Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2014/2015

Lesson 15 1 april 2015

Clustering and non-supervised learning: K-Means and Expectation maximization

Sources:

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

Jeff Bilmes, A Gentle Tutorial of the EM Algorithm, Tech Report, Univ of Washington, 1998. (available for download from course website).

Notation

x	a variable
X	a random variable (unpredictable value)
V	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.
k	index for cluster, data source or GMM Mode
K	Total number of clusters, or sources, of events
N	Total number of sample events. $N = \sum_{k=1}^{K} N_k$
$\{\vec{X}_n\}$	A set of N Sample Observations (a training set)
$\{\vec{y}_n\}$	A set of indicator vectors for the training samples in $\{\vec{X}_n\}$ $\vec{y}_n$ indicates the probability that sample $\vec{X}_n$ came from source S_k
$h(n, k) = y_n(k)$	Indicator variables in matrix form.

Expected Value: $$E\{X\} = \frac{1}{N}\sum_{n=1}^{N} X_n$$

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

Multivariate Normal Density Function

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

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

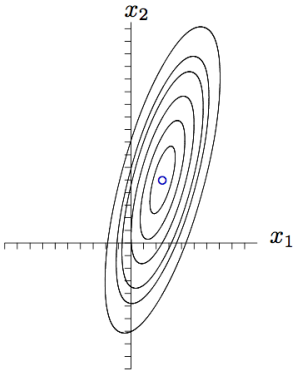
Where the parameters $\vec{\mu}, \Sigma$ and the mean and co-variance of the density. These are the first and second moments of the density.

Note that we use upper case for probabilities and lower case for functions.
Thus $P(\omega)$ is a value, $p(X)$ is a function.

The mean is $\vec{\mu} = E\{\vec{X}\} = \begin{pmatrix} E\{X_1\} \\ E\{X_2\} \\ \dots \\ E\{X_D\} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_D \end{pmatrix}$

and the Covariance is $\Sigma = E\{(\vec{X} - E\{\vec{X}\})(\vec{X} - E\{\vec{X}\})^T\} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1D}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2D}^2 \\ \dots & \dots & \dots & \dots \\ \sigma_{D1}^2 & \sigma_{D2}^2 & \dots & \sigma_{DD}^2 \end{pmatrix}$

The Normal density can be seen as a set of co-encentric ellipses. Each ellipse represents a contour of equal value (or equal probability) for a density function.

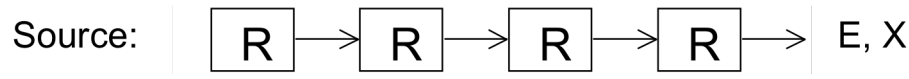


Ellipses for 99%, 95%, 90%, 75%, 50%, and 20%

Gaussian Mixture Models

Gaussian Mixtures as a Sum of Independent Sources

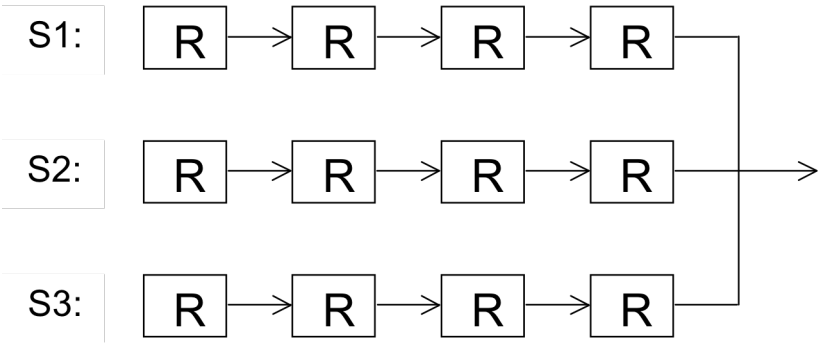
We can consider a sequence of random trials as a "source" of event



The central limit theorem tells us that in this case, the resulting probability can be described by Normal density function:

$$p(\vec{X}) = \mathcal{N}(\vec{X}; \vec{\mu}, \Sigma)$$

Sometimes a population will result from a set of K different sources, S_k.



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

$$p(\vec{X}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\vec{X}; \vec{\mu}_k, \Sigma_k)$$

Such a sum is referred to as a Gaussian Mixture Model. It can be used to represent density functions where the Central Limit theorem does not apply.

It can also be used to discover a set of subclasses within a global class.

Each normal density is considered to be produced from a different source. We can see the coefficients $\{\alpha_n\}$ as the relative frequencies (probabilities) for a set of independent "sources" , S_k, for events. The α_k coefficients represent the relative probability that an event came from a source S_k.

For this to be a probability, we must assure that $\sum_{k=1}^{K} \alpha_k = 1$

Our problem is to discover the source for each sample, and to estimate the mean and covariance $(\vec{\mu}_k, \Sigma_k)$ for each source.

We will look at two possible algorithms for this: K-Means Clustering, and Expectation Maximization.

In both cases, the algorithm will iteratively construct a table, *h(n,k)* that assigns each sample to one of K clusters or sources.

For K-Means, this will be a hard assignment,
with *h(n, k) = 1* if observation $\vec{X}_n$ is assigned to cluster S_k. and 0 otherwise.

This can be seen as equivalent to the indicator variable $y_n(k)$ seen in the last course.

$$h(n,k) = \vec{y}_n(k) = \begin{cases} 1 & \text{if sample } \vec{X}_n \in S_k \\ 0 & \text{Otherwise} \end{cases}$$

$y_n(k)$ = h(n, k) = 1 if E is assigned to cluster k, 0 otherwise.

In the case of EM, this will be a soft assignment, in which *h(n,k)* represents the probability that sample $\vec{X}_n$ comes from source (or cluster), S_k.

$$h(n,k) = P(E_n \in S_k)$$

In either case we must initialize the estimated clusters:

This can be initialized with, $\vec{\mu}_k^1 = k\vec{\mu}_0^1$, $\Sigma_k^1 = I$ or any other convenient value.

K-means is sensitive to the starting point and can converge to a local minimum that is not the best estimate. EM is not sensititve and will converge to the global best estimate

Because K-Means and EM operate on an unlabeled training set, they can be used to discover classes in an unlabeled set of data. This is called Unsupervised Learning.
They can also be used to estimate a multimodal density for a single class.

K-Means Clustering

Assume a set of N sample observations $\{\vec{X}_n\}$, with each observation drawn from one of K clusters S_k. Our problem is to discover an assignment table *h(n, k)* that assigns each observation, $\vec{X}_n$ in the sample set to the "best" cluster, S_k.

The assignment table is equivalent to the indicator variable $\vec{y}_n(k)$ seen in the last course.

$$h(n,k) = \vec{y}_n(k) = \begin{cases} 1 & \text{if sample } \vec{X}_n \in S_k \\ 0 & \text{Otherwise} \end{cases}$$

Given an estimate of the mean, $\vec{\mu}_k$, and covariance Σ_k for each cluster, S_k. we can use the Mahalanobis Distance to determine the best cluster.

For each cluster we can then refine the estimate of the mean, $\vec{\mu}_k$, and covariance Σ_k.

This suggests an iterative process composed of two steps:

1) Expectation: For each sample, $\vec{X}_n$, determine the most likely cluster $S_{k.}$ using the distance to the current estimate of the mean, $\vec{\mu}_k$, and covariance Σ_k.

2) Maximization: For each cluster re-calcuate the mean, $\vec{\mu}_k$, and covariance Σ_k using sample assignments in *h(n,k)*.

We can initialize the process to any value. For example, $\vec{\mu}_k^{(0)} = k\vec{\mu}_0$, $\Sigma_k^{(0)} = I$

However, it IS possible for K-means to be stuck in a local minimum, and the closer we start to the best values, the faster the process converges.

We will seek to minimize a quality metric:
For K-Means this is the sum of the mahalanobis distances.

$$Q^{(i)} = \sum_{n=1}^{N} \sum_{k=1}^{K} h^{(i)}(n,k)(\vec{X}_n - \vec{\mu}_k^{(i)})^T \Sigma_k^{(i)-1} (\vec{X}_n - \vec{\mu}_k^{(i)})$$

Initially $h^{(o)}(n,k) = 0$, $i=0$.

We can stop the process after a fixed number of iterations, or when the assignment table does not change or when $Q^{(i)}$ does not change.

Expectation:

$$i \leftarrow i+1$$

$$\forall n = 1,N: \; h^{(i)}(n,k) \leftarrow \begin{cases} 1 & \text{if k } = \arg\min_{k} \left\{ (\vec{X}_n - \vec{\mu}_k)^T \Sigma_k^{-1} (\vec{X}_n - \vec{\mu}_k) \right\} \\ 0 & \textit{Otherwise} \end{cases}$$

Maximization

Mean:
$$\mu_k^{(i)} = \frac{\sum_{n=1}^{N} h^{(i)}(n,k) \cdot \vec{X}_n}{\sum_{n=1}^{N} h^{(i)}(n,k)}$$

Covariance:
$$\Sigma_k^{(i)} = \frac{\sum_{n=1}^{N} h^{(i)}(n,k) \cdot (\vec{X}_n - \vec{\mu}_k)^2}{\sum_{n=1}^{N} h^{(i)}(n,k)}$$

The process stops after a fixed number of cycles, or when the sample assignment does not change or the quality metric does not change.

Each source can be interpreted as a separate class or as a mode in a Gaussian Mixture model, depending on the application.

The Expectation Maximization Algorithm (EM)

As before, assume a set of N sample observations $\{\vec{X}_n\}$, with each observation drawn from one of K sources S_k. Our problem is to discover an assignment table *h(n, k)* that assigns each observation, $\vec{X}_n$ in the sample set to the "best" cluster, S_k. For EM this will be a probability.

EM iteratively estimates the probability for the assignment of each observation to each source.

Expectation Maximization 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.

For a Gaussian Mixture model, a probability density is represented as a weighted sum of normal densities.

$$p(\vec{X}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\vec{X}; \vec{\mu}_k, \Sigma_k)$$

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

$$\vec{v}_k = (\alpha_k, \vec{\mu}_k, \Sigma_k)$$

The complete set of parameters is a vector with K·P coefficients.
For a feature vector of D dimensions, $\vec{v}_k$ has P = 1 + D + D(D+1)/2 coefficients.

To estimate $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$ we need the assignment of samples to source, *h(n,k)*.
To estimate *h(n,k)* we need the parameters $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$

This leads to an iterative two-step process in which we alternately estimate *h(n,k)*. and then $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$.

The EM algorithms constructs a table, h(m, n)
Unlike K-Means, *h(n,k)* will contain probabilities.

$$h(n,k) = P(E_n \in S_k)$$

Initialisation:

Choose K (the number of sources). Use domain knowledge if possible.

set *i=0*.

Form an initial estimate for $\vec{v}^{(0)} = (\alpha_n^{(0)}, \vec{\mu}_n^{(0)}, \Sigma_n^{(0)})$ *for k = 1 to K.*

This can be initializZed with $\alpha_k^{(0)} = \frac{1}{K}$, $\vec{\mu}_k^{(0)} = k\vec{\mu}_0$, $\Sigma_k^{(0)} = I$

or with any reasonable first estimation. The closer the initial estimate, the faster the algorithm converges. Domain knowledge is useful here.

Expectation step (E)

let $i \leftarrow i+1$

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

$$h(n,k)^{(i)} = p((h_n = k) \mid \{X_n\}, \vec{v}^{(i-1)})$$

which gives :

$$h^{(i)}(n,k) \leftarrow \frac{\alpha_k^{(i-1)} \mathcal{N}(\vec{X}_n, \vec{\mu}_k^{(i-1)}, \Sigma_k^{(i-1)})}{\sum_{j=1}^{K} \alpha_j^{(i-1)} \mathcal{N}(\vec{X}_n, \vec{\mu}_j^{(i-1)}, \Sigma_j^{(i-1)})}$$

Maximization Step (M)

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

M: (Maximisation)

$$\alpha_k^{(i)} \leftarrow \frac{1}{N} \sum_{n=1}^{N} h^{(i)}(n,k)$$

$$\vec{\mu}_k^{(i)} \leftarrow \frac{1}{\sum_{n=1}^{N} h^{(i)}(n,k)} \sum_{n=1}^{N} h^{(i)}(n,k) \vec{X}_n$$

$$\Sigma_k^{(i)} \leftarrow \frac{1}{\sum_{n=1}^{N} h^{(i)}(n,k)} \sum_{n=1}^{N} h^{(i)}(n,k)(\vec{X}_n - \vec{\mu}_k^{(i)})(\vec{X}_n - \vec{\mu}_k^{(i)})^T$$

Convergence Criteria

The quality metric is the Log-likelihood of the probability of obtaining the data given the parameters.

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

It can be shown that, for EM, the log likelihood will converge to a stable maximum. The change in Q will monotonically decrease. This can be used to define a halting condition:

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

Log Likelihood for a Parameter Vector

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

$$L(\vec{v} \mid \{X_n\}) = P(\{X_n\} \mid \vec{v}) = \prod_{n=1}^{N} P(X_n \mid \vec{v})$$

For normal density functions, $\vec{v} = \vec{\mu}, \Sigma$
and

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

it is more convenient to work with the Log-Likelihood

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

Maximum Likelihood Estimators

A Maximum Likelihood Estimator (MLE) can be used to derive the most likely values for the parameters a Gaussian Density.

To illustrate, consider that case of Univariate Gaussian Density function (D=1).

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

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

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

The maximum of the Log-Likelihood occurs when the derivative is zero.

$$\frac{\partial l(\vec{v})}{\partial \mu} = \sum_{n=1}^{N} \frac{1}{\sigma^2}(X_n - \mu) = 0$$

$$\frac{\partial l(\vec{v})}{\partial \sigma^2} = \sum_{n=1}^{N} \left(-\frac{1}{2\sigma^2} + \frac{(X_n - \mu)^2}{2\sigma^4} \right) = 0$$

We can formulate this as a gradient

$$\nabla_{\mu,\sigma} \mathcal{L}(\vec{v}) = \begin{pmatrix} \frac{\partial l(\vec{v})}{\partial \mu} \\ \frac{\partial l(\vec{v})}{\partial \sigma^2} \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{N} \frac{1}{\sigma^2}(X_n - \mu) \\ \sum_{n=1}^{N} \left(-\frac{1}{2\sigma^2} + \frac{(X_n - \mu)^2}{2\sigma^4} \right) \end{pmatrix} = 0$$

and with a little algebra discover that

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} X_n$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^{N} (X_n - \hat{\mu})^2$$

(here is the algebra).

$$\frac{\partial l(\vec{v})}{\partial \mu} = \sum_{n=1}^{N} \frac{1}{\sigma^2}(X_n - \hat{\mu}) = 0$$

$$\frac{1}{\sigma^2}\sum_{n=1}^{N} X_n = \frac{1}{\sigma^2}\sum_{n=1}^{N} \hat{\mu}$$

$$\sum_{n=1}^{N} X_n = \sum_{n=1}^{N} \hat{\mu} = N\hat{\mu}$$

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} X_n$$

In the same way

$$\frac{\partial l(\vec{v})}{\partial \sigma^2} = \sum_{n=1}^{N}\left(-\frac{1}{2\hat{\sigma}^2} + \frac{(X_n - \hat{\mu})^2}{2\hat{\sigma}^4}\right) = 0$$

$$\sum_{n=1}^{N}\left(-\frac{1}{2\hat{\sigma}^2} + \frac{(X_n - \hat{\mu})^2}{2\hat{\sigma}^4}\right) = 0$$

$$\sum_{n=1}^{N}\frac{1}{2\hat{\sigma}^2} = \sum_{n=1}^{N}\frac{(X_n - \hat{\mu})^2}{2\hat{\sigma}^4}$$

$$\frac{1}{2\hat{\sigma}^2}\sum_{n=1}^{N} 1 = \frac{1}{2\hat{\sigma}^2}\sum_{n=1}^{N}\frac{(X_n - \hat{\mu})^2}{\hat{\sigma}^2}$$

$$\sum_{n=1}^{N} 1 = \sum_{n=1}^{N}\frac{(X_n - \hat{\mu})^2}{\hat{\sigma}^2}$$

$$N = \frac{1}{\hat{\sigma}^2}\sum_{n=1}^{N}(X_n - \hat{\mu})^2$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(X_n - \hat{\mu})^2$$

The same can be done for D > 1, however the algebra is a bit more complex

Maximum Likelihood for a Multivariate Density Function

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

$$\vec{v} = (\alpha, \vec{\mu}, \Sigma)$$

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

$$\hat{v} = \max_v \{\prod_{n=1}^{N} P(\vec{X}_n \mid \vec{v})\} = \max_v \{\sum_{n=1}^{N} Log(P(\vec{X}_n \mid \vec{v}))\}$$

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

$$\nabla_{\mathbf{v}} \mathcal{L}(\vec{v}) = \nabla_{\mathbf{v}} \sum_{n=1}^{N} Log(P(\vec{X}_n \mid \vec{v})) = 0$$

$\nabla_{\mathbf{v}}$ is the gradient operator: $\nabla_v = \begin{pmatrix} \frac{\partial}{\partial v_1} \\ \frac{\partial}{\partial v_2} \\ \dots \\ \frac{\partial}{\partial v_D} \end{pmatrix}$

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

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

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

Notice that the MLE for the covariance is biased. Why?