Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2010/2011

Lesson 18 22 april 2011

EM and Gaussian Mixture Models

Sources Bibliographiques :

"Pattern Recognition and Machine Learning", C. M. Bishop, Springer Verlag, 2006.

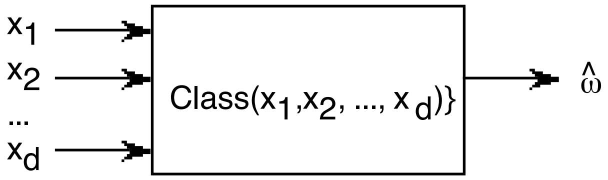
"Pattern Recognition and Scene Analysis", R. E. Duda and P. E. Hart, Wiley, 1973.

Notation

x	a variable
X	a random variable (unpredictable value)
N	The number of possible values for X (Can be infinite).
$\vec{x}$	A vector of D variables.
$\vec{X}$	A vector of D random variables.
D	The number of dimensions for the vector $\vec{x}$ or $\vec{X}$
E	An observation. An event.
C_k	The class (tribe) k. Not to be confused with:
$\mathbf{C_k}$	The covariance for class k.
k	Class index
K	Total number of classes
ω_k	The statement (assertion) that E $\in T_k$
$p(\omega_k) = p(E \in T_k)$	Probability that the observation E is a member of the class k. Note that $p(\omega_k)$ is lower case.
M_k	Number of examples for the class k. (think M = Mass)
M	Total number of examples. $M = \sum_{k=1}^{K} M_k$
$\{X_m^k\}$	A set of M_k examples for the class k. $\{X_m\} = \bigcup_{k=1,K} \{X_m^k\}$
$P(X)$	Probability density function for X
$P(\vec{X})$	Probability density function for $\vec{X}$
$P(\vec{X} \mid \omega_k)$	Probability density for $\vec{X}$ the class k. ω_k = E $\in T_k$.

Bayesian Classification

Our problem is to build a box that maps a set of features $\vec{X}$ from an Observation, E into a class T_k from a set of K possible Classes.



Let ω_k be the proposition that the event belongs to class k: $\omega_k = E \in T_k$

ω_k Proposition that event E $\in$ the class k

In order to minimize the number of mistakes, we will maximize the probability that $\omega_k \equiv E \in T_k$

1) $$\hat{\omega}_k = \arg-\max_k\left\{\Pr(\omega_k \mid \vec{X})\right\}$$

We will call on two tools for this:

2) Baye's Rule :

$$p(\omega_k \mid \vec{X}) = \frac{P(\vec{X} \mid \omega_k)p(\omega_k)}{P(\vec{X})}$$

3) Normal Density Functions

$$P(\vec{X} \mid \omega_k) = \frac{1}{(2\pi)^{\frac{D}{2}}\det(C_k)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}-\vec{\mu}_k)^T C_k^{-1}(\vec{X}-\vec{\mu}_k)}$$

Last week we looked at normal density function. Today we look at sums of Normal Density functions:

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

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".

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

For this to be a probability, 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.

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

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) = P{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) = 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)

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

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

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

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

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.

Likelihood

The Likelihood of a parameter vector, $\vec{v}$, given a training set, $\{X_m\}$ is defined as

$$L(\vec{v} \mid \{X_m\}) = P(\{X_m\} \mid \vec{v}) = \prod_{m=1}^{M} P(X_m \mid \vec{v})$$

For normal density functions, $P(\vec{X}) = \mathcal{N}(\vec{X};\vec{\mu},C) = \frac{1}{(2\pi)^{\frac{D}{2}} \det(C)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}-\vec{\mu})^T C^{-1} (\vec{X}-\vec{\mu})}$

it is more convenient to work with the Log-Likelihood

$$\mathcal{L}(v) = Log\{L(\hat{v} \mid \{X_m\}) = Log\{P(\{X_m\} \mid \hat{v})\} = \sum_{m=1}^{M} Log\{P(X_m \mid \hat{v})\}$$

MLE for a Univariate Gaussian Density functions

For D=1, $\mathcal{N}(X; \mu,\sigma)$ the paremeter vector is $\vec{v} = (\mu, \sigma)$

To estimate μ,σ using MLE, define the log likelihood.

$$\mathcal{L}(\vec{v}) = Log\{P(X_m \mid \vec{v})\} = -\frac{1}{2} Log\{2\pi\sigma^2\} - \frac{1}{2\sigma^2}(X_m - \mu)^2$$

The maximum Log Likelihood occurs when the derivative is zero.

$$\frac{\partial l(v)}{\partial \mu} = \sum_{m=1}^{M} \frac{1}{\sigma^2}(X_m - \mu) = 0$$

$$\frac{\partial l(\vec{v})}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \frac{(X_m - \mu)^2}{2\sigma^4} = 0$$

We formulate this as the gradient

$$\nabla_{\mu,\sigma}\mathcal{L}(\vec{v}) = \begin{pmatrix} \frac{\partial l(v)}{\partial \mu} \\ \frac{\partial l(\vec{v})}{\partial \sigma^2} \end{pmatrix} = \begin{pmatrix} \sum_{m=1}^{M} \frac{1}{\sigma^2}(X_m - \mu) \\ -\frac{1}{2\sigma^2} + \frac{(X_m - \mu)^2}{2\sigma^4} \end{pmatrix} = 0$$

$$\nabla_{\mu,\sigma}\mathcal{L}(\vec{v}) = \begin{pmatrix} \frac{1}{\sigma^2}(X_m - \mu) \\ -\frac{1}{2\sigma^2} + \frac{(X_m - \mu)^2}{2\sigma^4} \end{pmatrix} = 0$$

with a little algebra:

$$\hat{\mu} = \frac{1}{M}\sum_{m=1}^{M} X_m$$

$$\sigma^2 = \frac{1}{M}\sum_{m=1}^{M} (X_m - \mu)^2$$

See lecture 17 for the derivation.

Maximum Likelihood for a Multivariate Density Function

The principle is the same for D >1, however the equations are more complicated.

$\vec{v} = (\vec{v}_1, \vec{v}_2, ..., \vec{v}_n)$ with each $\vec{v}_n = (\alpha_n, \vec{\mu}_n, C_n)$

$$\mathcal{L}(\hat{v}) = Log\{P(\vec{X}_m \mid \vec{v})\} = -\frac{1}{2} Log\{(2\pi)^D \det(C)\} - \frac{1}{2}(\vec{X}_m - \mu)^T C^{-1}(\vec{X}_m - \mu)$$

$$\hat{v} = \max_v\{\prod_{m=1}^{M} P(\vec{X}_m \mid \vec{v})\} = \max_v\{\sum_{m=1}^{M} Log(P(\vec{X}_m \mid \vec{v}))\}$$

The most likely $\hat{v}$ may be found when the gradient of $\hat{v}$ is null.

$$\nabla_V \mathcal{L}(\vec{v}) = \nabla_V \sum_{m=1}^{M} Log(P(\vec{X}_m \mid \vec{v})) = 0$$

∇_V is the gradient operator: $\nabla_v = \begin{pmatrix} \frac{\partial}{\partial v_1} \\ \frac{\partial}{\partial v_2} \\ ... \\ \frac{\partial}{\partial v_{NP}} \end{pmatrix}$

$$\nabla_v \mathcal{L}(\vec{v}) = \begin{pmatrix} \frac{\partial}{\partial v_1} \\ \frac{\partial}{\partial v_2} \\ ... \\ \frac{\partial}{\partial v_{NP}} \end{pmatrix} \mathcal{L}(\vec{v}) = \begin{pmatrix} \frac{\partial \mathcal{L}(\vec{v})}{\partial v_1} \\ \frac{\partial \mathcal{L}(\vec{v})}{\partial v_2} \\ ... \\ \frac{\partial \mathcal{L}(\vec{v})}{\partial v_{NP}} \end{pmatrix}$$

Setting $\nabla_v l(\vec{v})$=0 gives the classic formulae :

$$\hat{\mu} = \frac{1}{M}\sum_{m=1}^{M} \vec{X}_m \qquad \Sigma = \frac{1}{M}\sum_{m=1}^{M} (\vec{X}_m - \hat{\mu})(\vec{X}_m - \hat{\mu})^T$$