Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2010/2011

Lesson 14 6 april 2011

Non-parametric Methods for Classification

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.

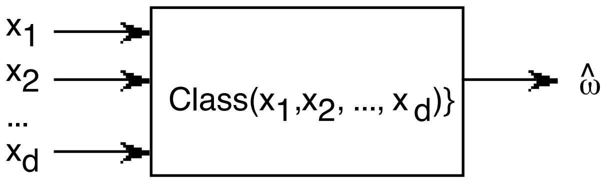
Notation

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.
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. $\omega_k = E \in T_k$.
h(n)	A histogram of random values for the feature n.
$h_k(n)$	A histogram of random values for the feature n for the class k. $h(x) = \sum_{k=1}^{K} h_k(x)$
Q	Number of cells in $h(n)$. $Q = N^D$
P	A sum of V adjacent histogram cells: $P = \sum_{\vec{X} \in V} h(\vec{X})$

Non-Parametric Methods for classification

Bayesian Classification (Reminder)

Our problem is to build a box that maps a set of features $\vec{X}$ from an Observation, E into a class C_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 $\omega_k \equiv E \in T_k$

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

Our primary tool for this is Baye's Rule :

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

To apply Baye's rule, we require a representation for the probalities $P(\vec{X} \mid \omega_k)$, $P(\vec{X})$, and $p(\omega_k)$.

The term $p(\omega_k)$ is a number that represents the a-priori probability of encountering an event of class K. For a training set of of M samples of which M_k are from class k, this is simply the frequency of occurrence of class k.

$$p(\omega_k) = \frac{M_k}{M}$$

The terms $P(\vec{X} \mid \omega_k)$, $P(\vec{X})$ are more subtle.
Today will look at three non-parametric representations for $P(\vec{X} \mid \omega_k)$ and $P(\vec{X})$

1) Ratio of Histograms
2) Kernel Density Estimators
3) K-Nearest Neighbors

Histogram Representation for a Bounded Integer

To estimate the probability of a value, one easy method it to count the number of times it occurs. For this we can use a table of "frequency of occurrence", also known as a "histogram", h(x).

To use a histogram to build a non-parametric representation for numerical featuresthe set of possible values for the feature must be finite. That is, each feature value must be represented by an integer x from a finite range:

$$x \in [x_{min}, x_{max}].$$

In many problems this occurs naturally. For example: the age, height, weight of a person, grades in a class, amount of change in a purse. In other cases, we can map the feature into a finite range.

For convenience, we will map features to integer values in the range $x \in [1, N]$,

If X is integer, with $x \in [x_{min}, x_{max}]$ we need only subtract x_{min}.

$$x := x - x_{min}.$$

We can then estimate the probability p(X) using a training set $\{X_m\}$.

Given a training set $\{X_m\}$ of features from M events, such that $x \in [1, N]$, we can build a table of frequency for the values of X. We allocate a table of N cells, and use the table to count the number of times each value occurs:

$$\forall m=1, M \ : \ h(X_m) := h(X_m) + 1;$$

Then the probability that a feature $X \in \{X_m\}$ from this set has the value x is then

$$P(X=x) = \frac{1}{M} h(x)$$

If the
1) the sample is large enough ($M > 8 Q$, where $Q=N^D$), and
2) the observing conditions are "ergodic" (do not change with time),
then the histogram will also predict frequency of occurrence for future events.

Histograms for unbounded integer x.

If x is unbounded we must first bound it. We define bounds: x_{min} and x_{max}.
Then

```
    If (x < $x_{min}$) then x := $x_{min}$;
    If (x > $x_{max}$) then x := $x_{max}$;
    x :=x- $x_{min}$.
```

Histograms for real x.

If X is real, we must quantize it with a function such as “trunc()” or “round()”. The function trunc() removes the fractional part of a number. Round() adds ½ then removes the factional part:

To quantize X to N discrete values :

For X real:

```
    If (x < $x_{min}$) then x := $x_{min}$;
    If (x > $x_{max}$) then x := $x_{max}$;
    x := x-$x_{min}$.
```

$$n = trunc\left(N \cdot \frac{x}{x_{\max} - x_{\min}}\right) + 1$$

if n > N then n=N.

This last line handles the rare case where $X=X_{max}$ and thus n=N+1.

When X is a vector of D features.

When X is a vector of D features each of the components must be normalized to a bounded integer between 1 and N. This can be done by individually bounding each component, x_d.

Assume a feature vector of D values $\vec{x}$

$$\vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_D \end{pmatrix}$$

Given that each feature $x_d \in [1, N]$, allocate a D dimensional table
$h(x_1, x_2, \ldots, x_D) = h(\vec{x})$.

The number of cells in $h(\vec{X})$ is $Q=N^D$.
As before,

$$\forall m=1, M \ : \ h(\vec{X}_m) = h(\vec{X}_m) + 1$$

Then:

$$p(\vec{X} = \vec{x}) = \frac{1}{M} h(\vec{x})$$

as we saw in the previous lecture, the average error depends on the ratio
$Q=N^D$ and M. : $E_{ms} \sim O(\frac{Q}{M})$

Example:

Suppose that we have 2 classes, k=1 and k=2, and that we observe a training set of M_1 events from class k=1: $\{\vec{X}_m^1\}$ and M_2 events from class k=2 $\{\vec{X}_m^2\}$

We assume that the feature vectors have D dimensions, each quantized to integer values in the range [1, N]. We assume stationary observing conditions with $M_1 \geq 8N^D$ and $M_2 \geq 8\ N^D$.

We build the histograms $h_1(\vec{x})$ and $h_2(\vec{x})$:

for m=1 to M_1 : $h_1(\vec{X}_m^1) := h_1(\vec{X}_m^1)+1$

for m=1 to M_2 : $h_2(\vec{X}_m^2) := h_2(\vec{X}_m^2)+1$

We also define $h(\vec{x}) = h_1(\vec{x}) + h_2(\vec{x})$ and $M = M_1 + M_2$

Thus, for a new observation, E, with features mapped to integers, then

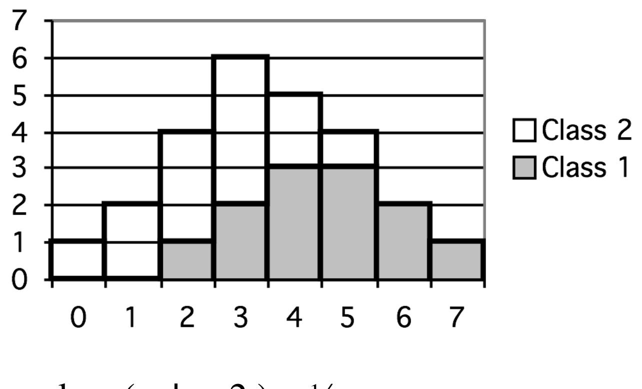
$p(\vec{X}) = \frac{1}{M}h(\vec{x})$ where $p(\vec{X})$ is shorthand for $p(\vec{X} = \vec{x})$

$p(\vec{X} \mid \omega_k) = \frac{1}{M_k}h_k(\vec{x})$

$p(E \in C_k) = p(\omega_k) = \frac{M_k}{M}$

Thus $p(\omega_1 \mid n) = \frac{p(\vec{X} \mid \omega_k)p(\omega_k)}{p(\vec{X})} = \frac{\frac{1}{M_k}h_k(\vec{x})\frac{M_k}{M}}{\frac{1}{M}h(\vec{x})} = \frac{h_k(\vec{x})}{h(\vec{x})}$

If D =1



For example, $p(\omega_1 \mid x=2) = ¼$

The probability of observing class k give feature x is $p(\omega_k \mid x) = h_k(x)/h(x)$

Histograms have the advantages:

1) They have a fixed size, Q, independent of the quantity of data. It is not necessary to store the data.
2) They can be composed and used incrementally.

The disadvantage is that

1) Each feature must be quantized over a limited range of N values.
2) We need M >> Q data samples.
3) There are discontinuities at the boundaries of each cell.

Because the $M = \sum_{\vec{X}} h(\vec{X})$ we are sure that $\sum_{\vec{X}} p(\vec{X}) = 1$

Variable size histogram cells

If the quantity of training data is too small, ie M < Q we can combine adjacent cells so as to amass enough data for a reasonable estimate.

Let us define the volume of each cell as 1.
Then the volume of the entire space is $Q=N^D$.

Suppose we merge V adjacent cells such that we obtain a combined sum of P. The volume of the combined cells would be V

$$P = \sum_{\vec{X} \in V} h(\vec{X})$$

The probability $p(\vec{X})$ for $\vec{X} \in V$ is $p(\vec{X}) = \frac{P}{MV}$

Suppose our samples $\{\vec{X}_m\}$ are drawn from a density $p(\vec{X})$.
If take a volume, V, from this density then

$$p(\vec{X}_m \in V) = \frac{P}{MV}$$

We can use this equation to develop two alternative non-parametric methods.

Fix V and determine P => Kernel density estimator.
Fix P and determine V => K nearest neighbtors.

(note in most developments the symbol "K" is used for the sum the cells. This conflicts with the use of K for the number of classes. Thus we substitute the symbol P for the sum of adjacent cells).

Kernel Density Estimators

For a Kernel density estimator, we will represent each data point with a kernel function $k(\vec{X})$.

Popular Kernel functions are
a hypercube centered of side w
a sphere of raduis w
a Gaussian of standard deviation w.

We can define the function for the hypercube as

$$k(\vec{u}) = \begin{cases} 1 & if\ |\mathrm{u_d}| \le 1/2 \ \text{ for all d} = 1,...,\mathrm{D} \\ 0 & otherwise \end{cases}$$

This is called a Parzen window.

For a position $\vec{X}$, the total number of points lying with a cube with side w will be:

$$P = \sum_{m=1}^{M} k\left(\frac{\vec{X}-\vec{X}_m}{w}\right)$$

The volume of the cube $V = \frac{1}{w^D}$.

Thus the probability $p(\vec{X}) = \frac{P}{MV} = \frac{1}{Mw^D}\sum_{m=1}^{M} k\left(\frac{\vec{X}-\vec{X}_m}{w}\right)$

The Hypercube has a discontinuity at the boundaries. We can soften this using a triangular function evaluated on a sphere.

$$k(\vec{u}) = \begin{cases} 1-2\|\vec{u}\| & if\ \|\vec{u}\| \le 1/2 \ \text{ for all d} = 1,...,\mathrm{D} \\ 0 & otherwise \end{cases}$$

Even better is to use a Gaussian kernel with standard deviation $\sigma = w$.

$$k(\vec{u}) = e^{-\frac{1}{2}\frac{\|\vec{u}\|^2}{w^2}}$$

We can note that the volume is $V = (2\pi)^{D/2} w^D$

In this case $p(\vec{X}) = \frac{P}{MV} = \frac{1}{M(2\pi)^{D/2} w^D} \sum_{m=1}^{M} k\left(\vec{X} - \vec{X}_m\right)$

This corresponds to placing a Gaussian over each point and summing the Gaussians. In fact, we can choose any function $k(\vec{u})$ such that

$k(\vec{u}) \geq 0$ and $\int k(\vec{u}) d\vec{u} = 1$

K Nearest Neighbors

For K nearest neighbors, we hold P constant and vary V. (We have used the symbol P for the number of neighbors, rather than K to avoid confusion with the number of classes).

As each data samples, $\vec{X}_m$, arrives, we construct a tree structure (such as a KD Tree) that allows us to easily find the P nearest neighbors for any point .

To compute $p(\vec{X})$ we P by volume of the sphere in D dimensions.

$$V = C_D \left\| \vec{X} - \vec{X}_K \right\|^D$$

where

$$C_D = + \frac{\pi^{\frac{D}{2}}}{\Gamma\left(\frac{D}{2}+1\right)}$$

Then as before:

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