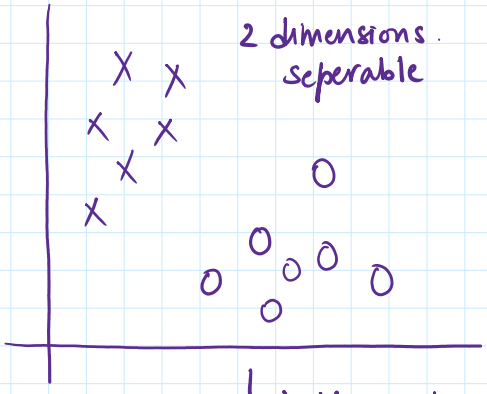
Linear Discriminant Analysis

14 October 2024 04:07



2 dimensions. seperable

$\xrightarrow{\text{PCA}}$

1 dimension not seperable.

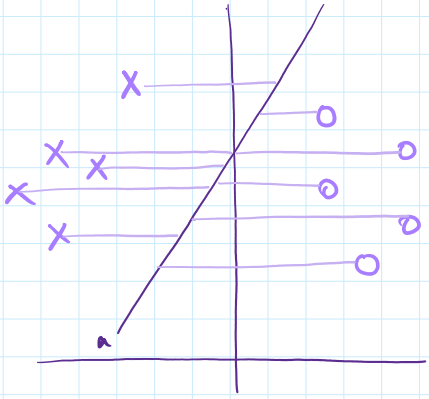
↳ Y axis has maximum variance
+
PCA projects in the direction of maximum variance.

So direction of max variance not very useful for classification.

LDA for the same set of samples.

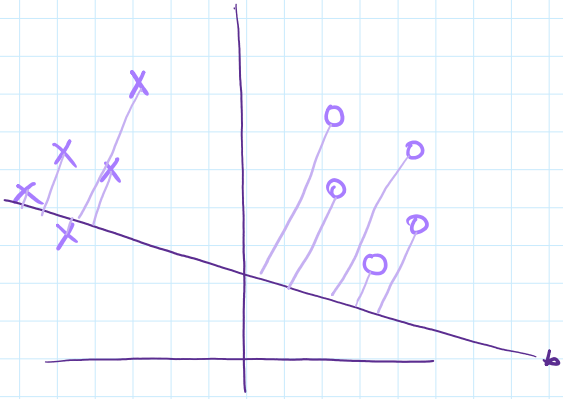
- we're using class labels

(a)



a is a bad line to project to

(b)



b is a good line to project to.

w/ b, means of both samples sufficiently seperated.

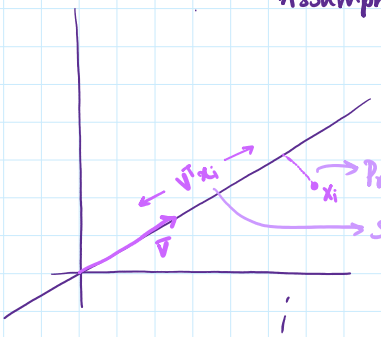
Deriving LDA.

Assumptions:

2 classes c_1, c_2

d dimensional samples.

n samples $x_1, x_2 \dots x_n$ — n_1 samples from c_1, n_2 samples from c_2



→ Project from x_i onto a line given by dir^n of unit vector v.

→ Scalar $v^T x_i$ is the projection length along v. → $v^T x_i$

Understood?

Let $\bar{\mu}_1 + \bar{\mu}_2$ be the mean of projections of classes 1, 2.

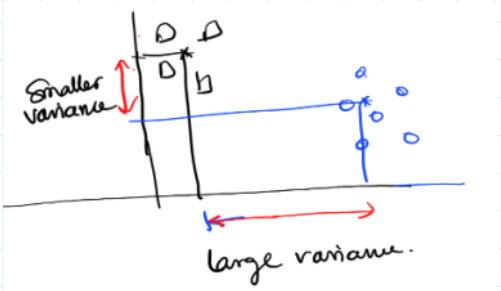
↓

$(\bar{\mu}_1 - \bar{\mu}_2)$ = measure of separation b/w classes.

$$\bar{\mu}_1 = \frac{1}{n_1} \sum_{x_i \in c_1}^{n_1} v^T x_i = v^T \left(\frac{1}{n_1} \sum_{x_i \in c_1}^{n_1} x_i \right) = v^T \mu_1$$

$$\bar{\mu}_2 = \frac{1}{n_2} \sum_{x_i \in c_2}^{n_2} v^T x_i = v^T \left(\frac{1}{n_2} \sum_{x_i \in c_2}^{n_2} x_i \right) = v^T \mu_2$$

$$\mu_2 = \frac{1}{n_2}\sum_{x_i \in C_2} v^T x_i = v^T\left(\frac{1}{n_2}\sum_{x_i \in C_2} x_i\right) = v^T \mu_2$$



But this doesn't look at the variance of the classes.

All we have to do is normalize $|\mu_2 - \mu_1|$ — but how?

Variance is our problem
⇒ normalize by a factor proportional to variance.

Keep that in mind.

Scatter:

Say you've n samples: $z_1, z_2 \ldots z_n$

Sample mean $\mu_z = \frac{1}{n}\sum_{i=1}^{n} z_i$

Small scatter

Large scatter.

Scatter ≡ variance BUT it's on a different scale.

Scatter $= \sum (z_i - \mu_z)^2$

or: (sample variance) × n.

Connect the dots swalpa

Normalizing $|\bar{\mu}_2 - \bar{\mu}_1|$ by scatter = FISHER LDA.

linear discriminant analysis.

Let projected i^{th} sample, $y_i = v^T x_i$

Scatter for class 1
$\tilde{S}_1^2 = \sum (y_i - \bar{\mu}_1)^2$

Scatter for class 2
$\tilde{S}_2 = \sum (y_i - \bar{\mu}_2)^2$

Fisher LDA: "project on the line in the dir^n of $\bar{v}$ that MAXIMIZES $J(\bar{v})$"

$$J(v) = \frac{(\bar{\mu}_1 - \bar{\mu}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2}$$

If $\tilde{S}_i^2$ is small, that means samples are crowded around $\bar{\mu}_i$.

Now express J <u>explicitly</u> as a function of v.
Then maximize!

① Define separate class matrices, S_1 + S_2 for C_1 + C_2 — scatter of original samples x_i BEFORE projection.

$$S_1 = \sum_{x_i \in C_1} (x_i - \mu_1)(x_i - \mu_1)^T$$

$$S_2 = \sum_{x_i \in C_2} (x_i - \mu_2)(x_i - \mu_2)^T$$

⇒ TOTAL scatter WITHIN classes $S_W = S_1 + S_2$

② Remember $\tilde{S}_1$ + $\tilde{S}_2$? Scatter for projected sample!

② Remember $\tilde{S}_1$ + $\tilde{S}_2$? Scatter for projected sample!

$$\tilde{S}_1^2 = \sum (y_i - \bar{\mu}_1)^2$$
$$\tilde{S}_2^2 = \sum (y_i - \bar{\mu}_2)^2$$

look @ projection y_i for a sample x_i

$y_i = V^T x_i$

$\mu_1 = V^T \mu_1$

$\mu_2 = V^T \mu_2$

③ Plug in these values of y_i + μ_i

$$\tilde{S}_1^2 = \sum_{y_i \in c_1} (V^T x_i - V^T \mu_1)^2$$

$$= \sum [V^T(x_i - \mu_1)]^T [V^T(x_i - \mu_1)]$$

transpose rule. rearrange

$$= \sum [(x_i - \mu_1)^T V]^T [(x_i - \mu_1)^T V]$$

$$= \sum V^T (x_i - \mu_1)(x_i - \mu_1)^T V$$

$$\tilde{S}_1^2 = V^T S_1 V$$

$$\Rightarrow \tilde{S}_2^2 = V^T S_2 V$$

④ Sum it all up now:

$$\tilde{S}_1^2 + \tilde{S}_2^2 = V^T S_1 V + V^T S_2 V$$
$$= V^T (S_1 + S_2) V$$
$$= V^T S_W V \text{ !!!}$$

Quick review:

$S_1 + S_2 = S_W$

$\tilde{S}_1^2 + \tilde{S}_2^2 = V^T S_W V$

So we got scatter within, S_W (basically, scatter of data points within the class)

↓

Between class scatter then becomes the difference b/w $\bar{\mu}_i$ (means of classes)

Between class separation matrix $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ → Before projection.

↓

What about after projection?

$(\tilde{\mu}_1 - \tilde{\mu}_2)^2$

But we can write that better.

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (V^T \mu_1 - V^T \mu_2)^2$$

cuz $\tilde{\mu}_i = V^T \mu_i$

$$= V^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T V$$
$$= V^T S_B V$$

Quick recap:

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

$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = V^T S_B V$

Okay cool.
But why did we do all this?

From before: $J(V) = \dfrac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2} = \dfrac{V^T S_B V}{V^T S_W V}$

From before: $J(v) = \dfrac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \dfrac{v^T S_B v}{v^T S_W v}$

INTUITION TIME

$J(v) = \dfrac{v^T S_B v}{v^T S_W v}$ → So $J(v)$ ends up being a sort of quantitative measure for understanding variations b/w different classes as well as within.

big interclass variation

If $J(v)\uparrow$ it tells us that between class variation has increased relative to within class variation.
(ie change in $S_B >>> S_W$)

big within class variation

So if $J(v)\downarrow$ it tells us that within class variation has increased relative to b/w class variation
(ie change in $S_W >>> S_B$)

Now for parameter estimation:

what we have: $J(v)$

what we need to find: some value of v to optimise $J(v)$

what really matters: NOT the magnitude of v. — remember $y_i = v^T x_i$? y is a f^n of x. → $f(x_i) = v^T x_i$

the general DIRECTION of v.

cuz

we want to rank $f(x_i) = v^T x_i$ values.
That's all.
Using a vector proportional to v won't change ranking.

Now before we go further, look @ this:

For a symmetric matrix $\mathbf{W}$ and a vector $\mathbf{s}$, we have

$$\frac{\partial \mathbf{s}^\top \mathbf{W}\mathbf{s}}{\partial \mathbf{s}} = 2\mathbf{W}\mathbf{s}.$$

Notice that Σ_w is symmetric since its $(i,j)^{\text{th}}$ element is

$$\sum_{n\in S_1}(x_{ni}-\mu_{1i})(x_{nj}-\mu_{1j}) + \sum_{n\in S_2}(x_{ni}-\mu_{2i})(x_{nj}-\mu_{2j}),$$

which is equivalent to its $(j,i)^{\text{th}}$ element.

Done!

Now, $$\frac{d}{dv} J(v) = \frac{\left(\frac{d}{dv} v^T S_B v\right) v^T S_W v - \left(\frac{d}{dv} v^T S_W v\right) v^T S_B v}{(v^T S_W v)^2}$$

Now, we're trying to minimize $J(v)$ so we'll set the derivative to 0.
Denom is a scalar so that goes.

$$0 = (2 S_B v) v^T S_W v - (2 S_W v) v^T S_B v$$

$$0 = S_B v v^T S_W v - S_W v v^T S_B v$$

$$0 = S_B v (v^T S_W v) - S_W v (v^T S_B v)$$

divide by $v^T S_W v$ both sides.

$$0 = S_B v - \underbrace{\frac{(v^T S_B v)}{(v^T S_W v)}}_{\lambda} (S_W v)$$

$\lambda = \dfrac{v^T S_B v}{v^T S_W v}$

$$S_B v = \lambda S_W v$$

$\rightarrow \lambda$

$S_B V = \lambda S_W V$

If S_W has a full rank (and $\therefore$ is invertible)

$$(S_W^{-1} S_B) V = \lambda V$$

Just like we'd done w/ PCA, we obtain the LDA by taking