Pattern Recognition and Machine Learning

James L. Crowley

ENSIMAG 3 MMIS First Semester 2010/2011

Lesson 6 17 November 2010

Generative Methods for Classification

Outline:

Sources Bibliographiques :
"Neural Networks for Pattern Recognition", C. M. Bishop, Oxford Univ. Press, 1995.
"Pattern Recognition and Scene Analysis", R. E. Duda and P. E. Hart, Wiley, 1973.

Notation

x	a variable
X	a random variable (unpredictable value)
$\vec{X}$	A vector of D random variables.
D	The number of dimensions for the vector $\vec{x}$ or $\vec{X}$
$\{\vec{X}_m\}$	A set of M examples
M	Total number of examples.
$p(X)$	Probability density function for X
$p(\vec{X})$	Probability density function for $\vec{X}$
N	The number components in a Gaussian Mixture model

Gaussian Mixture model: $p(\vec{X}) = \sum_{n=1}^{M} \alpha_n \mathcal{N}(\vec{X}; \vec{\mu}_n, \Sigma_n)$

V	a volume of the space $\vec{X}$
K	The number of samples from $\{\vec{X}_m\}$ in the volume V.

The probability $p(\vec{X})$ for $\vec{X} \in V$ is $p(\vec{X}) = \frac{K}{MV}$

Gaussian Mixture Models

Gaussian Mixtures as sum of Independent Sources

The "Central Limit Theorem" tells us that whenever an observation is the result of a sequence of N independent random events, the probability density of the features will tend toward a Normal or Gaussian density.

$$p(\vec{X}) = \mathcal{N}(\vec{X};\vec{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}-\vec{\mu})^T \Sigma^{-1}(\vec{X}-\vec{\mu})}$$

Unfortunately, this hypothesis does not always apply. A common case occurs when the event may come from one of a set of different "sources", each with its own density function.

In this case, the probability density is better represented as a weighted sum of normal densities.

$$p(\vec{X}) = \sum_{n=1}^{M} \alpha_n \mathcal{N}(\vec{X};\vec{\mu}_n,\Sigma_n)$$

Each normal density results from a different source. We can see teh coefficients $\{\alpha_n\}$ as the relative frequencies (probabilities) for a set of independent "sources" for the event. The α_n coefficients represent the relative probability that event came from source "n".

$$\alpha_n = p(E \leftarrow Source(n))$$

Thus we must assure that $\sum_{n=1}^{N} \alpha_n = 1$

Such a sum is referred to as a Gaussian Mixture Model. It can also be used to represent density functions where the Central Limit theorem does not apply or that have more complex forms. It can also be used to discover a set of subclasses within a global class.

It is sometimes convenient to group the parameters for each source into a single vector:

$$\vec{v}_n = (\alpha_n, \vec{\mu}_n, \Sigma_n)$$

For a feature vector of D dimensions, $\vec{v}_n$ has P = 1 + D + D(D+1)/2 coefficients.

The complete set of parameters is a vector with N·P coefficients.

$$\vec{v} = (\vec{v}_1, \vec{v}_2, ..., \vec{v}_n)$$

To estimate the parameters $\{\alpha_n\}$ we need the parameters $\{\vec{\mu}_n, \Sigma_n\}$

To estimate $\{\vec{\mu}_n, \Sigma_n\}$ we need $\{\alpha_n\}$.

This leads to an iterative two-step process in which we alternately estimate $\{\vec{\mu}_n, \Sigma_n\}$ and $\{\alpha_n\}$. To do this, we construct a table, h(m, n)

h(m, n) = Pr{the event E_m is from source n}

The iterative algorithm for this estimation is called EM: Expectation Maximisation.

Expectation Maximisation Algorithm

EM iteratively estimates a model for the density function as a composition of N unknown sources. Each source is assumed to have a different Normal density.
This has many uses, including estimating the density functions for a Hidden Markov Model (HMM) as well as for estimating the parameters for a Gaussian Mixture model.

EM operates on an unlabeled training set of M observations $\{\vec{X}_m\}$.

The EM algorithm will iterate between estimating the probability that each observation belongs to each of N sources, and estimate the mean and covariance for each source.

Each source can be interpreted as a separate class.
Because EM operates on an unlabeled training set it can be used to discover classes by Unsupervised Learning.

We suppose that each observation, m, is from one of N sources: h_m=n
The sources are unknown (hidden).

h_m = n is equivalent to writing then $h_m(n)$=1 else $h_n(m)$=0.

However, we will not estimate Boolean values, but probabilities.

$h_m(n) = h(m,n) = \text{Prob}\{$ Observation m is from Source n$\}$

Initialisation:

Choose N (the number of sources).
set *i=1*.
Form an initial estimate for $\vec{v}^{(1)} = (\alpha_n^1, \vec{\mu}_n^1, \Sigma_n^1)$ *for n = 1 to N.*
This can be initialised with $\alpha_n^1 = \frac{1}{N}$, $\vec{\mu}_n^1 = n\vec{\mu}_0^1$, $\Sigma_n^1 = I$ or with any reasonable first estimation. The closer the initial estimate, the faster the algorithm converges.

Expectation step (E)

Calculate the table $h(m,n)^{(i)}$ using the training data and estimated parameters.

$$h(m,n)^{(i)} = p((h_m = n) \mid \{X_m\}, \vec{v}^{(i)})$$

$$h(m,n)^{(i)} = \frac{\alpha_n \mathcal{N}(\vec{X}_m, \vec{\mu}_n, \Sigma_n)}{\sum_{j=1}^{N} \alpha_j \mathcal{N}(\vec{X}_m, \vec{\mu}_j, \Sigma_j)}$$

Maximization Step (M)

Estimate the parameters $\vec{v}^{(i+1)}$ using $h(m,n)^{(i)}$

M: (Maximisation)

$$S_n^{(i+1)} := \sum_{m=1}^{M} h(m, n)^{(i)}$$

$$\alpha_n^{(i+1)} := \frac{1}{M} S_n^{(i+1)}$$

$$\mu_n^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m, n)^{(i)} X_m$$

$$\Sigma_n^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i+1)} (\vec{X} - \vec{\mu}_n^{(i+1)})(\vec{X} - \vec{\mu}_n^{(i+1)})^T$$

Convergence Criteria

The Log-likelihood of the parameter vector is

$$Q^{(i)} = \ln\{p(\{\vec{X}_m\} \mid \vec{v}^{(i)})\} = \sum_{m=1}^{M} \ln\left\{\sum_{j=1}^{N} \alpha_j^{(i)} \mathcal{N}(\vec{X}_m \mid \mu_j^{(i)}, \Sigma_j^{(i)})\right\}$$

It can be shown that, for EM, the log likelihood will converge to a stable maximum. The change in Q will monotonically decrease. When

$\Delta Q = Q^{(i)} - Q^{(i-1)}$ is less than a threshold, halt.

Non Parametric Methods

Three popular Non-parametric methods are
1) Multi-dimensional Histograms
2) Kernel Density Estimators
3) K-Nearest Neighbors

Histograms are tables of size $Q=N^D$ where N is the number of quantizations for each feature and D is the number of features.

Given a training set of M observations $\{\vec{X}_m\}$.

for m=1 to M $h(\vec{X}_m) = h(\vec{X}_m) + 1$

Then for the data set $p(\vec{X}) = \frac{1}{M} h(\vec{X})$

As we have seen, if we have K classes of training data, with M_k training samples $\{\vec{X}_m^k\}$ we can build K histograms $h_k(X)$.

Then $p(\vec{X} \mid C_k) = \frac{1}{M_k} h_k(\vec{X})$ $p(\vec{X}) = \frac{1}{M} h(\vec{X})$ $p(C_k) = \frac{M_k}{M}$

so that $p(C_k \mid \vec{X}) = \frac{h_k(\vec{X})}{h(\vec{X})}$

We have used histograms extensively in previous lectures, and will not spend much time on them now. We will however use them to illustrate a point about non-parametric methods.

Histograms have the advantages:
1) They have a fixed size, Q, independent of the quantity of data. It is not necessary to store the data.
2) They can be composed and used incrementally.

The disadvantage is that
1) Each feature must be quantized over a limited range of N values.
2) We need M >> Q data samples.
3) There are discontinutities at the boundaries of each cell.

Because the $M = \sum_{\vec{X}} h(\vec{X})$ we are sure that $\sum_{\vec{X}} p(\vec{X}) = 1$

If the quantity of training data is too small, ie M < Q we can combine adjacent cells so as to amass enough data for a reasonable estimate.

Let us define the volume of each cell as 1.
Then the volume of the entire space is $Q=N^D$.

Suppose we combine V adjacent cells such that we obtain K samples. The volume of the combined cells would be V and K be the sum of the histogram in these V cells.

$$K = \sum_{\vec{X} \in V} h(\vec{X})$$

The probability $p(\vec{X})$ for $\vec{X} \in V$ is $p(\vec{X}) = \frac{K}{MV}$

Suppose our samples $\{\vec{X}_m\}$ are drawn from a density $p(\vec{X})$.
If take a volume, V, from this density then

$$p(\vec{X}_m \in V) = \frac{K}{MV}$$

We can use this equation to develop two alternative non-parametric methods.

Fix V and determine K => Kernel density estimator.
Fix K and determine V => K nearest neightbors.

Kernel Density Estimators

For a Kernel density estimator, we will represent each data point with a kernel function $k(\vec{X})$.

Popular Kernel functions are
- a hypercube centered of side w
- a sphere of raduis w
- a Gaussian of standard deviation w.

We can define the kernel function for the hypercube as

$$k(\vec{u}) = \begin{cases} 1 & if \ |u_d| \le 1/2 \ \text{ for all } d = 1,...,D \\ 0 & otherwise \end{cases}$$

This is called a Parzen window.

For a position $\vec{X}$, the total number of points lying with a cube with side w will be:

$$K = \sum_{m=1}^{M} k\left(\frac{\vec{X} - \vec{X}_m}{w}\right)$$

The volume of the cube $V = \frac{1}{w^D}$.

Thus the probability $p(\vec{X}) = \frac{K}{MV} = \frac{1}{Mw^D} \sum_{m=1}^{M} k\left(\frac{\vec{X} - \vec{X}_m}{w}\right)$

The Hypercube has a discontinuity at the boundaries. We can soften this using a triangular function evaluated on a sphere.

$$k(\vec{u}) = \begin{cases} 1 - 2\|\vec{u}\| & if \ \|\vec{u}\| \le 1/2 \ \text{ for all d} = 1,...,\text{D} \\ 0 & otherwise \end{cases}$$

Even better is to use a Gaussian kernel with standard deviation $\sigma = w$.

$$k(\vec{u}) = e^{-\frac{1}{2}\frac{\|\vec{u}\|^2}{w^2}}$$

We can note that the volume is $V = (2\pi)^{D/2} w^D$

In this case $p(\vec{X}) = \frac{K}{MV} = \frac{1}{M(2\pi)^{D/2} w^D} \sum_{m=1}^{M} k\left(\vec{X} - \vec{X}_m\right)$

This corresponds to placing a Gaussian over each point and summing the Gaussians.

In fact, we can choose any Kernel $k(\vec{u})$ such that

$$k(\vec{u}) \ge 0 \ \text{ and } \ \int k(\vec{u}) d\vec{u} = 1$$

K Nearest Neighbors

For K nearest neighbors, we hold K constant and vary V.

As each data samples, $\vec{X}_m$, arrives, we construct a tree structure (such as a KD Tree) that allows us to easily find the K nearest neighbors for any point $\vec{X}$

Then to compute $p(\vec{X})$ we determine the K nearest points to $\vec{X}$. We use the distance to the K^{th} point as the radius of a sphere and compute the volume of the sphere in D dimensions.

$$V = C_D \left\| \vec{X} - \vec{X}_K \right\|^D$$

where

$$C_D = + \frac{\pi^{\frac{D}{2}}}{\Gamma\left(\frac{D}{2}+1\right)}$$

Then as before:

$$p(\vec{X}) = \frac{K}{MV}$$