Logistic regression:

- Nice, useful model binary classifier

Model

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

So

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

We can fit with maximum likelihood

\[
L(w) = \sum_{i \in \text{examples}} \log P(y_i| x_i)
\]
This is equiv to minimizing -ve log-likelihood:

\[ w = \arg \min_w - \sum_i \log P(y_i | x_i) \]

\[ = \arg \min_w - \sum_i \left[ \frac{(y_i + 1) w^t x_i}{2} - \log(1 + e^{w^t x_i}) \right] \]

Now regard

\[ \log(1 + e^{w^t x_i}) - \frac{(y_i + 1) w^t x_i}{2} \]

as a loss.

Value of example

Score = \( w^t x_i \)

\[ \mathcal{L}(y_i, u_i) \]

\[ \text{exponential loss} \]
looks a lot like hinge loss:

So we can think of L_i as minimizing a loss function. How? \( \nabla = 0 \), Newton.

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

\[
H w L_e = \sum_i \frac{e^{w x_i}}{(1 + e^{w x_i})^2} \cdot x_i x_i^T
\]
Notice:

- examples where $w^x_i$ has large abs values have little effect on $H$.
- For these others, $H$ looks like a covariance

- what if features are correlated?
  - $H$ has small eigenvalues
  - ests of $w$ will be unreliable in these dirs.
  - Should manifest as large $w$

$\Rightarrow$ Regularize.
Solve

argmin \| x \| \text{ s.t. } L_w(w, \text{examples}) + \lambda \| w \|_w \leq ?

Some norm
Stochastic gradient descent

Many learning problems look like

\[ \arg \min_w L(w, x; y) + \lambda \|w\| \]

\[ \uparrow \]

loss on examples

eg. SVM.

loss is \[ \sum_i \text{hinge loss}(w, \text{example } i) \]

eg. logistic regression

loss is \[ \sum_i \text{exp loss} \]
In these cases, two important obs.

1) We don't so much care about

\[ E = \sum_{\text{examples}} L(w, \text{example}_i) \text{ as about} \]

\[ \sum_{\text{everything}} L(w, \text{example}_i) = \int \text{ for true loss} \]

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

\[ w^{opt} = \arg\min J \]

\[ w^* = \arg\min E \]

\[ \Rightarrow \text{error incurred by exact optimization} \]

\[ \Rightarrow \text{This means inexact opt might be OK.} \]
2) Exact optimization could be hugely expensive

(millions of examples, millions of feats)

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

STOCHASTIC GRADIENT DESCENT

. How far?

- Steplengths \( S_i \) should have property

  - \( S_N^{\frac{R}{N}} \to 0 \), \( N \to \infty \)
  - \( \sum_{i=1}^{R} S_i \to \infty \), \( R \to \infty \)

(ie. path must be infinitely long, but steps must become infinitely small)
Logistic regression with $\ell_2$ norm

\[ \min_w \sum_i \left[ L_e(w, y, x) + \frac{\lambda}{N} \| w \|_2^2 \right] = E + R \]

for each step $\text{UAR}$, choose $I$ examples, say $x_{k_i}$

\[ g^{(n)} = \left[ \frac{x_k}{1 + e^{x_i w^{(n-1)} x_k}} - \left( \frac{y_k + 1}{2} \right) x_k \right] \]

\[ w^{(n+1)} = w^{(n)} + \delta_i \cdot g^{(n)} \]

Notice, because we choose $\text{UAR}$,

\[ E(g^{(n)}) = \nabla_w [E + R] \]
typically, step lengths look like

\[ s_i = \frac{1}{i^c} \]

(this satisfies constraints).

We get (with work)

\[ \frac{1}{(w_t - w^*)^2} \]

grows linearly in \( t \)

i.e. early steps really help, late steps don't do much.

Notice: algorithm is on-line
What about SVM?

\[ \sum \left[ l_h(w, x_i, y_i) + \frac{1}{N} \omega^2 \right] \]

which isn't differentiable.

\[ l_h = \max(0, 1 - y_i(w^T x_i + b)) \]

\[ l_h, y_i = 1 \]

However, this is convex.

We invoke the sub-gradient
Consider the graph of a function

\((x_1, x_2, x_3, \ldots, x_n, f(x_1, x_n))\)

This is a **surface**.

- Assume \( f \) is differentiable
  - Surface has a **normal**

\[
N = k \left( -\frac{\partial f}{\partial x_1}, -\frac{\partial f}{\partial x_2}, \ldots, +1 \right)
= k \left( -\nabla f, +1 \right)
\]

**eg:** curve in the plane is graph of \( f(u, v) \) of 1 var

Notice we can read gradient off normal
Now, if the function is not differentiable, we can come up with a normal cone.

- Notice, if needs to be convex so we know what inferior is for this construction.

\[ \text{Normal cone} \]

\[ \text{Graph of } f = \text{of two vars} \]
read off gradient corresp to any element of normal cone
this is subgradient (\nabla)
moving backward along subgradient will guarantee descent for small enough step
Subgradient of diff. fn = gradient

\[
\nabla_{w} L_{h}(w) = \begin{cases} 
0 & \text{if } L_{h}(w) = 0 \\
-y_{i} x_{i} & \text{otherwise}
\end{cases}
\]
So stochastic subgradient descent is

\[
\begin{cases}
    \text{choose } k\text{'th example at random} \\
    \text{if right, } w^{(n+1)} = w^{(n)} - \frac{\lambda \delta_i}{2N} \\
    \text{if wrong, } w^{(n+1)} = w^{(n)} + 8i \left[ -y_i x_i \cdot \frac{\lambda w^{(n)}}{2N} \right]
\end{cases}
\]

This is amazingly effective.
L1 regularization

\[ \|w\|_1 = \sum_i |w_i| \]
\[ \|w\|_2 = \sum_i w_i^2 \]

**Notice:** in L2 norm, small \(w_i\) have little effect hence, not much advantage in trying to Zero; in L1 norm, effect of small \(w_i\) is substantial; they tend to go to Zero, resulting in sparsity (desirable)
\[ \min_{\mathbf{w}} \sum_{i} h_{e}(\mathbf{w}, y_{i}, x_{i}) + \lambda \| \mathbf{w} \|, \]

- not differentiable
- subgradient unlikely to enforce sparsity

alternative (equivalent)

\[ \min_{\mathbf{w}} \sum_{i} h_{e}(\mathbf{w}, y_{i}, x_{i}) + \sum_{e} h_{e}. \]

\[ -h_{k} \leq w_{k} \leq h_{k} \]

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