Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2016/2017

Lesson 11 17 march 2017

Clustering and non-supervised learning with K-Means and EM

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 $\vec{X}$
k	index for cluster, data source or GMM Mode
K	Total number of clusters, or sources, of events
M	Total number of sample events.
	$M = \sum_{k=1}^{K} M_k$
$\{\vec{X}_m\}$	A set of M Sample Observations (a training set)
$\{\vec{y}_m\}$	A set of indicator vectors for the training samples in $\{\vec{X}_m\}$
	$\vec{y}_m$ indicates the source S_k for each training sample $\vec{X}_m$
Note that	$\vec{y}_m$ can be a binary vector with k rows (1 for S_k and 0 for others) or
	$\vec{y}_m$ can be the probability that $\vec{X}_m \in S_k$

$h(k,m) = \left(\vec{y}_1 \quad \cdots \quad \vec{y}_m\right)$ Indicator variables in matrix form. k rows, m columns

Expected Value: $E\{X\} = \frac{1}{M}\sum_{m=1}^{M} X_m$

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 $\vec{\mu}$, Σ 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(ω)* is a value, *p(X)* is a function.

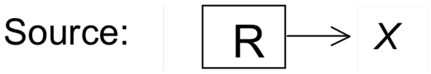
The mean is $\vec{\mu} = E\{\vec{X}\} = \begin{pmatrix} E\{X_1\} \\ E\{X_2\} \\ ... \\ E\{X_D\} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ ... \\ \mu_D \end{pmatrix}$

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

Gaussian Mixture Models

Gaussian Mixtures as a Sum of Independent Sources

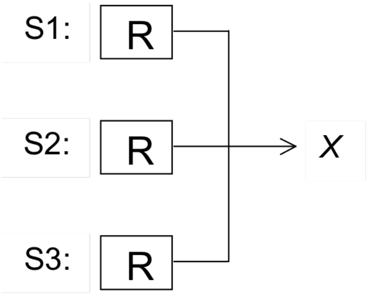
We can consider a sequence of random trials as a "source" of event



The central limit theorem tells us that in this case, the sum of many independent, identically distributed random variable will converge to a 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, each with it own unique independent random variables and Normal Density function.



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 (GMM).

A GMM can be used to represent density functions multiple sources. 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, indicated by the coefficients α_k.
We can see the coefficients $\{\alpha_k\}$ 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 $\sum_{k=1}^{K}\alpha_k = 1$

Thus the α_k form a probability Distribution.

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(k,m)* that assigns each sample to one of K clusters or sources.

Thus an algorithm for estimating *h(k, m)* can be used for unsupervised learning.

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

This can be seen as equivalent to the indicator variable $\vec{y}_m$

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

h(k, m) = 1 if $\vec{X}_m$ is assigned to cluster <u>k</u>, 0 otherwise.

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

$$h(k,m) = P(X_m \in S_k)$$

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 sensitive and will converge to the global best estimate.

K-Means and EM can be used to discover the classes for each training sample, and are thus used for <u>Unsupervised Learning</u>.

They can also be used to estimate a multimodal density for a single class.

K-Means Clustering

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

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

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, $\vec{X}_m$, 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(k,m)*.

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.

$$Q^{(i)} = \sum_{m=1}^{M} \sum_{k=1}^{K} h^{(i)}(m,k)(\vec{X}_m - \vec{\mu}_k^{(i)})^T \Sigma_k^{(i)-1}(\vec{X}_m - \vec{\mu}_k^{(i)})$$

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

```
$i \leftarrow i+1$
$\forall m = 1, M$ :
    $\forall k = h^{(i)}(k,m) = 0$
    k $= \underset{k}{\arg-\min}\left\{(\vec{X}_m - \vec{\mu}_k)^T \Sigma_k^{-1} (\vec{X}_m - \vec{\mu}_k)\right\}$
    $h^{(i)}(k,m) \leftarrow 1$
```

Maximization

Mass: $M_k = \sum_{m=1}^{M} h^{(i)}(k,m)$ is the number of samples attributed to source k.

If $M_k \neq 0$:

Mean: $\mu_k^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} h^{(i)}(k,m) \cdot \vec{X}_m$

Covariance: $\Sigma_k^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} h^{(i)}(k,m) \cdot (\vec{X}_m - \vec{\mu}_k)(\vec{X}_m - \vec{\mu}_k)^T$

That is, for each component of the covariance, $\sigma_{ij}^{(i)}$:

$$\sigma_{ij}^{2(i)} = \frac{1}{M_k} \sum_{m=1}^{M} h^{(i)}(k,m) \cdot (x_{mi} - \mu_{ki})(x_{mj} - \mu_{kj})$$

At the end of each cycle:

Quality: $Q^{(i)} = \sum_{m=1}^{M} \sum_{k=1}^{K} h^{(i)}(m,k)(\vec{X}_m - \vec{\mu}_k^{(i)})^T \Sigma_k^{(i)-1} (\vec{X}_m - \vec{\mu}_k^{(i)})$

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 *M* sample observations $\{\vec{X}_m\}$, with each observation drawn from one of K sources S_k. Our problem is to discover an assignment table *h(k, m)* that assigns each observation, $\vec{X}_m$ 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(k,m)*.
To estimate *h(k,m)* 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(k,m)*. and then $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$.

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

$$h(k,m) = P(\vec{X}_m \in S_k)$$

Initialization:

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

set *i=0.*

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

This can be initialiZed 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 $h^{(i)}(k,m)$ using the training data and estimated parameters.

$$h^{(i)}(k,m) = P(\vec{X}_m \in S_k \mid \{X_m\}, \vec{v}^{(i-1)})$$

which gives :

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

Maximization Step (M)

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

Mass: $M_k^{(i)} \leftarrow \sum_{m=1}^{N} h^{(i)}(k,m)$ (Note: M_k is a real)

Probability: $\alpha_k^{(i)} \leftarrow \frac{1}{M} \sum_{m=1}^{M} h^{(i)}(k,m) = \frac{M_k^{(i)}}{M}$

Mean: $\vec{\mu}_k^{(i)} \leftarrow \frac{1}{M_k^{(i)}} \sum_{m=1}^{M} h^{(i)}(k,m) \vec{X}_m$

Covariance: $\Sigma_k^{(i)} \leftarrow \frac{1}{M_k^{(i)}} \sum_{m=1}^{M} h^{(i)}(k,m)(\vec{X}_m - \vec{\mu}_k^{(i)})(\vec{X}_m - \vec{\mu}_k^{(i)})^T$

Convergence Criteria

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

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

It can be shown that, for EM, the log likelihood will converge to a stable maximum. The change in Q will monotonically decrease. 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, $\{X_m\}$ 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, $\vec{v} = \vec{\mu}, \Sigma$ 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

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

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.

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

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

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

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

We can formulate this as a gradient

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

and with a little algebra discover that

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

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

(here is the algebra).

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

$$\frac{1}{\sigma^2}\sum_{m=1}^{M} X_m = \frac{1}{\sigma^2}\sum_{m=1}^{M} \hat{\mu}$$

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

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

In the same way

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

$$\sum_{m=1}^{M}\left(-\frac{1}{2\hat{\sigma}^2} + \frac{(X_m - \hat{\mu})^2}{2\hat{\sigma}^4}\right) = 0$$

$$\sum_{m=1}^{M}\frac{1}{2\hat{\sigma}^2} = \sum_{m=1}^{M}\frac{(X_m - \hat{\mu})^2}{2\hat{\sigma}^4}$$

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

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

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

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

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.

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

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

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

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: $\nabla_v = \begin{pmatrix} \frac{\partial}{\partial v_1} \\ \frac{\partial}{\partial v_2} \\ \ldots \\ \frac{\partial}{\partial v_D} \end{pmatrix}$

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

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?