Lecture 10 - Feb 12

Assessing Classifiers
Fisher's Linear Discriminant Analysis (LDA)

Reading

Week 6 Notes in GitHub

Data Mining and Machine Learning
1.3.3: Orthogonal Projection
20.1: Linear Discriminant Analysis

Elements of Statistical Learning
4.3 Discriminant Analysis

Upcoming Deadlines

Homework 2 (Feb 18)

Suppose $M: \mathbb{R}^d \to \{c_1, \ldots, c_k\}$ is a classifier

To build a classifier requires a training set.
We assess its performance on a testing set.

22.1 Classification Performance Measures

D - testing set of n points $x_i \in \mathbb{R}^d$ with labels y_i

$\hat{y}_i = M(x_i)$ - predicted label in $\{c_1, \ldots, c_k\}$

$$\mathbb{1}_A = \begin{cases} 0, & \text{if } A \text{ is false} \\ 1, & \text{if } A \text{ is true} \end{cases}$$

Error Rate: $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{y_i \neq \hat{y}_i\}}$ → fraction that is incorrectly classified

Accuracy: $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{y_i = \hat{y}_i\}}$ → fraction correctly classified

These metrics are global and do not consider how different classes contribute to the error.

Let $D_j = \{x_i \mid y_i = c_j\}$ points in class j

- $n_j = |D_j|$ = # of points in class j

$R_j = \{x_i \mid \hat{y}_i = c_j\}$ points with predicted class j

- $m_j = |R_j|$ = # of points with predicted class j

A confusion matrix is a $k \times k$ table N with

$N_{ij} = n_{ij}$ - number of points with predicted class c_i with true label c_j

row i, col j

$= |R_i \cap D_j| = \{x_a \in D \mid \hat{y}_a = c_i \text{ and } y_a = c_j\}$

	true class c_1	c_2	c_3
c_1	n_{11}	n_{12}	n_{13} → Errors

Precision: the class-specific precision of M for class c_i is

$$prec_i = \frac{n_{ii}}{m_i} = \frac{\text{true class } i \text{ predictions}}{\text{class } i \text{ predictions}} \quad (1 = \max)$$

i.e. the fraction of our class i predictions that are correct.

Global precision is

$$\text{precision} = \sum_{i=1}^{k} \left(\frac{m_i}{n}\right) prec_i = \sum_{i=1}^{k} \frac{n_{ii}}{n} \quad (= \text{accuracy})$$

Recall:

$$recall_i = \frac{n_{ii}}{n_i} \quad (1 = \max)$$

the fraction of class i points we correctly identify

There is a trade-off between precision and recall...
for example, if $M(x_i) = y_2$ for all x_i, then $recall_2$ is

$$\frac{n_2}{n_2} = 1$$

but precision is $\frac{n_2}{n}$, which is likely low.

The class-specific F-measure tries to balance them

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2 \cdot \frac{n_{ii}}{m_i} \cdot \frac{n_{ii}}{n_i}}{\frac{n_{ii}}{m_i} + \frac{n_{ii}}{n_i}}$$

$2n_{ii}^2$ $\quad$ $2n_{ii}$

$$= \frac{2n_{ii}^2}{n_{ii}n_i + n_{ii}m_i} = \frac{2n_{ii}}{n_i + m_i}$$

for F_i to be near 1, we need both to be near 1.

Global F-score $= \frac{1}{k}\sum_{i=1}^{k} F_i$ AKA F1-score

Binary Classification: Suppose there are only 2 classes, we can call

Predicted class \ True Class	pos.	neg.
pos	n_{11} = True Positives	n_{12} = False Positives
neg	n_{21} = False Negatives	n_{22} = True Negatives

$$\text{prec}_p = \frac{TP}{TP+FP}$$ % of pos. predictions that are correct

$$\text{recall}_p = \frac{TP}{TP+FN}$$ % of pos. points that are correct

Receiver Operating Characteristic (ROC)

2 classes

Popular strategy for assessing a binary classifier which outputs a score value for the positive class for each point in the test set, e.g. Bayes classifier outputs a posterior probability $\hat{P}(c_1|x_i)$.

Typically, $M(x_i) = \begin{cases} c_1, & \text{if score} > \theta \\ c_2, & \text{else} \end{cases}$ but θ is a somewhat arbitrary threshold

decision threshold

Plot the false positive rate vs. true positive rate

ROC plots false positive rate vs. true positive rate for each value of θ from θ^{min} to θ^{max}

FP / neg. points

TP / pos. points

$\min_i S(x_i)$

$\max_i S(x_i)$

S - score for pos. class

At θ^{min} ...

pred. \ truth	P	N
P	0	0
N	FN	TN

At θ^{max} ...

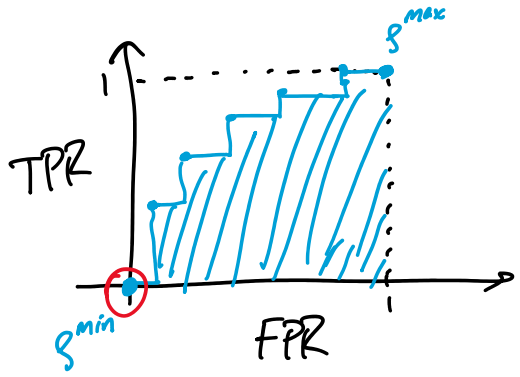
pred. \ truth	P	N
P	TP	FP
N	0	0

Ideal Classifier

	P	N
P	TP	0
N	0	TN

between...

$R_i(\theta) = \{x_i \in D \mid S(x_i) > \theta\}$ - points classified pos. for threshold θ



Ideally, the curve should grow to 1 as quickly as possible

← perfect ROC

Area under ROC curve (AUC) is a one-number performance metric in $[0,1]$ - the prob. classifier will rank a random pos. test point higher than a random neg. point.

imp: ROC/AUC is not sensitive to class imbalance

The angle between vectors a and b is

$$\cos\theta = \frac{a^T b}{\|a\|\,\|b\|} = \left(\frac{a}{\|a\|}\right)^T \left(\frac{b}{\|b\|}\right)$$

unit vectors

$$\Rightarrow \quad \theta = 0 \iff \frac{a^T}{\|a\|}\frac{b}{\|b\|} = 0 \iff a^T b = 0$$

Any vector b can be written

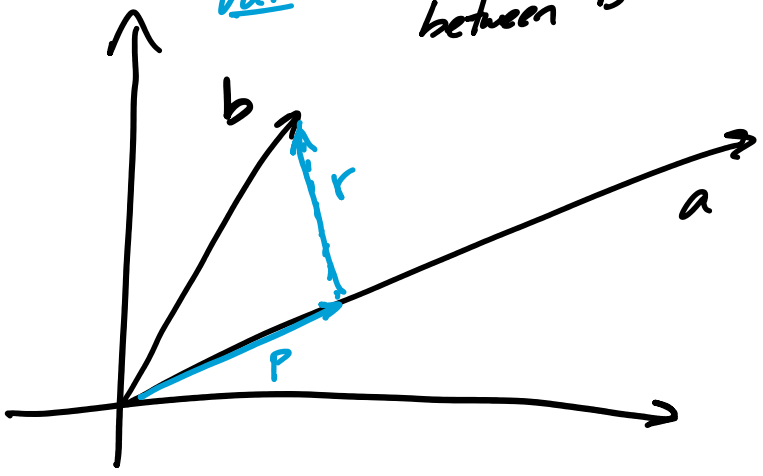
$$b = b_{\parallel} + b_{\perp} = p + r$$

where $b_{\parallel} = p$ is parallel to a vector a and $b_{\perp} = r$ is orthogonal to a.

p: orthogonal projection of b onto a

r: error vector

$\|r\| = \|b - p\|$ is the perpendicular distance between b and a



note $p = ca$ for some constant c.

note $p = ca$ for some constant c.

Since p and r are orthogonal,

$$0 = (ca)^T r = ca^T(b - ca) = ca^Tb - c^2a^Ta$$

$$ca^Ta = a^Tb$$

$$c = \frac{a^Tb}{a^Ta}$$

scalar projection of b on a $= \text{proj}_a(b)$ = "offset" along a

$$\Rightarrow \quad p = ca = \left(\frac{a^Tb}{a^Ta}\right)a$$

Let $A \in \mathbb{R}^{n \times n}$, $u \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$

If $Au = \lambda u$, u is an eigenvector of A
λ is an eigenvalue of A

a vector that points in the same direction when multiplied by A

How do we find eigenvectors?

$$Au = \lambda u \Rightarrow Au - \lambda u = 0$$
$$(A - \lambda I)u = 0$$

By the invertible matrix theorem, this system has a nonzero solution if and only if

computed at $O(n^{2.3\ldots})$

$$\det(A - \lambda I) = 0$$

fundamental theorem of algebra

$$(\lambda - \lambda_1) \cdots (\lambda - \lambda_n) = 0$$

where $\lambda_1, \ldots, \lambda_n$ are solutions (i.e. the eigenvalues)

$(x_i, y_i) \in \mathbb{R}^d \times \{c_1, \ldots, c_k\}$ dataset

LDA seeks a vector w that maximizes the separation between classes after projection onto w "discriminates"

20.1 Optimal Linear Discriminant

D - dataset of n points $x_i \in \mathbb{R}^d$ & labels $y_i \in \{c_1, \ldots, c_k\}$

D_i - class c_i points $\{x_j \in D \mid y_j = c_i\}$

$|D_i| = n_i$

Assume $k = 2$, so $D = D_1 + D_2$

Let w be a unit vector. i.e. $\|w\|^2 = w^T w = 1$

The projection of x_i onto w is

$$x_i' = \left(\frac{w^T x_i}{w^T w}\right) w = \underbrace{(w^T x_i)}_{a_i} w = a_i w$$

a_i — offset or scalar projection of x_i on the line w
AKA projected point

$\Rightarrow$ projected points $\{a_1, \ldots, a_n\}$ map $\mathbb{R}^d$ to $\mathbb{R}$ from d-dim points to 1-dim offsets

Projected means:

$$m_1 = \frac{1}{n_1} \sum_{x_i \in D_1} a_i = \frac{1}{n_1} \sum w^T x_i$$

$$= w^T \left(\frac{1}{n_1} \sum x_i \right) = w^T \mu_1$$

$$m_2 = w^T \mu_2$$

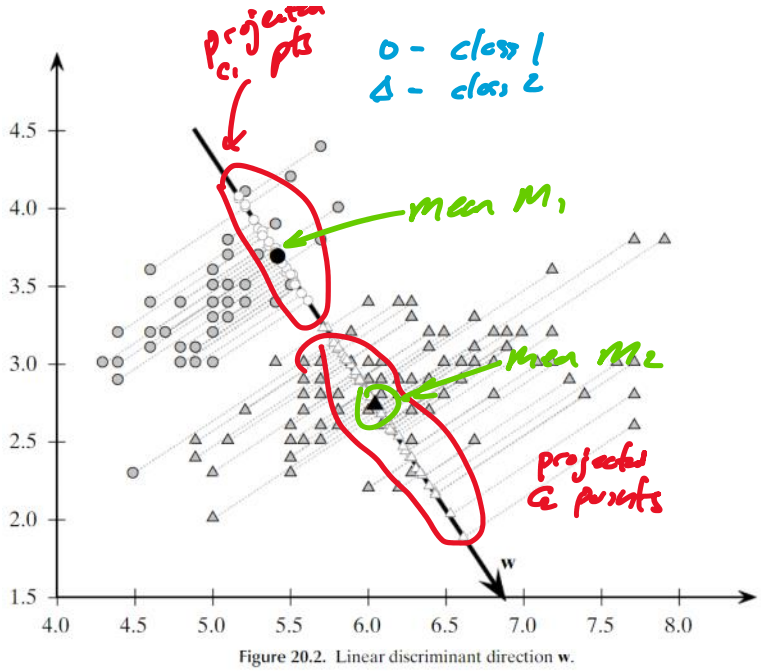


Figure 20.2. Linear discriminant direction w.

To maximize separation, maximizing $|m_1 - m_2|$ sounds reasonable, but we also need variance to not be too large so that the classes do not overlap too much

$\Rightarrow$ LDA ensures the scatter s_i in each class is small

$$s_i = \sum_{x_j \in D_i} (a_j - m_i)^2 = n_i \sigma_i^2$$

↑ variance for class c_i

Fisher LDA objective: $\max_w J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$

Solving this finds a direction w s.t. projected means are far apart without allowing large scatter

Let's solve it... first, express it in terms of input data:

$$(\quad)^2 - (\quad)^2 = (w^T(\mu_1 - \mu_2))(w^T(\mu_1 - \mu_2))^T$$

Let's ~~m~~

$$(m_1 - m_2)^2 = \left(w^T(\mu_1 - \mu_2)\right)^2 = w^T(\mu_1 - \mu_2)\left[w^T(\mu_1 - \mu_2)\right]^T$$

$$= w^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w$$

$$= w^T B w$$

B ← between-class scatter matrix

$$s_i^2 = \sum_{x_j \in D_i} (c_j - m_i)^2 = \sum_{x_j \in D_i} \left(w^T x_j - w^T \mu_i\right)^2$$

$$= \sum_{x_j \in D_i} \left(w^T(x_j - \mu_i)\right)^2 =$$

$$= w^T \left(\sum_{x_j \in D_i} (x_j - \mu_i)(x_j - \mu_i)^T \right) w$$

$\sum_{x_j \in D_i} (x_j - \mu_i)(x_j - \mu_i)^T$ ← scatter matrix for class $c_i = n_i \Sigma_i$

$$= w^T S_i w$$

Thus, $$J(w) = \frac{w^T B w}{w^T S_1 w + w^T S_2 w} = \frac{w^T B w}{w^T S w}$$ $S = S_1 + S_2$

Note: $S_1, S_2, S \in \mathbb{R}^{d \times d}$ are symmetric, positive semidefinite

$$\max_w J(w) = \max_w \frac{w^T B w}{w^T S w}$$

Let's seek critical values...

$$2Bw(w^T S w) - 2Sw(w^T B w)$$

$$\nabla J(w) = \frac{2Bw(w^T Sw) - 2Sw(w^T Bw)}{(w^T Sw)^2} = 0$$

$$Bw(w^T Sw) = Sw(w^T Bw)$$

$$Bw = Sw\left(\frac{w^T Bw}{w^T Sw}\right)$$

$$Bw = J(w)Sw \quad \text{(constant)}$$

$$Bw = \lambda Sw$$

generalized eigenvalue problem

If S is nonsingular (S^{-1} exists), then

$$S^{-1}Bw = \lambda S^{-1}Sw = \lambda w$$

$$(S^{-1}B)w = \lambda w \Rightarrow \lambda \text{ is an eigenvalue}$$

Solving $\det(S^{-1}B - \lambda I) = 0$ gives several solutions, so we choose the largest since $\lambda = J(w)$ is to be maximized

(use linear algebra to compute eigenvalues)

Fisher's

Algorithm 20.1: Linear Discriminant Analysis

LINEARDISCRIMINANT (D):

Algorithm 20.1: Linear Discriminant Analysis

LINEARDISCRIMINANT (D):

1 $\mathbf{D}_i \leftarrow \{\mathbf{x}_j^T \mid y_j = c_i, j = 1, \ldots, n\}, i = 1, 2$ // class-specific subsets

2 $\boldsymbol{\mu}_i \leftarrow \text{mean}(\mathbf{D}_i), i = 1, 2$ // class means

3 $\mathbf{B} \leftarrow (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ // between-class scatter matrix

4 $\overline{\mathbf{D}}_i \leftarrow \mathbf{D}_i - \mathbf{1}_{n_i} \boldsymbol{\mu}_i^T, i = 1, 2$ // center class matrices

5 $\mathbf{S}_i \leftarrow \overline{\mathbf{D}}_i^T \overline{\mathbf{D}}_i, i = 1, 2$ // class scatter matrices

6 $\mathbf{S} \leftarrow \mathbf{S}_1 + \mathbf{S}_2$ // within-class scatter matrix

7 $\lambda_1, \mathbf{w} \leftarrow \text{eigen}(\mathbf{S}^{-1}\mathbf{B})$ // compute dominant eigenvector

$(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

$\begin{bmatrix} \mu_{11} - \mu_{21} \\ \vdots \\ \mu_{1d} - \mu_{2d} \end{bmatrix}_{d \times 1} \begin{bmatrix} \mu_{11} - \mu_{21} & \cdots & \mu_{1d} - \mu_{2d} \end{bmatrix}_{1 \times d}$

In the 2-class case...

$$Bw = \lambda Sw$$

$$c(\mu_1 - \mu_2) = \lambda Sw$$

$c(\mu_1 - \mu_2)$

$$\frac{c}{\lambda} S^{-1}(\mu_1 - \mu_2) = w \Rightarrow \text{optimal solution } w = S^{-1}(\mu_1 - \mu_2)$$

constant