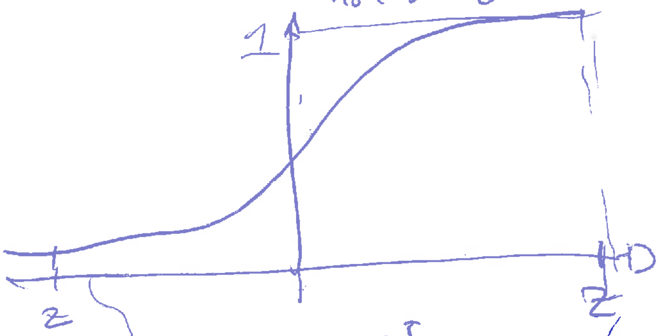
Week 7

Support Vector Machines ~ Optimization objective

Alternative view of logistic regression

$h_\theta(x) = g(z)$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \;?$$



$z = \theta^T x$

If $y=1$, we want $h_\theta(x) \simeq 1$, $\theta^T x \gg 0$

If $y=0$, we want $h_\theta(x) \simeq 0$, $\theta^T x \ll 0$

(x, y)

Alternative view of logistic regression

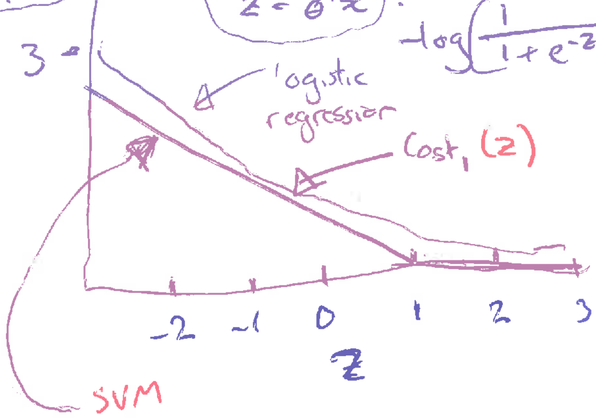
Cost of example: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

$$= -y \cdot \log \frac{1}{1+e^{-\theta^T x}} - (1-y)\log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$$

If $y=1$ (want $\theta^T x \gg 0$)

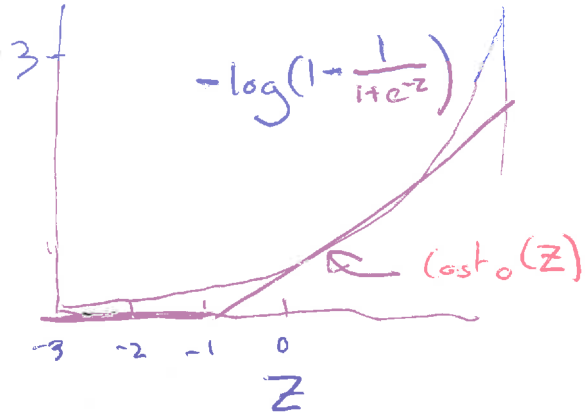
$z = \theta^T x$

$-\log\left(\frac{1}{1+e^{-z}}\right)$



If $y=0$ (want $\theta^T x \ll 0$)

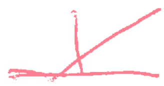
$-\log\left(1 - \frac{1}{1+e^{-z}}\right)$



Support Vector Machine

replace with:

$\text{cost}_1(\theta^T x^{(i)})$ $\text{cost}_0(\theta^T x^{(i)})$



logistic regression

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1-y^{(i)})\left(\left(-\log(1-h_\theta(x^{(i)})\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Support vector machine:

remove constant

$$\min_\theta \cancel{\frac{1}{m}} \sum_{i=1}^{m} y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\,\text{cost}_0(\theta^T x^{(i)}) + \frac{\lambda}{2\cancel{m}}\sum_{j=0}^{n}\theta_j^2$$

remove constant

In LR we used $A + \lambda B$ to control the balance of the parameters θ

In SVMs we use $CA + B$, C is similar to $\frac{1}{\lambda}$

So overall SVMs are defined by:

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1 (\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0 (\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

The SVM does not output a probability...

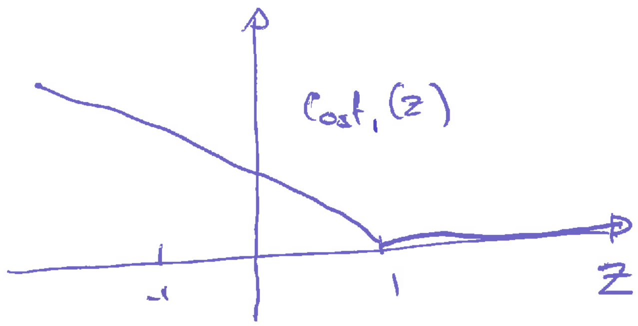
hypothesis:

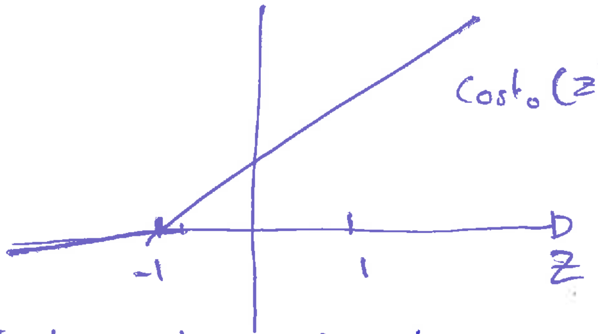
$$h_\theta(x) \begin{cases} 1 - \text{if } \theta^T x \geq 0 \\ 0 - \text{otherwise} \end{cases}$$

Week 7

Support Vector Machines: Large margin intuition

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$





if $y=1$, we want $\theta^T x \geq 1$ (not just ≥ 0) | $\theta^T x \geq 0$

if $y=0$, we want $\theta^T x \leq -1$ (not just ≤ 0) | $\theta^T x < 0$

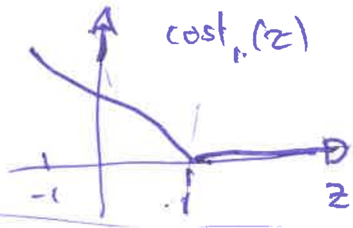
$C = 100,000$

SVM decision boundary

$$\min_{\theta} C \underbrace{\sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right]}_{=0} + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

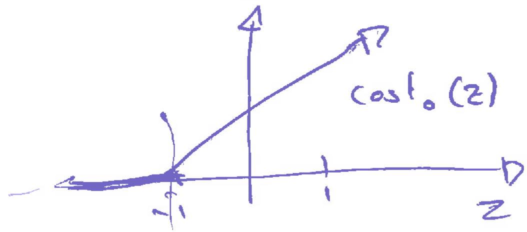
whenever $y^{(i)} = 1$:

$\theta^T x^{(i)} \geq 1$



whenever $y^{(i)} = 0$:

$\theta^T x^{(i)} \leq -1$



$$\min \; C \times 0 + \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

s.t. $\theta^T x^{(i)} \geq 1$ if $y^{(i)} = 1$

$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$

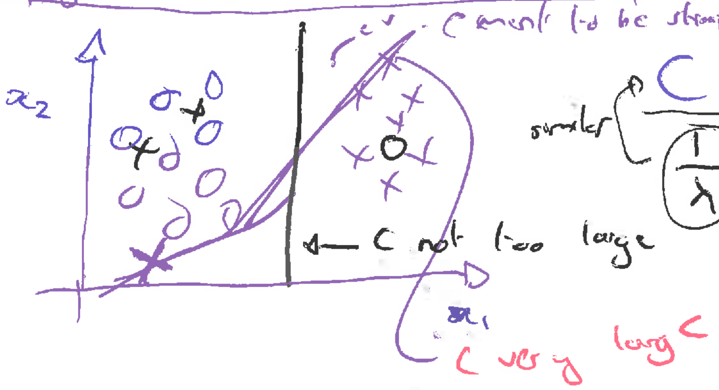
Week 7

Support vector machines: large margin intuition

The margin gives the SVM some robustness because it tries to separate the margin with as large as possible

"Large Margin classifier"

Large margin classifier in presence of outliers



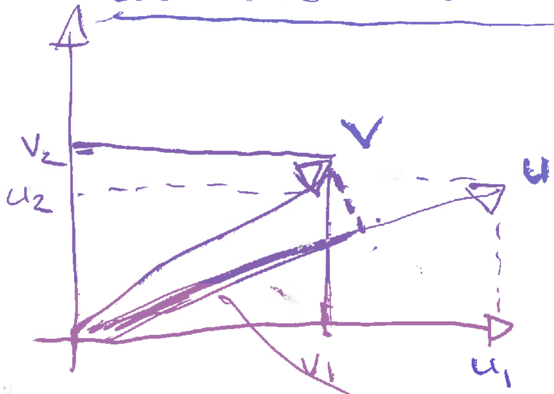
C very large

smaller $\frac{1}{\lambda}$

Week 7

Support Vector Machines – The mathematics behind large margin classification

Vector Inner Product



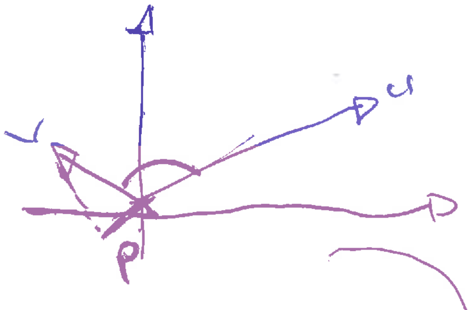
$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ $\quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$

$u^T v = ?$

$\|u\|$ = length of vector u

$= \sqrt{u_1^2 + u_2^2} \leftarrow \mathbb{R}$

p = length of projection of v onto u.



signed $\boxed{u^T v} = p \cdot \|u\| = \boxed{v^T u}$

$= u_1 v_1 + u_2 v_2 \qquad p \in \mathbb{R}.$

$u^T v = p \cdot \|u\|$

$p < 0$ (signed)

SVM decision boundary

$w = (\sqrt{w})^2 = (w^{\frac{1}{2}})^2$

$$\min_\theta \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2}(\theta_1^2 + \theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

$= \|\theta\|$

s.t. $\theta^T x^{(i)} \geq 1$ if $y^{(i)} = 1$

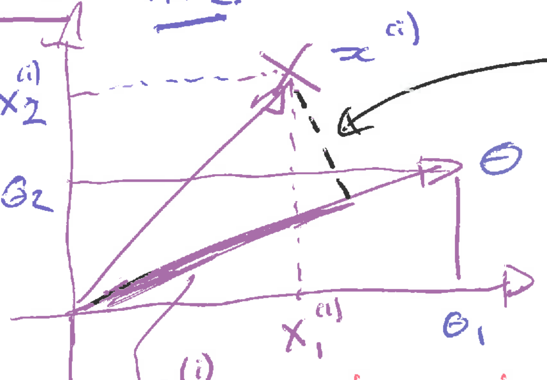
$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$

$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$ $\quad \theta_0 = 0$

Simplification: $\theta_0 = 0$ $\quad n = 2.$

$\theta^T x^{(i)} = ?$

$u^T v$



project $\theta^T x^{(i)} = p^{(i)} \cdot \|\theta\|$ (inner product)

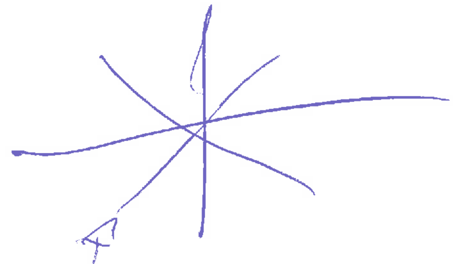
$= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$

$p^{(i)}$ – projection of the i^{th} traning example onto the parameter vector θ

Week 7

Support vector Machines - TMBLMC

SVM Decision boundary



$$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}\|\theta\|^2$$ ← find a setting of parameters where $\|\theta\|$ is small

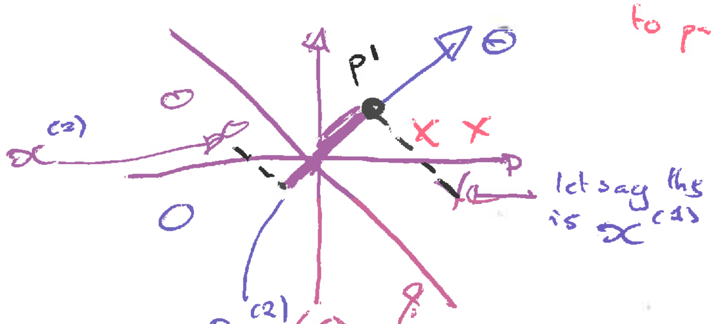
small numbers (from example below)

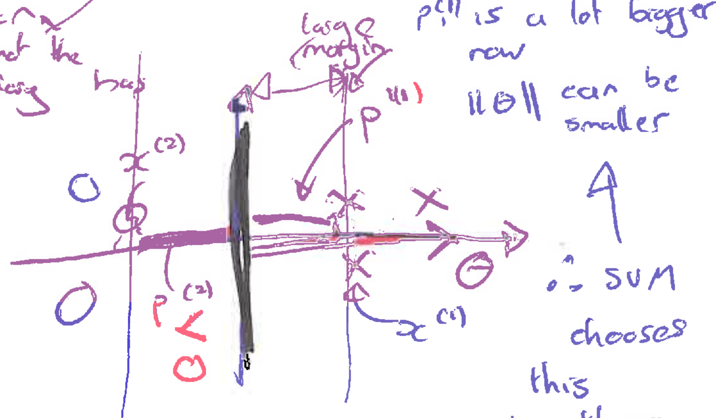
s.t. $p^{(i)} \cdot \|\theta\| \geq 1$ if $y^{(i)} = 1$ ⊗

$p^{(i)} \cdot \|\theta\| \leq -1$ if $y^{(i)} = 1$

where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ

Simplification: $\theta_0 = 0$ → (this means that the decision boundary has to pass through origin) simplification





$$\boxed{\begin{array}{l} p^{(i)} \cdot \|\theta\| \geq 1 \\ \|\theta\| \text{ large} \end{array}}$$ for ⊗ to hold true

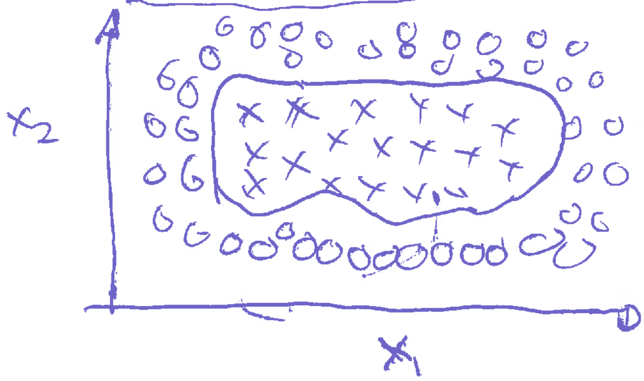
→ what we are doing for SVM's is optimizing θ such that $\|\theta\|$ (norm of theta) is small

∴ SVM's work to minimize $\|\theta\|$.

Week 7

Support Vector Machines – Kernels I

Non-linear decision boundary



Predict $y = 1$ if

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \dots \geq 0$$

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \dots \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

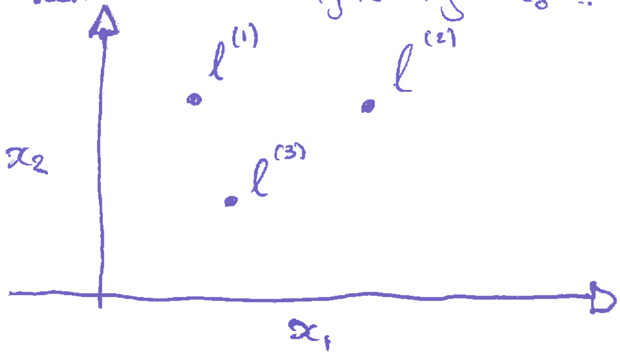
$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \dots$$

$$f_1 = x_1,\ f_2 = x_2,\ f_3 = x_1 x_2,\ f_4 = x_1^2,\ f_5 = x_2^2 \dots$$

Is there a better choice of features than $f_1, f_2, f_3 \dots$?

Kernel

ignoring $x_0 \dots$



Given x, compute new feature depending on proximity $l^{(1)}, l^{(2)}, l^{(3)}$

Given x:

$$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\| x - l^{(1)} \|^2}{2\sigma^2}\right)$$

$$f_2 = \text{similarity}(x, l^{(2)}) = \exp\left(-\frac{\| x - l^{(2)} \|^2}{2\sigma^2}\right)$$

$$f_3 = \text{similarity}(x, l^{(3)}) = \exp(\dots)$$

going to be represented by a kernel function. (Gaussian Kernel) $K(x, l^{(i)})$

Week 7

Support vector machines – Kernels I

Kernels & similarity

also called a "kernel"

this will be close to zero

$$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^{(1)})^2}{2\sigma^2}\right)$$

$l^{(1)} \rightarrow f_1$

$l^{(2)} \rightarrow f_2$ ← given some x.

$l^{(3)} \rightarrow f_3$

If x is close to l...

if $x \approx l^{(1)}$ ← landmark:

$$f_1 \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$$

f will be 1 when close to landmark

if x is far from $l^{(1)}$:

$$f_1 = \exp\left(-\frac{(\text{large numb})^2}{2\sigma^2}\right) \approx 0$$

will be zero when x is far from l

---

example:

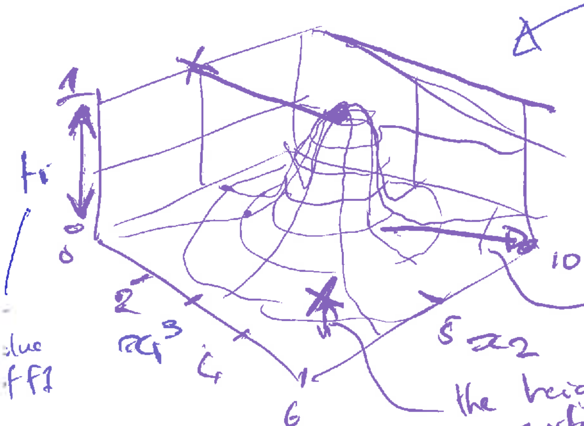
$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\boxed{\sigma^2}}\right)$

$\sigma^2 = 1$

if $\sigma^2 = 0.5$: the width of the contour plot becomes narrower.

If $\sigma^2 = 3$ the width of the contour plot widens and ∴ the value of f1 falls away much more slowly.

This is what f1 looks like

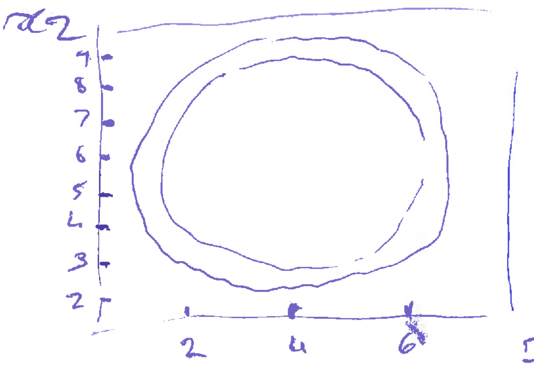


when $x = \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $f_1 = 1$

as x moves away $f_1 \approx 0$.

the height above the surface shows the corresponding value of f1.

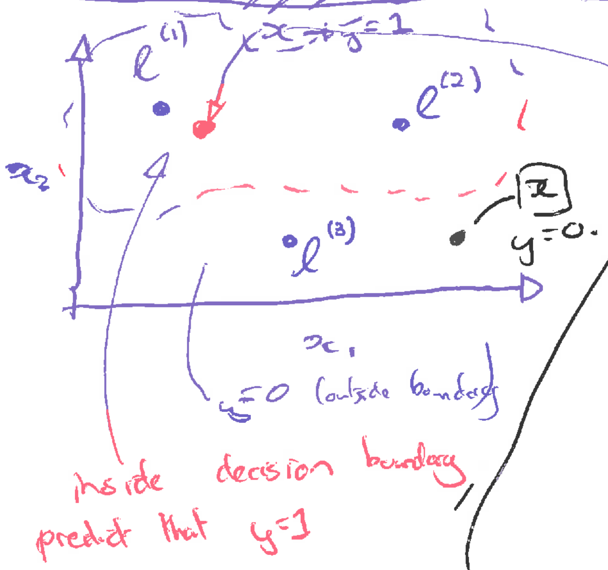
alue f f1



contour plot of 3d surface.

Week 7

Support vector machines — kernels I



Predict "1" when

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$$

x

let's say we already have fcn an example and

$\theta_0 = -0.5, \quad \theta_1 = 1, \quad \theta_2 = 1, \quad \theta_3 = 0$

f_1 is close to l_1 so $f_1 \simeq 1,$	f_2 is far from l^2 so $f_2 \simeq 0$	f_3 same... $f_3 \simeq 0.$

$\therefore \; \theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0$

$= -0.5 + 1 = 0.5 \geq 0 \rightarrow \therefore \; y = 1$

for $\boxed{x}$: $f_1, f_2, f_3 \simeq 0$

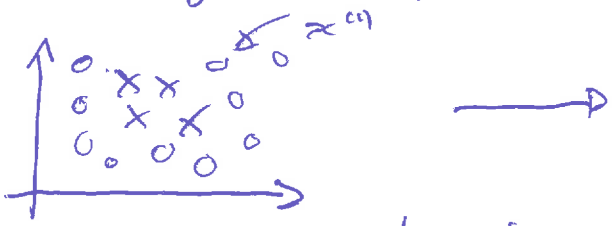
$\theta_0 + \theta_1 f_1 + \theta_2 \ldots \simeq -0.5 < 0.$

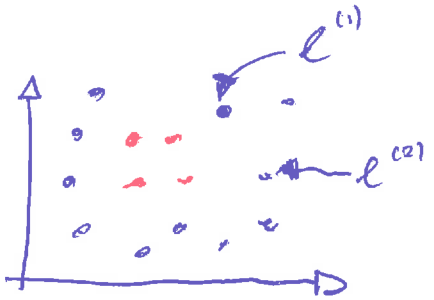
so $\therefore \; y = 0$

Week 7

Support Vector Machines - Kernels II

where to get $\ell^{(1)}, \ell^{(2)}, \ell^{(3)} \dots$?





& set landmark at the same location as the training examples.

what we are going to end up with is:

$\ell^{(1)}$
$\ell^{(2)}$
$\vdots$
$\ell^{(m)}$

} - with one landmark per location for each of the training examples above

SVM with kernels

Given: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

choose: $\ell^{(1)} = x^{(1)}, \ell^{(2)} = x^{(2)}, \dots, \ell^{(m)} = x^{(m)}$ ← exactly the same locations...

Given an example x:

$f_1 = \text{similarity}(x, \ell^{(1)})$

$f_2 = \text{similarity}(x, \ell^{(2)})$

...

→ these give us a feature vector →

$$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad f_0 = 1$$

For training example: $(x^{(i)}, y^{(i)})$:

$x^{(i)} \in \mathbb{R}^{n+1}$ (or $\mathbb{R}^n$)

given $x^{(i)}$ → map it

$f_1^{(i)} = \text{sim}(x^{(i)}, \ell^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, \ell^{(2)})$

$\vdots$

$f_i^{(i)} = \text{sim}(x^{(i)}, \ell^{(i)}) = \exp\left(-\frac{0}{2\sigma}\right) = 1$ ← $x^{(i)}$ & where the similarity is equal to itself....

$f_m^{(i)} = \text{sim}(x^{(i)}, \ell^{(m)})$

$$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix} \quad f_0^{(i)} = 1$$

→ new feature vector with which to represent our new training example

week 7

Support vector machine - kernels II

SVM with kernels

(m traing examples ∴ m landmarks...)

Hypothesis: Given x, compute features $f \in \mathbb{R}^{m+1}$

predict: "$y=1$" if $\boxed{\theta^T f} \geq 0$

$\rightarrow \theta_0 f_0 + \theta_1 f_1 + \dots + \theta_m f_m$

parameters for θ are given by the SVM learning algorithm:

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1-y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2}\sum_{j=1}^{\cancel{n}=m} \theta_j^2$$

$\rightarrow \cancel{\theta^T x^{(i)}}$ $\theta^T f^{(i)}$

(n=m)

solve this minimization problem to get parameters (θ) for SVM.

we still do not (regularize) the parameter θ_0 ...

this term can be computed by

$\sum_j \theta_j^2 = \theta^T \theta$ ← ignoring θ_0

$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}$ [ignore θ_0]

instead of minimizing $\|\theta\|^2$ (norm of Theta^2)

we use $\theta^T M \theta$ ← optimizes something slightly different.

(don't get this!)

if m = 10 000

computational tricks like landmarks ($\ell^{(i)}$ etc) do not generalize well to logistic regression etc.

SVM parameters

$C = \frac{1}{\lambda}$

Large C: lower bias, high variance ← (small λ) → prone to overfitting

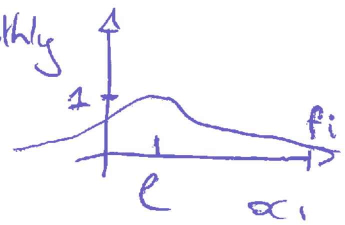
small C: higher bias, low variance. ← (large λ) → prone to underfitting

σ^2 large σ^2: features f_i vary more smoothly

higher bias, lower variance.

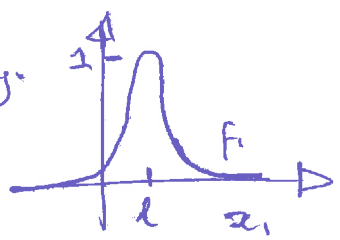
$\exp\left(-\frac{\|x-\ell^{(i)}\|^2}{2\sigma^2}\right)$

← similarity function



small σ^2: features f_i vary less smoothly.

lower bias, higher variance.



Week 7

Support Vector machines - using an SVM

Use svm software package (e.g liblinear, libsvm, ...) to solve for parameters θ,
↳ do not write own software to solve for θ

what we do need to do:

- choice of parameter C
- choice of kernel (similarity function):

e.g. No kernel ("linear kernel")
predict "$y=1$" if $\theta^T x \geq 0$

$\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n \geq 0$

n (number of features large)
m (number of training examples) is small.

Gaussian kernel

$$f_i = \exp\left(-\frac{\|x - \ell^{(i)}\|^2}{2\sigma^2}\right),$$

where $\ell^{(i)} = x^{(i)}$.

$\boxed{\text{Need to choose } \sigma^2}$

n small
m large e $x \in \mathbb{R}^n$

Kernel (similarity) functions:

$$\text{function } f = \text{kernel}(x1, x2)$$

f_i

$x^{(i)}$

$\ell^{(j)} = x^{(j)}$

$x \to f_1, f_2, \dots, f_m.$

$$f = \exp\left(-\frac{\|x1 - x2\|^2}{2\sigma^2}\right)$$

return

Note: Do perform feature scaling before using the gaussian kernel why: because:

normalization is given by:

$\|x - \ell\|^2 \to v = x - \ell$

$\|v\|^2 = v_1^2 + v_2^2 + \dots + v_n^2$

the same e.g.

$$= (x_1 - \ell_1)^2 + (x_2 - \ell_2)^2 + \dots + (x_n - \ell_n)^2$$

if we use the example of houses

could be: m^2

$\boxed{1000\, m^2 \text{ etc}}$

# of bed room

e $\boxed{1-5}$ $\therefore$ need feature scaling

Week 7

Support vector machines - using an SVM

Other choices of kernel:

Note: Not all similarity functions (similarity(x, ℓ)) make valid kernels.
(Need to satisfy technical condition called "Mercer's theorem"
to make sure SVM packages' optimizations run correctly, and
do not diverge).

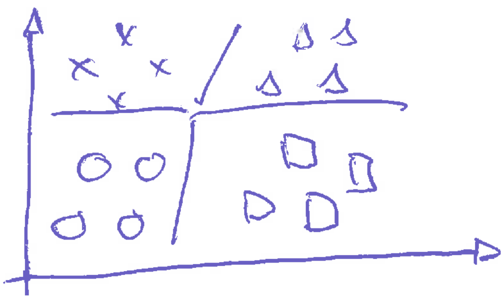
general form
$(x^T \ell + \text{constant})^{\text{degree}}$

Many off the shelf kernel's available:

- polynomial kernel: $k(x, \ell) = (x^T \ell)^2$
  $(x^T\ell)^3$, $(x^T\ell + 1)^3$, $(x^T\ell + 5)^4$
- more esoteric: string kernel, chi-square kernel, histogram intersection...

Multi Class classification

$y \in \{1, 2, 3, \dots k\}$



Many SVM packages already have built in multi-class classification functionality.
Otherwise, use one vs all method. (train k SVMs, one to distinguish $y = i$ from the rest, for $i = 1, 2, \dots, k$), get $\theta^{(1)}, \theta^{(2)}, \dots \theta^{(k)}$
Pick class i with largest $(\theta^{(i)})^T x$.

$y = 1$ (classified as...) $y = 2$. $\theta = k$.

Week 7

SVM with linear kernel or logistic regression

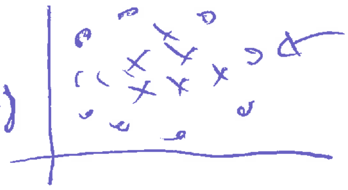
email classification

Support Vector Machines - logistic regression vs. SVMs

Logistic regression vs. SVMs

n = number of features ($x \in \mathbb{R}^{n+1}$), m = number of training examples

- If n is large (relative to m): (eg $n \geq m$) n = 10,000 m = 10 ... 1000
  Use logistic regression, or SVM without a kernel ("linear kernel")
- If n is small, m is intermediate (n = 1 - 1000, m = 10 - 10000)
  use SVM with Gaussian kernel
- If n is small, m is very large (n = 1 - 1000, m = 50000+)
  create / add more features, then use logistic regression or SVM without a kernel.



gaussian kernel

SVM starts to struggle.

- neural network likely to work well for most of these settings, but may be slower to train