$

The Gaussian normalisation terms for the individual experts are cancelled by the normalisation term, they can be ignored. In this case the Gaussians are effectively a set of exponential experts.

Product of Gaussian "Pancakes"

From the previous slide if full covariance models are used then there is no advantage of a product of Gaussians. However is is not necessary for individual experts to be valid PDFs, provided that the overall, producted, distribution is a valid PDF.

This is simply illustrated by considering two Gaussian experts each only modelling one dimension. Thus the experts for observation $\boldsymbol{x} = [x_1, x_2]'$

$$\begin{aligned} &\text{dimension 1} \quad \mathcal{N}(x_1; \mu_1, \sigma_1^2) \\ &\text{dimension 2} \quad \mathcal{N}(x_2; \mu_2, \sigma_2^2) \end{aligned}$$

A product of expert system then becomes

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma_1\sigma_2)} \prod_{i=1}^{2} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$

This is a diagonal covariance matrix multivariate Gaussian.

More generally consider exponential experts of the form

$$\exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_m)'\boldsymbol{\Lambda}_m(\boldsymbol{x} - \boldsymbol{\mu}_m)\right)$$

The resultant covariance matrix of the M experts is

$$\boldsymbol{\Sigma}^{-1} = \left(\sum_{m=1}^{M} \boldsymbol{\Lambda}_m\right)$$

This is a valid PDF if $\boldsymbol{\Sigma}$ is of full rank

Training PoEs

Products of experts can be trained using maximum likelihood. Thus

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \log(p(\boldsymbol{x}_i|\boldsymbol{\theta})) \\ &= \sum_{i=1}^{N} \left[-\log(Z) + \sum_{m=1}^{M} \lambda_m \log(p(\boldsymbol{x}_i|\boldsymbol{\theta}_m)) \right]\end{aligned}$$

The second term in the above expression is the standard log-likelihood, weighted by λ_m. Unfortunately the first term is problematic.

Consider the simple case of Gaussian experts. Here

$$Z = \frac{\Pi_{m=1}^{M}(2\pi)^{d/2}|\boldsymbol{\Sigma}_m|^{1/2}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}$$

The means of the Gaussian experts is exactly the same as the standard Gaussian

$$\boldsymbol{\nabla}_{\mu} = \sum_{i=1}^{N} \lambda_m \boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)$$

If full covariance matrices are used for the Gaussians - the final distribution is Gaussian - no need to estimate the individual experts. If the experts are pancakes then gradient descent can be used.

So far we have assumed that it is possible to get an analytical solution to the normalisation tern, not always true. How to estimate the model parameters if it is not possible to find Z?

Summary

This lecture has described two modifications to the mixture models described in the previous lectures

- Mixture of Experts (MoEs): the component prior is specified by a gating function (a function of the observation $\boldsymbol{x}$). This allows for example different experts to be specified (and combined) in different regions of observation space.
- Product of Experts (PoEs): rather than adding the likelihoods together as in the mixture model, likelihoods are producted and the resultant value normalised to yield a valid PDF.

One issue observed with PoE is that training can be problematic because of the normalisation term. The next lecture will discuss approaches for training models where it is not possible to get analytic expressions for the normalisation term.