Computational Intelligence: Methods and Applications

Lecture 22

Linear discrimination - variants

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

WEKA LDA Wine example

LDA with WEKA, using Wine data.

Classification via regression is used, regression may also be used.

In each crossvalidation regression coefficients may be quite different, discriminant functions (one per class) may use different features, ex:

- C1: 0.1391 * alcohol
- C2: -0.1697 * alcohol
- C3: 0 * alcohol

Weka "ridge" parameter - removes small weights

Good results: 2-3 errors are expected in 10x CV using a one hyperplane:

```
 a  b  c   <= classified as
59  0  0 |  a = 1
 0 68  3 |  b = 2
 0  0 48 |  c = 3
```

Not much knowledge is gained, not a stable solution .

WEKA LDA voting example

Not all LD schemes work with more than two classes in WEKA.

LDA with WEKA (choose "Classification via Regression" + Linear Regression), using Vote data: predict who belongs to democratic and who to republican party, for all 435 members of USA Congress, using results of voting on 16 issues. Largest components:

- 0.72 * physician-fee-freeze=y +
- 0.16 * adoption-of-the-budget-resolution=n +
- 0.11 * synfuels-corporation-cutback=n ... mx-missiles, ...

```
   a   b   <= classified as
 253  14 |   a = democrat
   5 163 |   b = republican
```

Overall 95.6% accuracy

Unfortunately CV and variance of CV are not so easy to calculate,

C 4.5 makes 12 (6+6) errors, same attributes as most important.

LDA conditions

Simplification of conditions defining linear discrimination.

Conditions:

$$\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)} > 0 \text{ for } \mathbf{X}^{(i)} \in \omega_1$$
$$\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)} < 0 \text{ for } \mathbf{X}^{(i)} \in \omega_2$$

using (d+1) dimensional vectors, extended by X_0=1 and W_0.
Using negative vectors for the second class leads to simpler form:

$$\mathbf{W}^{\mathrm{T}}\mathbf{X}'^{(i)} > 0 \text{ for } \mathbf{X}'^{(i)} = \mathbf{X}^{(i)} \in \omega_1$$
$$\text{and } \mathbf{X}'^{(i)} = -\mathbf{X}^{(i)} \in \omega_2$$

Instead of >0 take some small positive values > $b^{(i)}$

This will increase the margin of classification.

If b is maximized it will provide best solution.

Solving linear equations

Linear equations $\mathbf{W}^{\mathrm{T}}\mathbf{X}=\mathbf{b}^{\mathrm{T}}$ may be solved in the least-square sense using the pseudoinverse matrix, like in the regression case.

$$\mathbf{X}^{\mathrm{T}}\mathbf{W}=\mathbf{b}$$

If X is a singular matrix or not a square matrix then the inverse X^{-1} does not exist, therefore a pseudoinverse matrix is used:

$$\mathbf{X}^{\mathrm{T}\dagger}=\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}\mathbf{X},\quad \mathbf{X}^{\dagger}\mathbf{X}=\mathbf{I}$$

XX^T is square, has d dim.

$$\mathbf{W}=\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}\mathbf{X}\mathbf{b}=\mathbf{X}^{\mathrm{T}\dagger}\mathbf{b}$$

Multiplying by pseudoinverse of X^T will leave W on the left.

Singular Value Decomposition is another, very good method for solving such equation: see discussion and the algorithm in

See Numerical Recipes, chapter 2.6, on-line version is at:
http://www.nr.com

LDA perceptron algorithm

Many algorithms for creating linear decision surfaces have been inspired by perceptrons, very simplified models of neurons.

Criterion: $$J_p\left(\mathbf{W}\right)=-\sum_{i\in Er}\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)}>0$$

where summation goes over the set *Er* of misclassified samples, for which $W^TX<0$. This criterion $J_p(W)$ measures the sum of the misclassified distances from the decision boundary.
Minimization by gradient descent may be used:

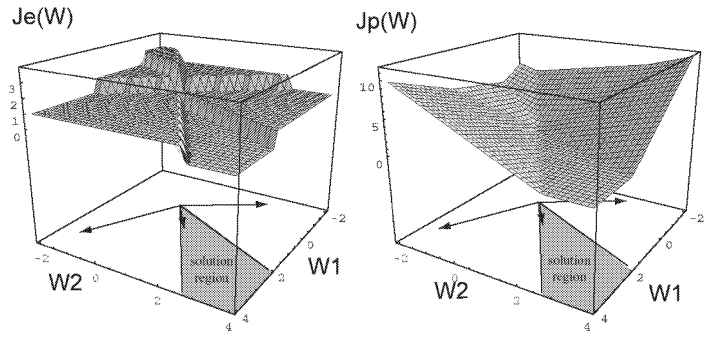
$$\mathbf{W}_{(k+1)}=\mathbf{W}_{(k)}-\eta_k\nabla J_p\left(\mathbf{W}\right)=\mathbf{W}_{(k)}+\eta_k\sum_{i\in Er}\mathbf{X}^{(i)}$$

where a learning constant η_k is introduced, dependent on the number of iterations *k*. This solves any linearly separable problem!

No large matrices, no problems with singularity, on-line learning OK.

Perceptron J_p(W)

Note that the perceptron criterion is piecewise linear.



Left side: number of errors;
right side: perceptron criterion; zero *J*(W) values in the solution region are possible only for linearly separable data.

(after Duda et all, fig. 5.11)

Back to Fisher DA

Fisher linear discrimination (FDA) was used to find canonical coordinates for visualization; they may also be used for classification.

Fisher projection: $\mathbf{Y}=\mathbf{W}^{\mathrm{T}}\mathbf{X}$, criterion $$\max_{\mathbf{W}} J\left(\mathbf{W}\right)=\frac{\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}}{\mathbf{W}^{\mathrm{T}}\mathbf{S}_I\mathbf{W}}$$

where the between-class scatter matrix is:

$$\mathbf{S}_B=\left(\bar{\mathbf{X}}_1-\bar{\mathbf{X}}_2\right)\left(\bar{\mathbf{X}}_1-\bar{\mathbf{X}}_2\right)^{\mathrm{T}}$$

and the within-class scatter matrix is:

$$\mathbf{S}_I=\sum_{k=1}\sum_{j=1}^{n(C_k)}\left(\mathbf{X}^{(j)}-\bar{\mathbf{X}}_k\right)\left(\mathbf{X}^{(j)}-\bar{\mathbf{X}}_k\right)^{\mathrm{T}}$$

How is this connected to LDA? $\mathbf{W}^{\mathrm{T}}\mathbf{X}$ defines a projection on a line, this line is perpendicular to the discrimination hyperplane going through **0** (since here $W_0=0$, $\mathbf{W}$ is *d*-dimensional here).

Relation to FDA

Linear discrimination with augmented d+1 dimensional vectors, and with -X taken for X from the second class, with $d \times n$ data matrix $\mathbf{X}$, is:

$$[W_0, \mathbf{W}]^{\mathrm{T}} \mathbf{X} = \mathbf{b}^{\mathrm{T}};$$

$$\mathbf{b}^{\mathrm{T}} = \left[\frac{n}{n_1}\mathbf{1}_1 \quad \frac{n}{n_2}\mathbf{1}_2 \right]$$

with special choice for b;
$\mathbf{1}_i$ – a row with n_i elements =1
for all n_1 samples from ω_1 class n/n_1
for all n_2 samples from ω_2 class n/n_2

With this choice we can show that W is the Fisher solution:

$$W_0 = -\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}; \quad \bar{\mathbf{X}} = \frac{n_1}{n}\bar{\mathbf{X}}_1 + \frac{n_2}{n}\bar{\mathbf{X}}_2$$

the mean of all X

$$\mathbf{W} = \alpha \mathbf{S}_I^{-1}\left(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right)$$

α is a scaling constant

$$\mathbf{W}^{\mathrm{T}}\left(\mathbf{X} - \bar{\mathbf{X}}\right) > 0 \text{ then Class} = \omega_1$$

decision rule; in practice W_0 threshold is estimated from data.

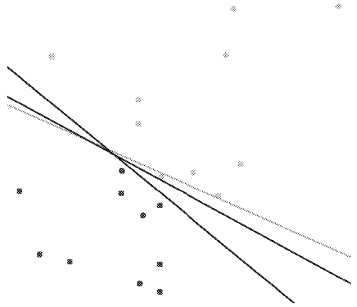
LD solutions

FDA provides one particular way of finding the linear discriminating plane; detailed proofs are in the Duda, Hart and Stork (2000) book.

Method	Criterion	Solution
LMS with pseudoinverse:	$J_L = \left\|\mathbf{W}^{\mathrm{T}}\mathbf{X} - \mathbf{b}\right\|^2$	$\mathbf{W} = \mathbf{X}^{+}\mathbf{b}$
Perceptron:	$J_p = -\sum_{i \in Er} \mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)}$	$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} + \eta_k \sum_{i \in Er} \mathbf{X}^{(i)}$
Fisher	$J_F = \frac{\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}}{\mathbf{W}^{\mathrm{T}}\mathbf{S}_I\mathbf{W}}$	$\mathbf{W} = \alpha \mathbf{S}_I^{-1}\left(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right)$
Relaxation	$J_R = \frac{1}{2}\sum_{i \in \Upsilon}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)} - b\right)^2 \Big/ \left\|\mathbf{X}^{(i)}\right\|^2$	
	$\Upsilon = \left\{i \mid \mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)} \le b\right\}$	Use only vectors closer than $b/\|\mathbf{W}\|$ to the border.

Differences

LMS solution – orange line;
two perceptron solutions, with different random starts – blue lines.



Optimal linear solution: should optimize both W to be as far as possible from the data on both sides; it should have large margin b.

Quadratic discrimination

Bayesian approach to optimal decisions for Gaussian distributions leads to the quadratic discriminant analysis (QDA)

$$g_i(\mathbf{X}) = -\frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i) - \frac{1}{2}\left(\mathbf{X} - \bar{\mathbf{X}}_i\right)^T \Sigma_i^{-1}\left(\mathbf{X} - \bar{\mathbf{X}}_i\right)$$

Hyperquadratic decision boundaries: $g_i(\mathrm{X}) = g_j(\mathrm{X})$;

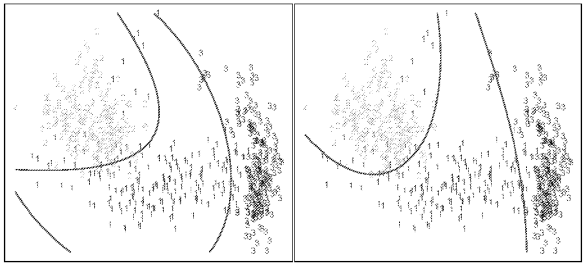
For equal a priori probabilities Mahalanobis distance to the class center is used as discriminant.

Decision regions do not have to be simply connected.

Estimate covariance matrices from data, or fit $d(d+2)/2$ parameters to the data directly to obtain QDA – both ways are rather inaccurate.

QDA example

QDA for some datasets works quite well.



Left: LDA solution with 5 features X_1, X_2, X_1^2, X_2^2 and X_1X_2;
Right – QDA solution with X_1, $X_{2;}$

Differences are rather small, in both cases 6 parameters are estimated, but in real calculations QDA was usually worse than LDA.

Why? Perhaps it is already too complex.

Regularized discrimination (RDA)

Sometimes linear solution is almost sufficient, while quadratic discriminant has too many degrees of freedom and may overfit the data preventing generalization.

RDA interpolates between the class covariance matrices Σ_k and the average covariance matrices for the whole data Σ:

$$\Sigma_k(\alpha) = \alpha\Sigma_k + (1-\alpha)\Sigma$$

$$\Sigma_k = \frac{1}{n_k}\sum_{i\in\omega_k}\left(\mathbf{X}^{(i)} - \bar{\mathbf{X}}_k\right)\left(\mathbf{X}^{(i)} - \bar{\mathbf{X}}_k\right)^{\mathrm{T}}$$

$$\Sigma = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in\omega_k}\left(\mathbf{X}^{(i)} - \bar{\mathbf{X}}_k\right)\left(\mathbf{X}^{(i)} - \bar{\mathbf{X}}_k\right)^{\mathrm{T}}$$

Best α is estimated using crossvalidation; classification rules are based on QDA, for $\alpha = 0$, QDA is reduced to LDA – simpler decision borders. RDA gives good results, but where is the software (except in S or R)?