# Logistic regression:

- Nice, useful model binary classifier

## Model

$$\log \frac{P(1|x)}{P(-1|x)} = w^T x$$

you can have a constant here by extending $x$

So $$P(1|x) = \frac{e^{w^T x}}{1 + e^{w^T x}}$$

We can fit with maximum likelihood

$$\mathcal{L}(w) = \sum_{i \in \text{examples}} \log P(y_i | x_i)$$

this is equiv to minimizing -ve log-likelihood

$$w = \underset{w}{\arg\min} \; - \sum_{i \in \text{examples}} \log P(y_i | x_i)$$

$$= \underset{w}{\arg\min} \; - \sum_{i} \left[ \frac{(y_i + 1)}{2} \, w^T x_i - \log\left(1 + e^{w^T x_i}\right) \right]$$

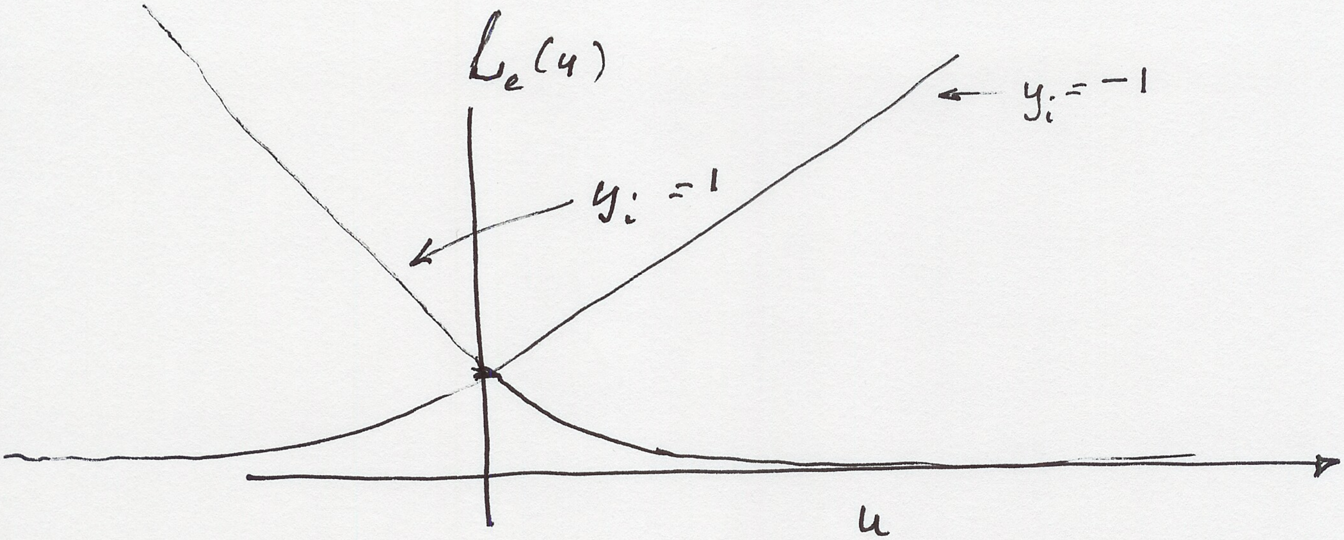
Now regard

$$\cancel{=} \; \log\left(1 + e^{\cancel{w^T x_i} \, u_i}\right) - \frac{(y_i + 1)}{2} \, \cancel{w^T x_i} \, u_i$$ as a loss

$L(y_i, u_i)$ ← exponential loss

value of example

score $= w^T x_i$




looks a lot like hinge loss:

So we can think of LR as minimizing a loss function.

How? $\nabla = 0$, Newton.

$$\nabla_w L_e = \sum_i \left[ \frac{x_i}{1 + e^{w^T x_i}} - \frac{(y_i + 1)}{2} x_i \right]$$

$$H_w L_e = \sum_i \frac{e^{w^T x_i}}{(1 + e^{w^T x_i})^2} \cdot x_i x_i^T$$

Notice:

- examples where $w^T x_i$ has large abs value have little effect on $H$.
- For ~~those~~ others, $H$ looks like a covariance
- what if features are correlated?
  - $H$ has small eigenvalues
  - ests of $w$ will be unreliable in these dirns.
  - Should manifest as $\underline{\text{large}}$ $w$

$\Longrightarrow$ Regularize.

Solve

$$\underset{w}{\operatorname{argmin}} \quad \mathcal{L}_e(w, \text{examples}) + \lambda \|w\|_?$$

Some norm

# Stochastic gradeint descent

Many learning problems look like

$$\arg\min_{w} \; L(w, x_i, y_i) + \lambda \|w\|_2$$

loss on examples

some regularize typically 2-norm or 1-norm.

eg. SVM.

loss is $\sum_i$ hinge loss ( $w$, example $i$ )

eg. logistic regression

logs is $\sum_i$ exp loss.

In these cases, two important obs.

1) We don't so much care about

$$\mathcal{E} = \sum_{i \in \text{examples}} L(w, \text{example}_i) \quad \text{as about}$$

($\mathcal{E}$ for empirical loss)

$$\sum_{i \in \underline{\text{everything}}} L(w, \text{example}_i). = J$$

for true loss

this means that the min of the optimization problem $w^*$, may not be best $w$ (which is $w^{opt}$)

$$w^{opt} = \arg\min J$$

$$w^* = \arg\min \mathcal{E}$$

error incurred by exact optimization

$$\boxed{\Rightarrow \text{This means inexact opt might be OK.}}$$

2) Exact optimization could be <u>hugely</u> expensive

(millions of examples; millions of feats)

<u>Idea:</u> Pick example (feature) at random, compute gradient for that alone, take small step.

<u>STOCHASTIC GRADIENT DESCENT</u>

- How far?

  steplengths $\delta_i$ should have property

  - $\delta_N \longrightarrow 0$, $N \to \infty$
  - $\sum_{i=1}^{R} \delta_i \longrightarrow \infty$ $R \to \infty$

(ie. path ~~can~~ must be infinitely long, but steps must become infinitely small)

## e.g. Logistic regression with 2 Norm

$$\min_{w} \sum_i \left[ \mathcal{L}_e(\omega, y, x) + \frac{\lambda}{N} \|w\|_2^2 \right] = E + R$$

- for each step choose 1 example$_N$ UAR. Say ~~a~~ $k$'th

$$\cancel{\nabla} g^{(n)} = \left[ \frac{x_k}{1 + e^{w^{(n)T} x_k}} - \left( \frac{y_k + 1}{2} \right) x_k \right] + \left[ \frac{2\lambda}{N} w^{(n)} \right]$$

$$w^{(n+1)} = w^{(n)} - \delta_i \cdot g^{(n)}$$

- Notice, because we choose UAR,

$$E(g^{(n)}) = \nabla_w [E + R]$$

typically, step lengths look like

$$\delta_i = \frac{1}{i}$$

(this satisfies constraints).

We get (with work)

$$\frac{1}{(w_t - w^*)^2}$$ grows <u>linearly</u> in $t$

<u>i.e.</u> early steps really help, late steps don't do much.

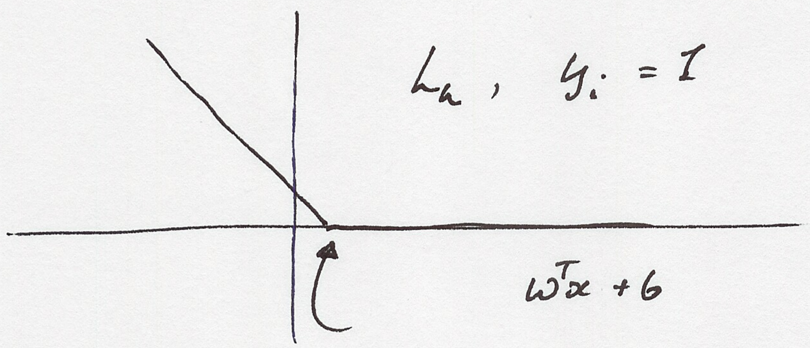
<u>Notice</u>: <u>algorithm is on-line</u>

What about SVM?

$$\sum_i \left[ L_h(w, x_i, y_i) + \frac{\lambda}{N} w^T w \right]$$

which isnt differentiable.

$$L_h = \max(0, 1 - y_i(w^T x_i + b))$$




however, this is convex.

- We invoke the <u>Sub-gradient</u>

consider the <u>graph</u> of a function

$$(x_1, x_2, x_3, \cdot, x_n, f(x_1 \cdot\cdot x_n))$$

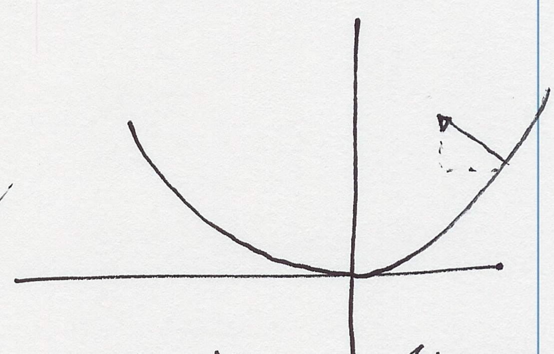
this is a <u>surface</u>.

- assume ~~the~~ $f$ is differentiable
  — surface has a <u>normal</u>

$$N = k\left(-\frac{\partial f}{\partial x_1}, -\frac{\partial f}{\partial x_2}, \cdots, +1\right)$$

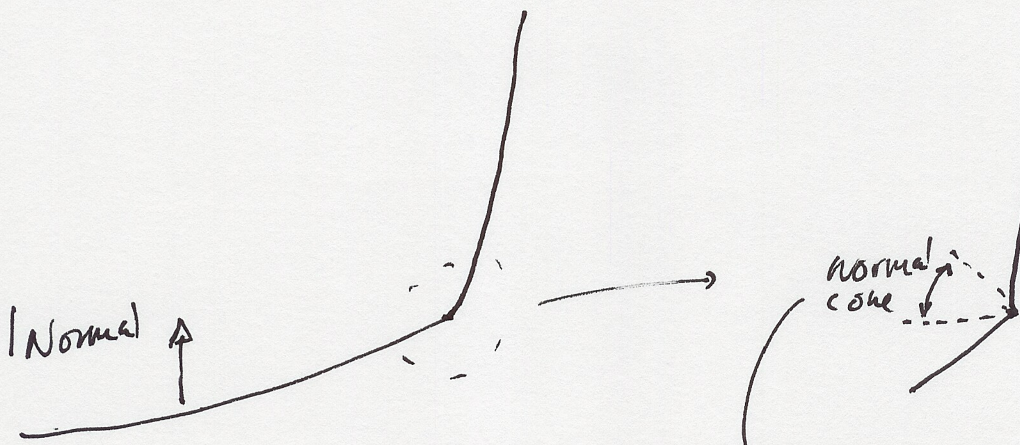
$$= k(-\nabla f, +1)$$

eg curve in the plane is graph of fn of 1 var



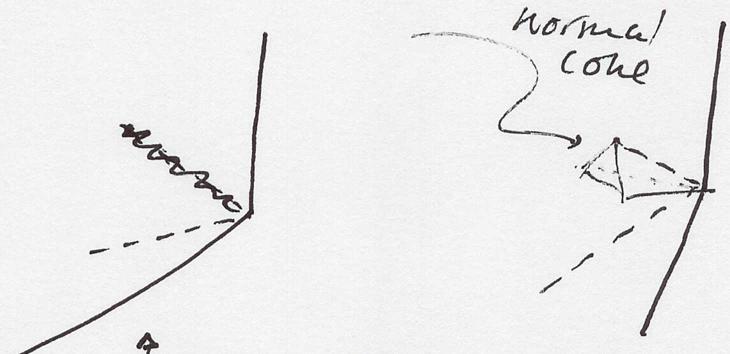
Notice we can read <u>gradient</u> off <u>normal</u>

Now, if ~~curve~~ function is <u>not</u> diff, we can come up w/ normal <u>cone</u>




all of these vectors are normals at this pt.

( • Notice, f needs to be convex so we know what interior is for this construction )




graph of fn of two vars

- read off gradient corresp to any element of normal cone
- this is subgradient ( $\tilde{\nabla}$ )
- moving backward along subgradient will guarantee descent for small enough step
- subgradient of diff. fn = gradient

SVM:

$$\tilde{\nabla}_w L_h(w) = \begin{cases} 0 & \text{if } L_h(w) = 0 \\ -y_i x_i & \text{otherwise} \end{cases}$$

So ~~Subgr~~ Stochastic Subgradient Descent is

- choose k'th example at random
- if right, $w^{(n+1)} = w^{(n)}\left(1 - \frac{\lambda \delta_i}{2N}\right)$
- if wrong, $w^{(n+1)} = w^{(n)} - \delta_i \left[ -y_i x_i + \frac{\lambda w^{(n)}}{2N} \right]$

This is amazingly effective.

# $L_1$ regularization

$$\|w\|_1 = \sum_i |w_i|$$

$$\|w\|_2^2 = \sum_i w_i^2$$

<u>Notice</u>: in $L_2$ norm, small $w_i$ have little effect hence, not much advantage in driving to zero; in $L_1$ norm, effect of small $w_i$ is substantial; they <u>tend</u> to go to zero, resulting in <u>sparsity</u> (desirable)

eg <u>$L_1$ reg logistic regression</u>

$$\min_{w} \sum_i L_e(w, y_i, x_i) \quad + \lambda \|w\|_1$$

- not differentiable
- subgradient unlikely to enforce sparsity

<u>alternative</u> (equivalent)

$$\min_{w} \sum_i L_e(w, y_i, x_i) + \sum_e h_e$$

$$-h_k \leq w_k \leq h_k$$

(Notice this is a box problem;
solve with interior point method;
excellent version due to
Koh et al)