Pattern Recognition and Machine Learning

James L. Crowley

ENSIMAG 3 MMIS First Semester 2010/2011

Lesson 4 3 November 2010

Estimating Parameters for a Gaussian

Contents

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

Notation

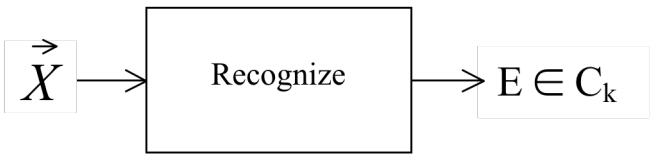
x	a variable
X	a random variable (unpredictable value)
$\vec{x}$	A vector of D variables.
$\vec{X}$	A vector of D random variables.
D	The number of dimensions for the vector $\vec{x}$ or $\vec{X}$
E	An observation. An event.
k	Class index
K	Total number of classes
C_k	The kth class.
ω_k	The statement (assertion) that E $\in C_k$
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.
$\{X_m\} = \bigcup_{k=1,K} \{X_m^k\}$	
$\{t_m\}$	A set of class labels (indicators) for the samples
$\mu = E\{X_m\}$	The Expected Value, or Average from the M samples.
$\hat{\sigma}^2$	Estimated Variance
$\tilde{\sigma}^2$	True Variance
$\mathcal{N}(x;\, \mu,\, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \; e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	Gaussian (Normal) Density function.

Reminder: The Pattern Recognition Problem

Assume that we have a sensor that produces discrete observations of the world. Each observation is an event, E. Assume that for each observation, the sensor provides a vector of D features, $\vec{X}$

Observation: $(E, \vec{X})$

Our problem is to build a box that assigns each observation to one of K classes $\{C_k\}$ labeled k=1 to K.



This problem is known as "Decision Theory". $\hat{\omega}_k = decide(E \in C_k)$

We can decompose this into two component functions $d()$ and $y(\vec{X})$:

$$\hat{\omega}_k \leftarrow d(y(\vec{X}))$$

Where $y(\vec{X})$ is a discriminant function that maps $R^D \rightarrow R^K$
d() is a decision function d(): $R^K \rightarrow \{\hat{\omega}_k\}$

Generally we choose d() to make as few mistakes as possible.
We can express this mathematically using probability theory as:

$$\hat{\omega}_k = \arg\max_{\omega_k} \{p(\omega_k \mid \vec{X})\}$$

In this case, our primary tool is Bayes Rule, that tells us:

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

Alternatively, we may choose d() to minimize the cost of a mistake.
We will examine alternative decision functions in the next few lectures.

Generative vs Discriminative Approaches

Generally, there are three approaches to building a decision function:

1) Generative Approach: Construct an explicit estimate of the probabilities $p(\vec{X} \mid \omega_k)$, $p(\vec{X})$ and $p(\omega_k)$ and use Bayes rule to compute the most likely decision.

2) Discriminative Approach: Use Bayesian theory to construct a function that partitions the feature space $\vec{X}$ into discrete regions for each class K.

3) Ad-hoc approach: "invent" an arbitrary theory of tests. This generally gives unreliable results.

Generative approach is more general, but imposes restrictions. In many cases a discriminative approach can provide a more practical solution. We will examine both in this course.

The Gaussian Assumption

The Gaussian assumption underlies many learning techniques.
If we assume that the

1) the feature vector $\vec{X}$ for each class k is generated by a stationary random process, and

2) The process for generating $\vec{X}$ is composed of a sequence of independent random events, then, then probability theory tells us that

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

where $\vec{\mu}_k$ is the average (center of gravity) of the features for class k.
and Σ_k is the co-variance of the features for class k.

We can use a training set to "learn" a discriminant function, $y_k(x)$, for each class as:

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

using a labeled set of training samples $\{\vec{X}_m\}$ for which we know the class labels $\{t_m\}$. The class labels partition the training set $\{\vec{X}_m\}$ into K subsets $\{X_m^k\}$ each of which contains M_k samples, so that $\{X_m\} = \bigcup_{k=1,K} \{X_m^k\}$ and $M = \sum_{k=1}^{K} M_k$

In this case we can use the training data to estimate the parameters $\vec{\mu}_k, \Sigma_k$ for

$$p(\vec{X} \mid \omega_k) = \mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)$$

We can estimate that $p(\omega_k) = \frac{M_k}{M}$

We can note that for the training data

$$p(\vec{X}) = \sum_{k=1}^{K} p(\vec{X} \mid \omega_k) \text{ and thus } p(\vec{X}) = \sum_{k=1}^{K} \mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)$$

Equivalently we can estimate $p(\vec{X}) = \mathcal{N}(\vec{X} \mid \vec{\mu}, \Sigma)$ from the training data $\{\vec{X}_m\}$

But note, even if the Gaussian assumption holds for each class, there is not assurance that P(X) is Gaussian!

Thus $p(\omega_k \mid \vec{X}) = \frac{\mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)}{\mathcal{N}(\vec{X} \mid \vec{\mu}, \Sigma)} \cdot \frac{M_k}{M}$ or if we prefer $p(\omega_k \mid \vec{X}) = \frac{\mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)} \cdot \frac{M_k}{M}$

Note that $p(\vec{X}) = \mathcal{N}(\vec{X} \mid \vec{\mu}, \Sigma)$ is the same for all classes.
If we are only interested in deciding among the K choices then

$$\hat{\omega}_k = \underset{\omega_k}{\arg-\max}\{p(\omega_k \mid \vec{X})\} = \underset{k}{\arg-\max}\{\frac{\mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)}{\mathcal{N}(\vec{X} \mid \vec{\mu}, \Sigma)} \cdot \frac{M_k}{M}\} = \underset{k}{\arg-\max}\{\mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k) \cdot \frac{M_k}{M}\}$$

However, $\vec{X}$ might be a value that was not seen in the training data, in which case, our decision is not reliable.

We can reduce the number of errors by adding a "Reject Option", refusing to decide if $p(\omega_k \mid \vec{X}) \leq$ Threshold.

In this case $p(\omega_k \mid \vec{X})$ provides an estimate of the "confidence" of the decision.

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

To apply the Gaussian assumption we need some way to estimate the Gaussian parameters.

Observation Error

Note that in addition to the within class variation, Σ_k, observations can be corrupted by "observation noise", with a covariance β. Observation noise is generally zero mean (unbiased). Otherwise we estimate the bias and subtract it from the observation!

Generally the observation noise is "uncorrelated" (represented by a diagonal covariance matrix), and independent of the class.

$$\beta = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D^2 \end{bmatrix}$$

The ratio of the energy of the class covariance det(Σ_k) to the observation noise det(β) will often determine the choise of recognition method.

This will also complicate our formulas for estimating $p(\vec{X} \mid \omega_k) = \mathcal{N}(\vec{X} \mid \vec{\mu}_k, \Sigma_k)$.

Multivariate Gaussian Density Functions

Assume a feature vector $\vec{X}$ of D random variables

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

The parameters $\vec{\mu}$ (mean) and Σ (covariance) are obtained from a training set $\{\vec{X}_m\}$ of M sample observations.

In the case where, we have K classes, C_k, k=1, …, K, we can estimate conditional densities for each class using a labeled training set. Commonly the labels will be in the form an "indicator" variable $\{t_m\}$ for each Sample $\{\vec{X}_m\}$.

Alternatively, we can partition the training set $\{\vec{X}_m\}$ into K subsets $\{X_m^k\}$ each of which contains M_k samples.

In this case, $\{X_m\} = \bigcup_{k=1,K} \{X_m^k\}$ and $M = \sum_{k=1}^{K} M_k$

$$\vec{\mu} = E\{\vec{X}\} = \frac{1}{M} \sum_{m=1}^{M} \vec{X} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_D \end{pmatrix} = \begin{pmatrix} E\{X_1\} \\ E\{X_2\} \\ \dots \\ E\{X_D\} \end{pmatrix}$$

For D dimensions, the second moment is a co-variance matrix composed of D^2 terms:

$$\sigma_{ij}^2 = E\{(X_i - \mu_i)(X_j - \mu_j)\} = \frac{1}{M} \sum_{m=1}^{M} (X_{im} - \mu_i)(X_{jm} - \mu_j)$$

This is often written

$$\Sigma = E\{(\vec{X} - E\{\vec{X}\})(\vec{X} - E\{\vec{X}\})^T\}$$

and gives $$\Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1D}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2D}^2 \\ \dots & \dots & \dots & \dots \\ \sigma_{D1}^2 & \sigma_{D2}^2 & \dots & \sigma_{DD}^2 \end{pmatrix}$$

Linear Algebraic Form for Moment Calculation

Calculation of the mean and covariance are often expressed using linear algebra. Such expressions are widely used in machine learning.

As before, assume a set of M training examples $\{X_m\}$

Recall

$$\vec{\mu} = E\{\vec{X}\} = \frac{1}{M}\sum_{m=1}^{M}\vec{X} = \begin{pmatrix}\mu_1\\ \mu_2\\ \dots\\ \mu_D\end{pmatrix} = \begin{pmatrix}E\{X_1\}\\ E\{X_2\}\\ \dots\\ E\{X_D\}\end{pmatrix}$$

Let us define $\vec{V}_m = \vec{X}_m - E\{\vec{X}_m\} = \vec{X}_m - \vec{\mu}_m$

and thus

$$\Sigma_x = E\{\vec{V}\vec{V}^T\}$$

We can compose a matrix with M columns and D rows from $\{V_m\}$.

$$V = \begin{pmatrix}v_{11} & v_{12} & \dots & v_{1M}\\ v_{21} & v_{22} & \dots & v_{2M}\\ \dots & \dots & & \dots\\ v_{D1} & v_{D2} & \dots & v_{DM}\end{pmatrix}$$

This can be used to write

$$\Sigma_x \equiv V\,V^T = \begin{bmatrix}\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet\end{bmatrix}\begin{bmatrix}\bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\end{bmatrix} = \begin{bmatrix}\bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\\ \bullet & \bullet & \bullet & \bullet\end{bmatrix}$$

$\Sigma_x \equiv V\,V^T$ is a D x D matrix that captures the "co-variance" of the elements of i,j of the vector X in $\{X_m\}$ Note that we can also write $\Sigma_m = V^T\,V$ of size M x M.

Linear Transforms of the Normal Multivariate Density

The Normal (Gaussian) function is a defined only by its moments.
It is thus invariant to transformations of its moments, that is affine transformations.
The affine transformations include rotation, translation, scale changes and other linear transformations.

For example consider a rotation (cosine) vector $\vec{R}$, such that $\| \vec{R} \| = 1$.

A vector $\vec{X}$ may be rotated by $\vec{R}$

$$\vec{Y} = \vec{R}^T \vec{X}$$

For the covariance: $\Sigma_Y = E\{(\vec{R}^T\vec{V})(\vec{R}^T\vec{V})^T\}$

$$= E\{(\vec{R}^T\vec{V})(\vec{V}^T\vec{R})\}$$
$$= E\{(\vec{R}^T(\vec{V}\vec{V}^T)\vec{R})\}$$
$$= E\{(\vec{R}^T\Sigma_X\vec{R})\}$$

$$(\vec{R}^T\ \vec{V})^T = (\vec{V}^T\ \vec{R})$$

Thus rotation of a covariance requires pre and post multiplication by $\vec{R}$.

Projection of a Gaussian is the Gaussian of the projection.

$$\mu_y = \vec{R}^T \vec{\mu}_x, \qquad \sigma_y^2 = \vec{R}^T \Sigma_x \vec{R}$$

$$p(y) = \mathcal{N}(y;\ \vec{R}^T\ \vec{\mu}_x, \vec{R}^T\Sigma_x\vec{R}^T) = \mathcal{N}(y; \mu_y, \sigma_y^2)$$

This can be computed by projecting the moments or by projecting the data and recomputing the moments.

$$\mu_y = E\{P(Y))\} = \vec{R}^T\ \vec{\mu}_x \qquad \sigma_y^2 = E\{(p(Y)-\mu_y)\,(p(Y)-\mu_y)\} = \vec{R}^T \Sigma_x \vec{R}$$

Likelihood Estimation for pdf parameters

To keep the discussion simple, consider D=1 and assume that we have a training set $\{X_m\}$ of M examples of a Gaussian density $\mathcal{N}(x;\ \mu,\sigma)$.

We wish to estimate the most likely (believable) parameters (μ,σ) from $\{X_m\}$.

We note that $p(\{X_m\}\mid\mu,\sigma)=\prod_{m=1}^{m}\mathcal{N}(X_m\mid\mu,\sigma)$

Simple case: For a normal density with D=1, the parameters are

$$\vec{\nu} = (\mu, \sigma)$$

Our best estimate of $\nu = (\mu, \sigma)$ is that which maximizes the probability for the training data $\{X_m\}$

Let us define the Likelihood for the parameters of the pdf as

$$L(\nu \mid X_1, X_2, ..., X_M)$$

Assuming that the X_m are independent,

$$P(X_1, X_2 \mid \vec{\nu}) = P(X_1 \mid \vec{\nu})\cdot P(X_2 \mid \vec{\nu})$$

in general for M events:

$$P(X_1,X_2,...,X_M\mid\vec{\nu})=P(\{X_m\}\mid\vec{\nu})=\prod_{m=1}^{M}\mathcal{N}(X_m\mid\vec{\nu})$$

We define the likelihood of ν given $\{X_m\}$ as

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

Our objective is to estimate $\hat{\nu}$ to maximise $L(\hat{\nu}\mid\{X_m\})$

$$\hat{\nu}=\arg\max_{\nu}\{L(\hat{\nu}\mid\{X_m\})\}=\arg\max_{\nu}\{\prod_{m=1}^{M}\mathcal{N}(X_m\mid\hat{\nu})\}$$

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

it is easier to work with the Log likelihood:

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

$P(X_m \mid \hat{v})$ is a Gaussian, but not necessarily a pdf. The integral may not be 1.

MLE for a Gaussian pdf from a sample set

For D=1, $\mathcal{N}(X; \mu, \sigma)$ the parameter 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

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

with a little algebra:

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

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

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

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

In the same way

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

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

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

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

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

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

Notice that the Maximum likelihood gives a "biased" estimate for σ^2.

The unbiased estimate would be:

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

Thus the unbiased estimate is related to the ML estimate by

$$\sigma^2 = \frac{M}{M-1}\hat{\sigma}^2$$