Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2009/2010

Lesson 18 30 april 2010

Gaussian Mixture Models and Expectation-Maximization

Contents

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 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.
T_k	The class (tribe) 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. $\omega_k = E \in T_k$.
N	The number components in a Gaussian Mixture model

Gaussian Mixture model:

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

Maximum Likelihood Estimation.

Our goal is to represent a density function as a weighted sum of normal densities.

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

For this, the problem is to represent the vector of parameters:

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

Where

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

For N components, a feature vector of D dimensions, $\vec{v}_n$ has

$$N \cdot P = N \cdot (1 + D + D(D+1)/2) \text{ coefficients.}$$

Our approach will be to estimate the coefficient vector with the highest probability. For this we need to calculate a Maximum Likelihood Estimate (MLE)

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.

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

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

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

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: $\nabla_v = \begin{pmatrix} \frac{\partial}{\partial v_1} \\ \frac{\partial}{\partial v_2} \\ \dots \\ \frac{\partial}{\partial v_{NP}} \end{pmatrix}$

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

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

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

The EM 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.

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

The EM algorithmwill iterates between estimating the probability that each observation belongs to each of N sources, and estimate the mean and covariance for each source. This has many uses, including estimating the density functions for a Hiddent Markov Model (HMM) as well as for estimating the parameters for a Gaussian Mixture model.

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.

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

Expectation step (E):
Calculate the table $h(m,n)^{(i)}$ using the training data.

$$h(m, n)^{(i)} = p(h_m=n \mid X_1, X_2, ..., X_M, \nu^{(i)})$$

$$h(m, n)^{(i)} = \frac{\alpha_n^{(i)} \mathcal{N}(X_m; \mu_n^{(i)},\sigma_n^{(i)})}{\sum_{j=1}^{N} \alpha_j^{(i)} \mathcal{N}(X_m; \mu_j^{(i)},\sigma_j^{(i)})}$$

Maximization Step: (M)
Calculate $\nu^{(i+1)}$ using $p(h_m \mid X_1, X_2, ..., X_M, \nu^{(i)})$

How can we know when to stop?
We need to have an estimate of the "goodness" of each estimate. This is precisely the likelihood of $\vec{v}_n$

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

$$\Delta Q^{(i)} = Q^{(i)} - Q^{(i-1)}$$

It can be shown that $\Delta Q^{(i)}$ only decreases : $\Delta Q^{(i)} \leq \Delta Q^{(i-1)}$

Thus the estimation is stopped when $\Delta Q^{(i)} \leq$ threshold.

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

E (Expectation):

$$h(m, n)^{(i)} := \frac{\alpha_n^{(i)} \mathcal{N}(X_m; \mu_n^{(i)}, \sigma_n^{(i)})}{\sum_{j=1}^{N} \alpha_j^{(i)} \mathcal{N}(X_m; \mu_j^{(i)}, \sigma_j^{(i)})}$$

M: (Maximisation)

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

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

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

$$\sigma^2{}_n^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i)} (X_m - \mu_n^{(i+1)})^2$$

For D> 1 the covariance C is composed of a matrix of coefficients σ_{jk}^2:

$$\sigma^2_{jkn}{}^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i)} (X_{jm} - \mu_{jn}^{(i+1)})(X_{km} - \mu_{kn}^{(i+1)})$$