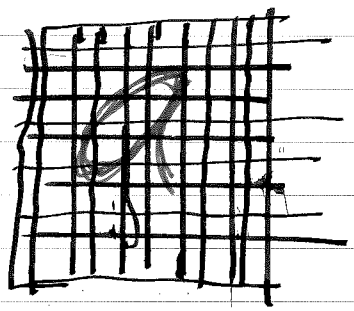
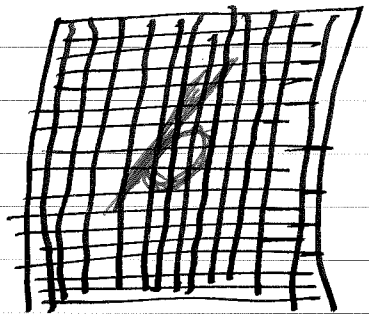
Some Pattern Recognition & Overview

PCA was an example of unsupervised learning algorithm in that it segregates data without feedback on training examples.

Consider the Character Recognition problem





Characters 'a' & 'b' are images represented in an array of pixels the i^{th} pixel having value x_i, say, where $x_i = 0$ means white & $x_i = 1$ means black. Together the image is a vector $\vec{x} = [x_1 \cdots x_d]^T$

Want a function $f: \mathbb{R}^d \to \{1, 2\}$ to be a character recognizer. Really f is "classifying" by mapping $\mathbb{R}^d$ into C_1 or C_2 ← class 1 & class 2.

We imagine we have a training set of classified images $\{(\vec{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$ An image might be a 256×256 object so $d = 65,536$. If colors are stored as 8-bit ints there's $2^{8 \cdot 65,536} \approx 10^{158,000}$ possible images! That's huge as there are only $\sim 10^{80}$ particles available in the universe.

MIQUELRIUS

On the other hand our training set might only have a few thousand images.

The classifier, f_1, is then desired to have the ability to generalize, namely to correctly classify unseen-before examples.

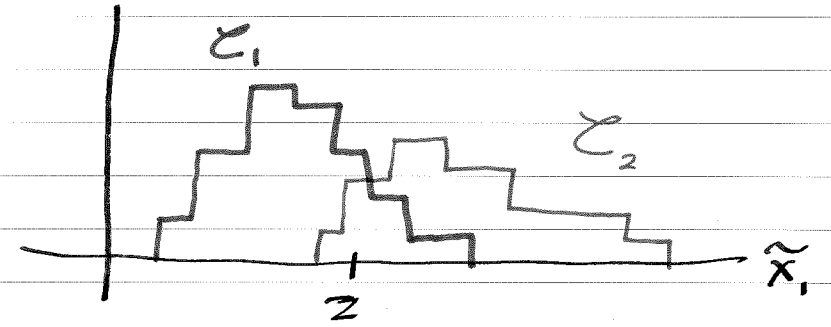
Note that the curse of dimensionality is acting against us here.

One way to try to fight dimensionality is to create new variables, called features which combine input variables and reduce the size of dimension. Design of features is important.

For example, maybe $\tilde{x}_1 = \frac{\text{height of character}}{\text{width of character}}$

$\tilde{x}_1$ then combines pairs of input variables (pixel values) into a single feature variable which, mainly, is more informative.

Consider the Histogram



How should a new character with $\tilde{x}_1 = 2$ be classified?

Since there is significant overlap a great # of misclassifications will occur if we use a simple threshold, namely,

let $x^* = \{\tilde{x}_1 \mid C_1 \text{ curve} = C_2 \text{ curve}\}$

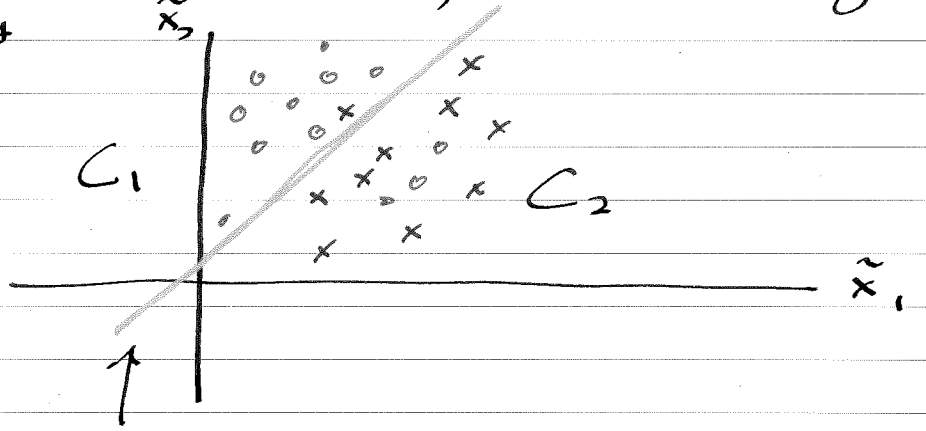
Then $f_{\text{thresh}}(\vec{x}) = \begin{cases} C_1 \text{ if } \tilde{x}_1(\vec{x}) < x^* \\ C_2 \text{ if } \tilde{x}_1(\vec{x}) \geq x^* \end{cases}$

MIQUELRIUS

But the sign of overlap indicates there will likely be many misclassifications. since the map

$\tilde{x}_1 : \mathbb{R}^d \longrightarrow \mathbb{R}$ represents severe loss of information.

Perhaps there is a second feature $\tilde{x}_2$ (whose details don't matter!)
and we can then represent the training set in $\mathbb{R}^2$
as



called a "decision boundary" and separates $(\tilde{x}_1, \tilde{x}_2)$
space into the 2 class regions.

Now $f_{dec}(\vec{x}) = \begin{cases} C_1 & \text{if } \vec{x} \text{ is above boundary} \\ C_2 & \text{else} \end{cases}$

In general we're mapping

$\vec{x} \in \mathbb{R}^d \ni \vec{x} \longrightarrow y_k \in \{1, 2, 3, \ldots, N\}$

N = # of classifiers as if classifying all letters
would have $N = 26$.

MIQUELRIUS

In general $f(\vec{x}) = y_k(\vec{x}; \vec{\theta})$

$\vec{\theta}$ = vector of model parameters or often called "weights".

Determining $\vec{\theta}$ based on data set is called "learning" or "training".

Then Classification problems are when Range(f) is a discrete set.
When only output of f is a continuous variable this is called regression.

Both regression & classification are a form of function approximation.

Now also we could try to divide each input variables axis into small intervals, which splits $\mathbb{R}^d$ into a grid of (hyper)-boxes. Each $\vec{x}^{(i)}$ in training set falls in a box and it has the label $y^{(i)}$.
Then, given a new example $\vec{x}_{new}$, we return
$y =$ {average of all training examples in that box}
But if each axis has M divisions then there are M^d boxes but

$$M^d >> \#\text{training examples}$$

So the data is very sparsely distributed.

So the naive approach won't work. That's one reason why dimension reduction is a valuable tool.

Fewer variables also reduces likelihood of overfitting.

MIQUELRIUS

In general, as model complexity grows so too does generalization error (caused by overfit).

Anyways, $P(\mathcal{C}_k)$ = prior probability of character in $\mathcal{C}_k$

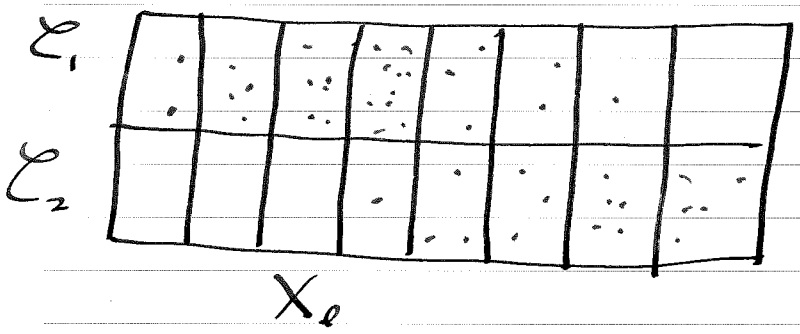
we can get these priors from the data themselves. For instance. if 'a' occurs 3 times more than 'b' then

$$P(\mathcal{C}_1) = 3/4 \qquad P(\mathcal{C}_2) = 1/4.$$

If we had to classify a new character without being able to see it we should assign to the class with the highest probability.

Now suppose $\tilde{x}_1 \in \{X_\ell\}$ ← discrete set of values, as used in histogram.

$P(\mathcal{C}_k, X_\ell)$ = probability the image has feature value X_ℓ & belongs to class $\mathcal{C}_k$



$\Rightarrow P(\mathcal{C}_k, X_\ell)$ = fraction of images in $(\mathcal{C}_k, X_\ell)$ cell

$P(\mathcal{C}_k)$ = fraction in each row
$P(X_\ell)$ = fraction in each column.

$P(X_\ell | \mathcal{C}_k)$ = probability of being in ℓ^{th} column given that you're in k^{th} row

MIQUELRIUS

That's the function in more & fully in cell X_e.

$$\mathbb{P}(\mathcal{C}_k \mid X_e) = \frac{\mathbb{P}(X_e \mid \mathcal{C}_k)\,\mathbb{P}(\mathcal{C}_k)}{\mathbb{P}(X_e)}$$

by Bayes. as before since

$$\mathbb{P}(\mathcal{C}_1 \mid X_e) + \mathbb{P}(\mathcal{C}_2 \mid X_e) = 1$$

Then $$\frac{\mathbb{P}(X_e \mid \mathcal{C}_1)\,\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(X_e)} + \frac{\mathbb{P}(X_e \mid \mathcal{C}_2)\,\mathbb{P}(\mathcal{C}_2)}{\mathbb{P}(X_e)} = 1$$

ie. $$\mathbb{P}(X_e) = \mathbb{P}(X_e \mid \mathcal{C}_1)\,\mathbb{P}(\mathcal{C}_1) + \mathbb{P}(X_e \mid \mathcal{C}_2)\cdot\mathbb{P}(\mathcal{C}_2)$$

ie.

$$\boxed{\mathbb{P}(\mathcal{C}_k \mid X_e) = \frac{\mathbb{P}(X_e \mid \mathcal{C}_k)\,\mathbb{P}(\mathcal{C}_k)}{\mathbb{P}(X_e \mid \mathcal{C}_1)\,\mathbb{P}(\mathcal{C}_1) + \mathbb{P}(X_e \mid \mathcal{C}_2)\cdot\mathbb{P}(\mathcal{C}_2)}}$$

Notice everything on RHS is obtainable from the histogram of training data.

We should then say

$$\boxed{f(\vec{x}) = \mathcal{C}_k \text{ where } \mathbb{P}(\mathcal{C}_k \mid \vec{x}) > \mathbb{P}(\mathcal{C}_j \mid \vec{x}) \quad \forall\, j \neq k}$$

MIQUELRIUS

Each point of feature space then is assigned to one of N regions viz

$$P(x|\mathcal{C}_k) > P(x|\mathcal{C}_j) \qquad j \neq k$$

i.e.

$$\boxed{f(x) = \underset{j \in \{1,\dots,N\}}{\operatorname{argmax}} P(x|\mathcal{C}_j)}$$

This splits $\mathbb{R}^d$ into c regions $R_1, \dots, R_c$

i.e. $\mathbb{R}^d = \bigcup_{i=1}^{c} R_i$ Boundaries of which ∂R_i's are decision boundaries or decision surfaces.

Now, consider again $N = 2$

$$P(\text{error}) = P(\vec{x} \in R_2, \mathcal{C}_1) + P(x \in R_1, \mathcal{C}_2)$$

$$= P(x \in R_2|\mathcal{C}_1)P(\mathcal{C}_1) + P(x \in R_1|\mathcal{C}_2)P(\mathcal{C}_2)$$

$$= \int_{R_1} p(x|\mathcal{C}_1)P(\mathcal{C}_1)\,dx + \int_{R_2} p(x|\mathcal{C}_2)P(\mathcal{C}_2)\,dx$$

if $p(x|\mathcal{C}_1)P(\mathcal{C}_1) > p(x|\mathcal{C}_2)P(\mathcal{C}_2)$ then we'd classify it as C_1 meaning R_1 should be R_2 should be picked so that $\vec{x} \in R_1$ since that integrand is smaller & contributes less error.

MIQUELRIUS

$y_k(\vec{x}) \stackrel{def}{=} P(\mathcal{C}_k | \vec{x})$ is called a discriminant function.

And our decisions are based on

$$f(\vec{x}) = \underset{j}{\text{argmax}} \{ y_j(\vec{x}) \}$$

Of course regions R_i, R_j are contiguous (i.e. $\partial R_i \cap \partial R_j \neq \emptyset$) then

$$\partial R_i \cap \partial R_j = \{ \vec{x} \in \mathbb{R}^d \mid y_i(\vec{x}) = y_j(\vec{x}) \}$$

& can characterize the 2-class classification problem in terms of

$$y(x) = y_1(\vec{x}) - y_2(\vec{x})$$

$$R_1 = \{ \vec{x} \mid y > 0 \} \qquad R_2 = \{ x \mid y < 0 \}$$

Discriminants can be found often without having to first do the probability estimations.

Generalization of this is with a loss matrix

$$L_{kj} = \{ \text{penalty for assigning to } \mathcal{C}_j \text{ when should be } \mathcal{C}_k \}$$

MIQUELRIUS

Then expected loss is

$$R_k = \sum_{j=1}^{N} L_{kj} \int_{R_j} p(x|\mathcal{C}_k)\,dx$$

and the <u>risk</u> is $R = \sum_{k=1}^{N} R_k P(\mathcal{C}_k)$

$$= \sum_{j=1}^{N} \int_{R_j} \sum_{k=1}^{N} L_{kj}\, p(x|\mathcal{C}_k) P(\mathcal{C}_k)\,dx$$

Risk is minimized when each integrand is minimized i.e. when

$$f(\vec{x}) = \underset{j}{\operatorname{argmin}} \sum_{k=1}^{N} L_{kj}\, p(x|C_k) P(\mathcal{C}_k)$$

The prior decision rule used $L_{kj} = 1 - \delta_{kj}$ i.e. indifferent to the kind of error. But sometimes the <u>type</u> of misclassification matters more than itself (e.g. medicine).

MIQUELRIUS