Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2012/2013

Lesson 19 19 April 2013

Linear Detectors and Boosted Learning

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.

Linear Classifiers as Pattern Detectors

Linear classifiers are widely used to define pattern "detectors". This is used in computer vision, for example to detect faces or publicity logos, or other patterns of interest.

In the case of pattern detectors, K=2.

Class k=1: The target pattern.
Class k=2: Everything else.

The detector is learned form a set of training data training composed of M sample observations $\{\vec{X}_m\}$ where each sample observation is labeled with an indicator variable

$y_m = +1$ for examples of the target pattern (class 1)
$y_m = -1$ for all other examples.

Our goal is to build a hyper-plane that provides a best separation of class 1 from class 2. The hyper plane has the form:

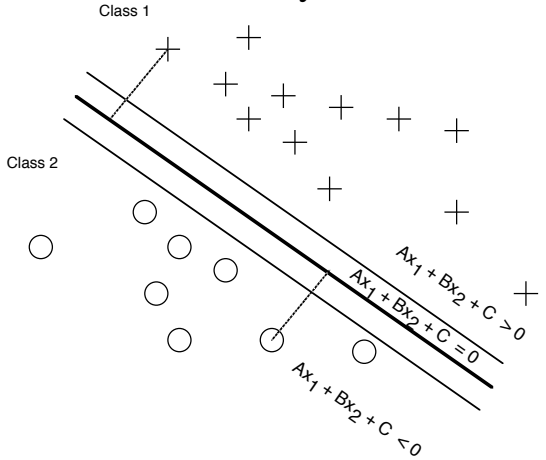
$$\vec{W}^T\vec{X} + B = 0$$

A hyperplane is a set of points such that

$$w_1x_1 + w_2x_2 + ... + w_Dx_D + B = 0$$

The decision rule is IF $\vec{W}^T\vec{X} + B > 0$ THEN $E \in C_1$ else $E \notin C_1$

B is an adjustable gain that sets the sensitivity of the detector.



Note that $\vec{W} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{pmatrix}$ is the normal to the hyperplane.

When $\vec{W}$ is normalized to unit length, $\|\vec{W}\| = 1$, then

$B = -\vec{W}^T \vec{X}$ is the perpendicular distance to the origin.

if $\|\vec{W}\| \neq 1$ then normalize as $\vec{W}' = \frac{\vec{W}}{\|\vec{W}\|}$ and $B' = \frac{B}{\|\vec{W}\|}$

A variety of techniques exist to calculate the plane. The best choice can depend on the nature of the pattern class as well as the nature of the non-class data.

ROC Curves

Two-class linear classifiers are practical for many problems. Among other uses, they provide the optimal solution to many signal detection problems in communications theories. In the case of radio communications, the noise is typically additive, Gaussian and independent of the signal, and the Bayesian Classifier reduces to a linear classifier.

Historically two class linear classifiers have been used to demonstrate optimality for some signal detection methods. The quality metric that is used is the Receiver Operating Characteristic curve. This curve should be used to describe or compare any method for signal or pattern detection.
We can bias the classifier to one or the other class by adjusting the Bias term B.

$$y(\vec{X}) = \vec{W}^T \cdot \vec{X} + B$$

B is a free variable that can be swept through a range of values.
Changing B changes the ratio of true positive detection to false detections.
This is illustrated by the Receiver Operating Characteristics (ROC) curve.

The ROC plots True Positive Rate (TPR) against False Positive Rate (FNR) as a function of B for the training data $\{\vec{X}_m\}$, $\{y_m\}$.

Les us define a detection as either Positive (P) or Negative (N)

IF $\vec{W}^T\vec{X}_m + B > 0$ THEN P else N

The detection can be TRUE (T) or FALSE (F) depending on the indiactor y_m

IF $y_m(\vec{W}^T\vec{X}_m + B) > 0$ THEN T else F

Combining these two values, any detection can be a True Positive (TP), False Positive (FP), True Negative (TN) or False Negative (FN).

For the M samples of the training data $\{\vec{X}_m\}$, $\{y_m\}$ let us define:
#P as the number of Positives,
#N as the number of Negatives,
#T as the number of True and
#F as the number of False,

From this we can define

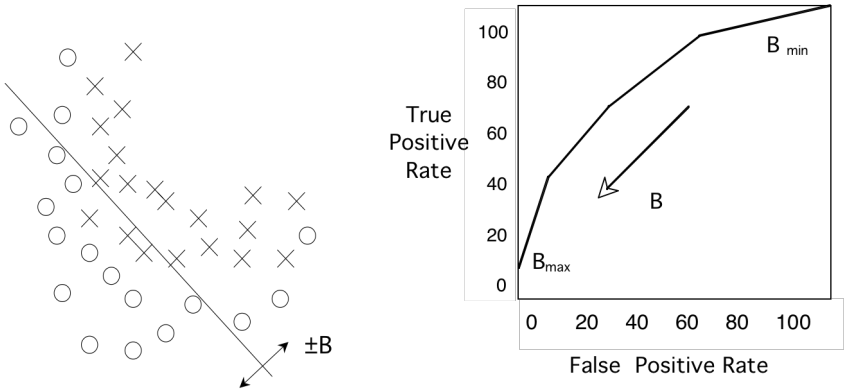
#TP as the number of True Positives,
#FP as the number of False Positives,
#TN as the number of True Negative,
#FN as the number of False Negatives.

Note that #P = #TP + #FN
And #N = #FP+ #TN

The True Positive Rate (TPR) is $TPR = \frac{\#TP}{\#P} = \frac{\#TP}{\#TP+\#FN}$

The False Positive Rate (FNR) is $FPR = \frac{\#FP}{\#N} = \frac{\#FP}{\#FP+\#TN}$

The ROC plots the TPR against the FPR as B is swept through a range of values.



When B is Large, all the samples are detected as N, and both the TPR and FPR are 0.
As B decreases both the TPR and FPR increase. Normally TPR is larger than FPR.
If TPR and FPR are equal, then the detector is no better than chance.

The more the curve approaches the upper left corner the better the detector.
The ROC is a powerful descriptor for the "goodness" of a linear classifier.
For a target class C_1 a Positive (P) detection is the decision that $E \in C_1$
a Negative (N) detection is the decision that $E \in C_2$

		$y_m(\vec{W}^T\vec{X}_m + B) > 0$	
$\vec{W}^T\vec{X}_m + B > 0$		T	F
	P	True Positive (TP)	False Positive (FP)
	N	False Negative (FN)	True Negative (TN)

Least squares estimation of a hyperplane

Assume a training set of M labeled training samples $\{y_m, \vec{X}_m\}$
such that $y_m=+1$ for class 1 and $y_m = -1$ for class 2.

Our goal is to determine a discriminant function $g(\vec{X}) = \vec{W}^T \vec{X} + b$

which can also be expressed as : $g(\vec{X}) = \vec{X}^T \vec{W} + b$

We seek the "best" $\vec{W}$. This can be determined by minimizing a "Loss" function:

$$L(\hat{W}) = \sum_{m=1}^{M} (y_m - \vec{X}_m^T \hat{W})^2$$

To build or function, we will use the M training samples to compose a matrix **X** and a vector **Y**.

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \cdots & \cdots & & \cdots \\ x_{D1} & x_{D2} & \cdots & x_{DM} \end{pmatrix}$$ (D row by M columns)

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix}$$ (M coefficients).

We write $L(W) = (\mathbf{Y} - \mathbf{X}^T W)^T (\mathbf{Y} - \mathbf{X}^T W)$ the square the sum of the errors

To minimize the loss function, we calculate the derivative and solve for W when the derivative is 0.

$$\frac{\partial L(W)}{\partial W} = -2\,\mathbf{X}^T\mathbf{Y} + 2\,\mathbf{X}^T\mathbf{X}\,W = 0$$

Thus : $\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\,W$

$$W = (\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{Y}$$

From this classifier an unknown event $\vec{X}$ as

if $\vec{W}^T\vec{X} + B > 0$ then $\hat{\omega}_1$ else $\hat{\omega}_2$

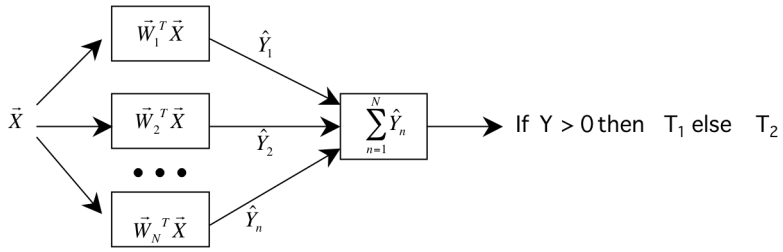
where "B" is an arbitrary "bias" constant.

We can trade False Positives for False negatives using the bias, B

A Committee of Boosted Classifiers

One of the more original ideas in machine learning the last decade is the discovery of a method by to learn a committee of classifiers by boosting. A boosted committee of classifiers can be made arbitrarily good: Adding a new classifier always improves performance.

A committee of classifiers decides by voting.



A feature vector is determined to be in the target class if the majority of classifiers vote > 0. Let us define v_i as the vote for the n^{th} classifier

For all i from 1 to I : If $\vec{W}_n^T\vec{X}_m + B > 0$ then $v_n = 1$ else $v_n = -1$.

$$\text{if } \sum_{n=1}^{N} v_i > 0 \text{ then } \hat{\omega}_1 \text{ else } \hat{\omega}_2$$

We can represent this with sgn :

$$v_n = sgn(\vec{W}_n^T\vec{X}_m + b)$$

where:

$$\text{sgn}(x) = \begin{cases} 1 & \text{if x} > 0 \\ -1 & \text{if x} \le 0 \end{cases}$$

$$\text{if } \sum_{n=1}^{N} \text{sgn}(\vec{W}_n^T\vec{X}_m + b) \text{ then } \hat{\omega}_1 \text{ else } \hat{\omega}_2$$

To learn a boosted committee we iteratively add new classifiers to the committee. In each cycle we change the data set and learn a new classifier, W_i

The data set will be changed by giving additional weight to improperly classified samples.

Learning a Committee of Classifiers with Boosting

We can iteratively apply the above procedure to learn a committee of classifiers using boosting. For this we will create a vector of "weights" a_m for each training sample. Initially, all the weights are 1.

For each cycle, the classifier $\vec{W}_n$ is learned from

$$\vec{W}_n = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\vec{A}_n^T \mathrm{I} \vec{Y})$$

Where I is the identity matrix.

After each new classifier is added, we recalculate the weights to give more weight to improperly classified training samples.

As we add classifiers, whenever a sample is miss-classified by the committee we will increase its weight so that it carries more weight in the next classifier added.

Recall the committee vote is $\sum_{n=1}^{N} \mathrm{sgn}(\vec{W}_n^T \vec{X}_m) > 0$ for class 1 (positive detection).

For m = 1 to M: if $(y_m \cdot \sum_{i=1}^{I} \mathrm{sgn}(\vec{W}_i^T \vec{X}_m + b)) < 0$ then $a_m = a_m + 1$

The result is the $(n+1)^{th}$ weight vector A_{n+1}

We then learn the $n+1^{th}$ classifier from the re-weighted set by

$$\vec{W}_{n+1} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\vec{A}_{n+1}^T I \vec{Y})$$

ROC Curve

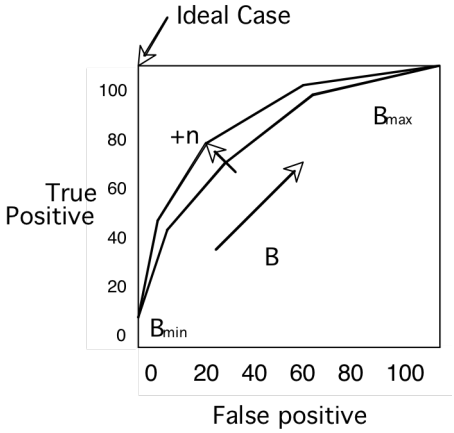
As we saw above, the ROC describes the True Positives (TP) and False Positives (FP) for a classifier as a function of the global bias b.

For m = 1 to M:

if $\sum_{n=1}^{N} \text{sgn}(\vec{W}_i^T \vec{X}_m + b) > 0$ and $y_m > 0$ then TP=TP+1

if $\sum_{n=1}^{N} \text{sgn}(\vec{W}_n^T \vec{X}_m + b) > 0$ and $y_m < 0$ then FP=FP+1

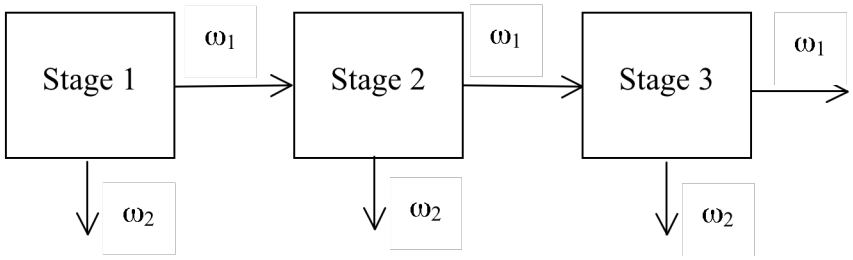
The Boosting theorem states that adding a new boosted classifier to a committee always improves the committee's ROC curve. We can continue adding classifiers until we obtain a desired rate of false positives and false negatives.



However, in general, the improvement provided for each new classifier becomes progressively smaller. We can end with a very very large number of classifiers.

Learning a Multi-Stage Cascade of Classifiers

We can optimize the computation time by using a multi-stage cascade.
With a multi-stage classifiers, only samples labeled as positive are passed to the next stage.



Each stage is applied with a bias, so as to minimize False negatives.

$$\sum_{n=1}^{N} \text{sgn}(\vec{W}_n^T \vec{X}_m + B) > 0$$

Stages are organized so that each committee is successively more costly and more discriminant.

Assume a set of M training samples $\{X_m\}$ with labels $\{y_m\}$.
Set a desired error rate for each stage j : (FP_j, FN_j).

For each stage, j, train the j+1 stage with all positive samples from the previous stage.

Each stage acts as a filter, rejecting a grand number of easy cases, and passing the hard cases to the next stage. The stages become progressively more expensive, but are used progressively less often. Globally the computation cost decreases dramatically.

Because we know the error rates for each committee we can estimate the probability of a detection based on the number of stages that an observation passes.