Computational Intelligence: Methods and Applications

Lecture 21
Linear discrimination, linear machines

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

Regression and model trees

Regression: numeric, continuous classes C(X), predict number.

Select the split to minimize variance in the node (make data piecewise constant)

$$\min_v \sum_{X\in D_v} \mathrm{Var}\big(C(X)\big)$$

Leaf nodes predict average values of training samples that reach it, so approximation is piecewise constant.

Stop criterion:
do not split the node if $\sigma(D_k) < \kappa\sigma(E)$.

Model trees: use linear regression in each node;

only a subset of attributes is used at each node.

Similar idea to the approximation by spline functions.

Some DT ideas

Many improvements have been proposed.

General idea: divide and conquer.

Multi-variate trees:
provide more complex decision borders;
trees using Fisher or Linear Discrimination Analysis;
perceptron trees, neural trees.

Split criteria:

information gain near the root, accuracy near leaves;
pruning based on logical rules, works also near the root;

Committees of trees:

learning many trees on randomized data (boosting) or CV,
learning with different pruning parameters.

Fuzzy trees, probability evaluating trees, forests of trees ...

http://www.stat.wisc.edu/~loh/ Quest, Cruise, Guide, Lotus trees

DT tests and expressive power

DT: fast and easy, recursive partitioning of data – powerful idea.

Typical DT with tests on values of single attribute has rather limited knowledge expression abilities.

For example, if N=10 people vote Yes/No, and the decision is taken when the number of Yes votes > the number of No votes (a concept: “majority are for it”), the data looks as follows:

1 0 0 0 1 1 1 0 1 0 No

1 1 0 0 1 1 1 0 1 0 Yes

0 1 0 0 1 1 1 0 1 0 No

Univariate DT will not learn from such data, unless a new test is introduced: ||X-W||>5, or W·X>5, with W=[1 1 1 1 1 1 1 1 1 1]

Another way to express it is by the M-of-N rule:

IF at least 5-of-10 (V_i=1) Then Yes.

Linear discrimination

Linear combination W·X > θ, with fixed W, defines a half-space.

W·X = 0 defines a hyperplane orthogonal to W, passing through **0**

W·X > 0 is the half-space in the direction of W vector

W·X > θ is the half-space, shifted by θ in the direction of W vector.

Linear discrimination: separate different classes of data using hyperplanes, learn the best W parameters from data.

$$\mathbf{W}^{\mathrm{T}}\cdot\mathbf{X}^{(i)}=\begin{cases}>0 & \text{for } \mathbf{X}^{(i)}\in\omega_1\\ \le 0 & \text{otherwise.}\end{cases}$$

Special case of Bayesian approach (identical covariance matrices); special test for decision trees.
Frequently a single hyperplane is sufficient to separate data, especially in high-dimensional spaces!

Linear discriminant functions

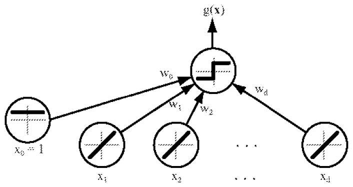
Linear discriminant function $g_{\mathrm{W}}(\mathbf{X}) = \mathbf{W}^{\mathrm{T}}\cdot\mathbf{X} + W_0$
Terminology: $\mathbf{W}$ is the weight vector, W_0 is the bias term (why?).

IF $g_{\mathrm{W}}(\mathbf{X})$ Then Class ω_1, otherwise Class ω_2

W = [W_0, W_1 ... W_d] usually includes W_0, and X=[1,X_1, .. X_d]

Discrimination function for classification may include in addition a step function $\Theta(\mathrm{W}^{\mathrm{T}}\cdot\mathrm{X}) = \pm 1$.

Graphical representation of the discriminant function $g_W(\mathrm{X})=\Theta(\mathrm{W}^{\mathrm{T}}\cdot\mathrm{X})$



One LD function may separate pairs of classes; for more classes or if strongly non-linear decision borders are needed many LD functions may be used. If smooth sigmoidal output is used LD is called a “perceptron”.

Distance from the plane

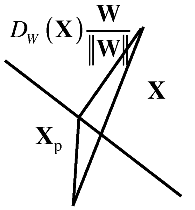
$g_{\mathrm{W}}(\mathbf{X}) = 0$ for two vectors on the *d*-D decision hyperplane means:
$\mathbf{W}^{\mathrm{T}}\cdot\mathbf{X}^{(1)} = -\mathrm{W}_0=\mathbf{W}^{\mathrm{T}}\cdot\mathbf{X}^{(2)}$, or $\mathbf{W}^{\mathrm{T}}\cdot(\mathbf{X}^{(1)}-\mathbf{X}^{(2)})=0$, so W^{T} is ⊥ (normal to) the plane. How far is arbitrary X from the decision hyperplane?

Let $\mathbf{V} = \mathbf{W}/\|\mathbf{W}\|$ be the unit vector normal to the plane and $V_0=W_0/\|\mathbf{W}\|$

$\mathbf{X}= \mathbf{X}_{\mathrm{p}}+ D_{\mathbf{W}}(\mathbf{X})\ \mathbf{V}$; but $\mathbf{W}^{\mathrm{T}}\cdot\mathbf{X}_{\mathrm{p}}=-W_0$,
therefore $\mathbf{W}^{\mathrm{T}}\cdot\mathbf{X} = -W_0+ D_{\mathbf{W}}(\mathbf{X})\ \|\mathbf{W}\|$

Hence the signed distance:

$$D_{\mathrm{W}}(\mathbf{X})=\frac{\mathbf{W}^{\mathrm{T}}\mathbf{X}+W_0}{\|\mathbf{W}\|}=\mathbf{V}^{\mathrm{T}}\mathbf{X}+V_0=\frac{g_{\mathrm{W}}(\mathbf{X})}{\|\mathbf{W}\|}$$


$D_W(\mathbf{X})\dfrac{\mathbf{W}}{\|\mathbf{W}\|}$
$\mathbf{X}$
$\mathbf{X}_{\mathrm{p}}$

Distance = scaled value of discriminant function, measures the confidence in classification; smaller $\|\mathbf{W}\|$ => greater confidence.

K-class problems

For K classes:
separate each class from the rest using K hyperplanes – but then ...

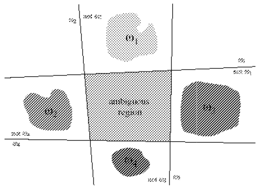
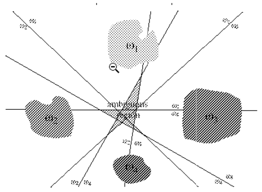


Fig. 5.3,

Duda, Hart, Stork

Perhaps separate each pair of classes using K(K-1)/2 planes?



Still ambiguous

region persist.

Linear machine

Define K discriminant functions:

$$g_i(\mathbf{X})=\mathbf{W}^{(i)\mathrm{T}}\mathbf{X}+W_{0i}\,,\ i=1\ ..\ \mathrm{K}$$

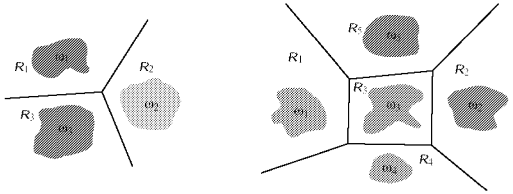
IF $g_i(\mathbf{X}) > g_j(\mathbf{X})$, for all $j\neq i$, Then select ω_i

Linear machine creates K convex decision regions R_i, ⇔ largest $g_i(\mathbf{X})$

H_{ij} hyperplane is defined by:

$$g_i(\mathbf{X}) = g_j(\mathbf{X}) \;\Rightarrow\; (\mathbf{W}^{(i)}-\mathbf{W}^{(j)})^{\mathrm{T}}\mathbf{X} + (W_{0i}-W_{0j}) = 0$$

$\mathbf{W} = (\mathbf{W}^{(i)}-\mathbf{W}^{(j)})$ is orthogonal to H_{ij} plane; distance to this plane is $D_{\mathrm{W}}(\mathbf{X})=(g_i(\mathbf{X})-g_j(\mathbf{X}))/||\mathbf{W}||$



Linear machines for 3 and 5 classes, same as one prototype + distance. Fig. 5.4, Duda, Hart, Stork

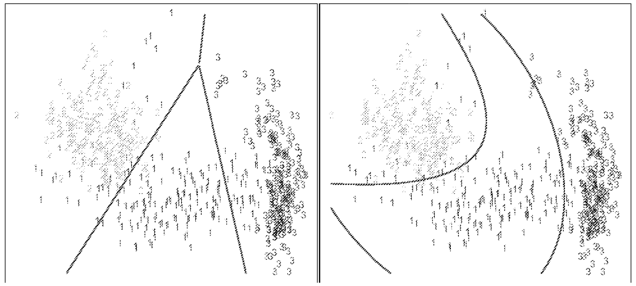
LDA is general!

Suppose that strongly non-linear borders are needed. Is LDA still useful?

Yes, but not directly in the input space!
Add to $\mathbf{X}=\{X_i\}$ input also X_i^2, and products X_iX_j, as new features.

Example: LDA in 2D => LDA in 5D adding $\{X_1,X_2,X_1^2, X_2^2, X_1X_2\}$

$g(X_1,X_2)=W_1X_1+...+W_5X_1X_2+W_0$ is now non-linear!



Hasti et al, Fig. 4.1

LDA – how?

How to find W?

There are many methods, the whole Chapter 5 in Duda, Hart & Stork is devoted to the linear discrimination methods.

LDA methods differ by:

formulation of criteria defining W;

on-line versions for incoming data, off-line for fixed data;

the use of numerical methods: least-mean square, relaxation, pseudoinverse, iterative corrections, Ho-Kashyap algorithms, stochastic approximations, linear programming algorithms ...

“Far more papers have been written about linear discriminants than the subject deserves” (Duda, Hart, Stork 2000).

Interesting papers on this subject are still being written ...

LDA – regression approach

Linear regression model (implemented in WEKA)

$$Y=g_{\mathrm{W}}(\mathbf{X})=\mathbf{W}^{\mathrm{T}}\mathbf{X}+W_0$$

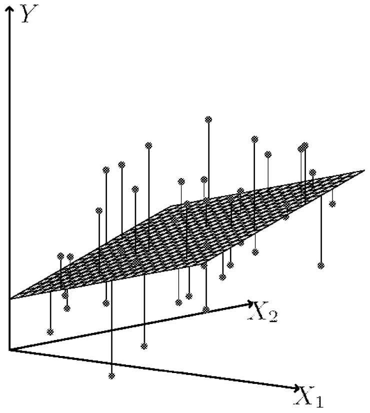
Fit the data to the known (X,*Y*) values, even if Y=±1.

Common statistical approach:

use LMS (Least Mean Square) method, minimize the Residual Sum of Squares (RSS).

$$\mathrm{RSS}(\mathbf{W})=\sum_{i=1}^{n}\left(Y^{(i)}-g_{\mathrm{W}}\left(\mathbf{X}^{(i)}\right)\right)^2$$

$$=\sum_{i=1}^{n}\left(Y^{(i)}-W_0-\sum_{i=1}^{d}W_jX_j^{(i)}\right)^2$$



LDA – regression formulation

In matrix form with X_0=1, and W_0

$$\mathbf{X}^{(i)\mathrm{T}} = \left[X_0^{(i)}, X_1^{(i)}, ..., X_d^{(i)}\right]; \quad \mathbf{X}=\left[\mathbf{X}^{(1)},...\mathbf{X}^{(n)}\right];(d+1) \text{ x } n$$

$$\mathrm{RSS}(\mathbf{W}) = \left\|\mathbf{Y}-\mathbf{X}^{\mathrm{T}}\mathbf{W}\right\|^2 = \left(\mathbf{Y}-\mathbf{X}^{\mathrm{T}}\mathbf{W}\right)^{\mathrm{T}}\left(\mathbf{Y}-\mathbf{X}^{\mathrm{T}}\mathbf{W}\right)$$

If X was square and non-singular than W = $(X^T)^{-1}$Y but $n \neq d+1$

$$\mathbf{Y}-\mathbf{X}^{\mathrm{T}}\mathbf{W} = \begin{pmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_n \end{pmatrix} - \begin{pmatrix} X_0^{(1)} & X_1^{(1)} & \cdots & X_d^{(1)} \\ X_0^{(2)} & X_1^{(2)} & \cdots & X_d^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ X_0^{(n)} & X_1^{(n)} & \cdots & X_d^{(n)} \end{pmatrix} \begin{pmatrix} W_0 \\ W_1 \\ \cdots \\ W_d \end{pmatrix}$$

LDA – regression solution

To search for the minimum of $(\mathbf{Y}-\mathbf{X}^{\mathrm{T}}\mathbf{W})^2$ put derivatives to zero:

$$\frac{\partial \mathrm{RSS}(\mathbf{W})}{\partial \mathbf{W}} = -2\mathbf{X}\left(\mathbf{Y}-\mathbf{X}^{\mathrm{T}}\mathbf{W}\right) = 0$$

$$\frac{\partial^2 \mathrm{RSS}(\mathbf{W})}{\partial \mathbf{W}^2} = 2\mathbf{X}\mathbf{X}^{\mathrm{T}} > 0$$

this is a dxd matrix, and it should be positive definite in the minimum.

Solution exist if X is non-singular matrix, i.e. all vectors are linearly independent, but if $n<d+1$ this is impossible, so sufficient number of samples is needed (there are special methods to solve it in $n<d+1$ case).

$$\mathbf{W} = \left[\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}\mathbf{X}\right]\mathbf{Y} = \left(\mathbf{X}^{\mathrm{T}}\right)^{\dagger}\mathbf{Y};$$

$$\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{I} \quad \text{but} \quad \mathbf{A}\mathbf{A}^{\dagger} \neq \mathbf{I}$$

pseudoinverse matrix has many interesting properties,
see Numerical Recipes
http://www.nr.com

LSM evaluation

The solution using the pseudoinverse matrix is one of many possible approach to LDA (for 10 other see for ex. Duda and Hart).

Is it the best result? Not always.

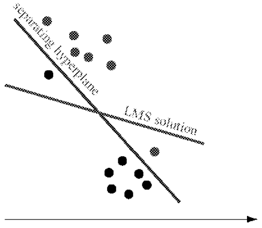
For singular X due to the linearly dependent features, the method is corrected by removing redundant features.

Good news: Least Mean Square estimates have the lowest variance among all linear estimates.

Bad news: hyperplanes found in this way may not separate even the linearly separable data!

Why? LMS minimizes squares of distances, not the classification margin.

Wait for SVMs to do that ...


separating hyperplane
LMS solution