ISTA 421 + INFO 521
Introduction to Machine Learning

Lecture 15:
Logistic Regression,
Gradient descent

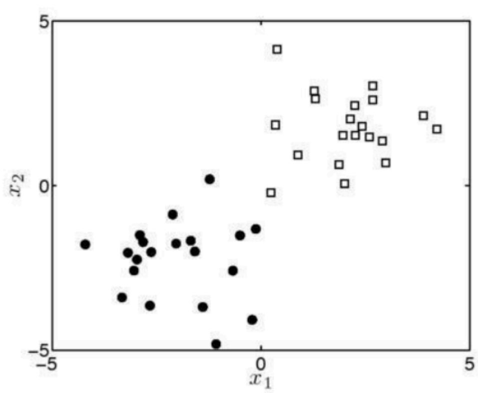
Clay Morrison
claytonm@email.arizona.edu
Harvill 437A
Phone 621-6609

16 October 2017

1

Binary Classification!

- A very common type of problem
- *Many* different approaches; we'll start with a probabilistic method: logistic regression



two attributes (x_1 and x_2)

binary target, $t = \{0, 1\}$

t = 0 are dark circles
t = 1 are white squares

2

The Likelihood

- Assume the elements of **t** are independent, conditioned on **w**

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w})$$

- In linear regression, **t** was Gaussian (normal) distributed b/c the target was real-valued. Now the target is a binary class label (0 or 1), so likelihood is a different RV:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(T_n = t_n|\mathbf{x}_n, \mathbf{w})$$

a binary random variable

3

The Likelihood

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(T_n = t_n|\mathbf{x}_n, \mathbf{w})$$

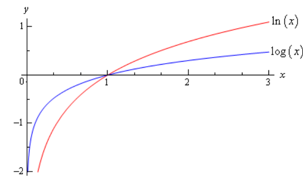
- Want likelihood to…
  … be high if model assigns
    high probabilities for class 1 when we observe class 1, ***and***
    high probabilities for class 0 when we observe class 0.
  … have a maximum value of 1 where all of the training points are predicted perfectly.
- **Popular approach**: take simple linear function and pass the result through a second function that "squashes" its output, to ensure it produces a valid probability.

4

The Log-odds

- The logistic function is formally derived as a result of a linear model of the ***log-odds*** (aka the ***logit***):

$$\log\left(\frac{P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})}{P(T_{\text{new}} = 0|\mathbf{x}_{\text{new}}, \mathbf{w})}\right) = \mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}}$$



- There are no constraints on this value: it can take any real value (-∞, ∞).

$P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w}) \ll P(T_{\text{new}} = 0|\mathbf{x}_{\text{new}}, \mathbf{w})$ Large *negative*

$P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w}) \gg P(T_{\text{new}} = 0|\mathbf{x}_{\text{new}}, \mathbf{w})$ Large positive

5

From the Logit to Logistic Function

Example of a **generalized linear model**: linear model passed through a transformation to model a quantity of interest.

- Now, derive $P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})$

Note: $P(T_{\text{new}} = 0|\mathbf{x}_{\text{new}}, \mathbf{w}) = 1 - P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})$

$$\begin{aligned}
\log\left(\frac{P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})}{P(T_{new} = 0|\mathbf{x}_{new}, \mathbf{w})}\right) &= \mathbf{w}^{\top}\mathbf{x} \\
\frac{P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})}{P(T_{new} = 0|\mathbf{x}_{new}, \mathbf{w})} &= \exp(\mathbf{w}^{\top}\mathbf{x}) \\
\frac{P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})}{1 - P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})} &= \exp(\mathbf{w}^{\top}\mathbf{x}) \\
P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) &= \exp(\mathbf{w}^{\top}\mathbf{x})(1 - P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})) \\
P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) &= \exp(\mathbf{w}^{\top}\mathbf{x}) - \exp(\mathbf{w}^{\top}\mathbf{x})P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) \\
P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) + \exp(\mathbf{w}^{\top}\mathbf{x})P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) &= \exp(\mathbf{w}^{\top}\mathbf{x}) \\
P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w})(1 + \exp(\mathbf{w}^{\top}\mathbf{x})) &= \exp(\mathbf{w}^{\top}\mathbf{x}) \\
P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) &= \frac{\exp(\mathbf{w}^{\top}\mathbf{x})}{1 + \exp(\mathbf{w}^{\top}\mathbf{x})} \\
P(T_{new} = 1|\mathbf{x}_{new}, \mathbf{w}) &= \frac{1}{1 + \exp(-\mathbf{w}^{\top}\mathbf{x})}
\end{aligned}$$

So the logistic function is really modeling the log-odds with a linear model!

The **Logistic** function (the inverse Logit)

6

Logistic as Likelihood

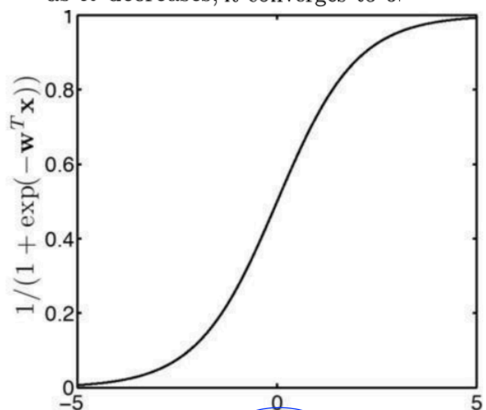
$$p(\mathbf{t}|\mathbf{X},\mathbf{w}) = \prod_{n=1}^{N} P(T_n = t_n|\mathbf{x}_n,\mathbf{w})$$

The Logistic (or sigmoid) function:

$$P(T_n = 1|\mathbf{x}_n,\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)}$$

Linear component

As $\mathbf{w}^\mathsf{T}\mathbf{x}$ increases, the value converges to 1 as it decreases, it converges to 0.



When target is 0:

$$\begin{aligned} P(T_n = 0|\mathbf{x}_n,\mathbf{w}) &= 1 - P(T_n = 1|\mathbf{x}_n,\mathbf{w}) \\ &= 1 - \frac{1}{1+\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)} \\ &= \frac{\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)}{1+\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)}. \end{aligned}$$

Combine both into a single probability function

$$P(T_n = t_n|\mathbf{x}_n,\mathbf{w}) = P(T_n = 1|\mathbf{x}_n,\mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n,\mathbf{w})^{1-t_n}$$

$\mathbf{w}^T\mathbf{x}$ (Note! Not just fn of x)

The Likelihood

$$P(T_n = t_n|\mathbf{x}_n,\mathbf{w}) = P(T_n = 1|\mathbf{x}_n,\mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n,\mathbf{w})^{1-t_n}$$

$$\begin{aligned} p(\mathbf{t}|\mathbf{X},\mathbf{w}) &= \prod_{n=1}^{N} P(T_n = t_n|\mathbf{x}_n,\mathbf{w}) \\ &= \prod_{n=1}^{N} P(T_n = 1|\mathbf{x}_n\mathbf{w})^{t_n} P(T_n = 0|\mathbf{x}_n,\mathbf{w})^{1-t_n} \\ &= \prod_{n=1}^{N} \left(\frac{1}{1+\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)}\right)^{t_n} \left(\frac{\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)}{1+\exp(-\mathbf{w}^\mathsf{T}\mathbf{x}_n)}\right)^{1-t_n} \end{aligned}$$

Substitute in the component likelihoods to get the final likelihood function

Problem (1): No closed form solution for solving for **w**

8

Bayesian Logistic Regression

$$\mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix}, \; \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \; \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}$$

Want to compute the posterior density over the parameters **w** of the model

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})}$$

$$p(\mathbf{t}|\mathbf{X}) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}) \; d\mathbf{w}$$

Prior: $p(\mathbf{w}) = \mathcal{N}\;(\mathbf{0}, \; \sigma^2\mathbf{I})$

9

Likelihood:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} \left(\frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x}_n)}\right)^{t_n} \left(\frac{\exp(-\mathbf{w}^\top \mathbf{x}_n)}{1+\exp(-\mathbf{w}^\top \mathbf{x}_n)}\right)^{1-t_n}$$

Prior:

$$p(\mathbf{w}|\sigma^2) = \mathcal{N}(0, \sigma^2\mathbf{I})$$

Once we have the Posterior…

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}, \mathbf{X}, \sigma^2)}$$

… can predict the response (class) of new objects by taking the expectation with respect to this density:

$$P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = \mathbb{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)}\left\{\frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x}_{\text{new}})}\right\}$$

Problem 2: the posterior is not in a standard form.

The numerator is fine: just calc prior and likelihood at observations, then multiply.

It's the denominator (marginal likelihood) that is the problem: can't integrate…

$$Z = p(\mathbf{t}|\mathbf{X}, \sigma^2) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)d\mathbf{w}$$

10

In Sum: Problems Optimizing Logistic Regression

- **Problem 1**: cannot directly compute max likelihood (or MAP) – cannot isolate **w**

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} \left(\frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x}_n)}\right)^{t_n} \left(\frac{\exp(-\mathbf{w}^\top \mathbf{x}_n)}{1+\exp(-\mathbf{w}^\top \mathbf{x}_n)}\right)^{1-t_n}$$

- **Problem 2**: cannot compute full Bayes because cannot integrate Bayes denominator (marginal likelihood)

$$p(\mathbf{t}|\mathbf{X}, \sigma^2) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) d\mathbf{w}$$

11

Numerator of Bayes Theorem

$$g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = \underbrace{p(\mathbf{t}|\mathbf{X}, \mathbf{w})}^{\text{likelihood}} \overbrace{p(\mathbf{w}|\sigma^2)}^{\text{prior}}$$

marginal Likelihood (denominator)

$$Z = p(\mathbf{t}|\mathbf{X}, \sigma^2) = \int p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) d\mathbf{w}$$

The Posterior

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = Z^{-1} g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$$

Our Options

1. Find the single value of **w** that corresponds to the highest value of the likelihood or posterior. As *g* is proportional to the posterior, a maximum of *g* will also correspond to a maximum of the posterior. Z^{-1} is not a function of **w**
2. Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with some other density that we can compute analytically.
3. Sample directly from the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$, knowing only $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$

12

Method 1: point estimate

- We can find **w** that maximizes the likelihood (MLE) or the posterior (MAP).
- **Consider MAP**: while we cannot derive a direct analytic posterior density, we can compute something proportional to it:

$$g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$$

- We will find the value of **w** that maximizes *g*
- This will correspond to the value at the maximum of the posterior.
- This will be the most likely value $\hat{\mathbf{w}}$ under the posterior.

13

The MAP Estimate

- It will again be helpful to work with the log

$$\log g(\mathbf{w}; \mathbf{X}, \mathbf{t}) = \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w}|\sigma^2)$$

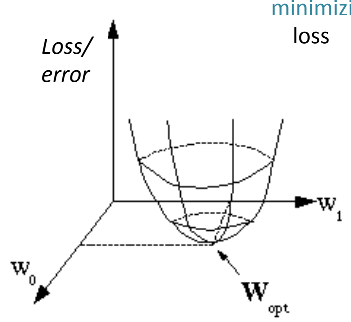
- As we've seen, unlike the max likelihood solution for the linear model, we **cannot** get an exact analytical expression for **w** by differentiating this expression and setting it to 0.
- Instead, need to use an **iterative optimization method**: guess value of **w** and apply *incremental update adjustments* to our estimate of **w** that increase log *g* until a maximum is found.

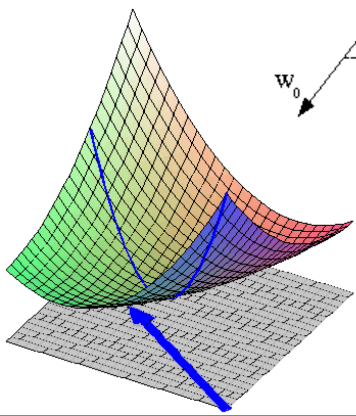
14

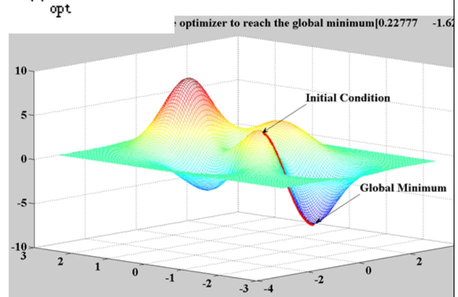
Gradient Methods: Descent/Ascent

minimizing
loss

maximizing
likelihood (MLE)
posterior (MAP)







Using the Squared Loss (of the linear reg model) to define an incremental update to w based on the gradient of w

- Recall the matrix version of the squared loss, multiplied out for easier differentiation

$$\mathcal{L} = \frac{1}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^\top\mathbf{X}^\top\mathbf{t}$$

- Drop the constant 2/N's

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{w}} &= \frac{2}{N}\mathbf{X}^\top\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{X}^\top\mathbf{t} \\ &= \mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{t}\end{aligned}$$

16

The parameter update rule (Widrow-Hoff)
(applied to linear least squares)

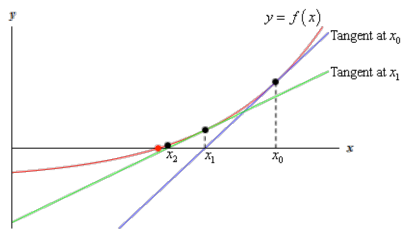
- The ***batch*** update version

$$\mathbf{w} := \boxed{\mathbf{w} - \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{w}}}$$
$$:= \mathbf{w} - \alpha \left(\mathbf{X}^\top \mathbf{X} \mathbf{w} - \mathbf{X}^\top \mathbf{t} \right)$$
$$:= \mathbf{w} - \alpha \sum_{n=1}^{N} \left(t_n - \mathbf{w}^\top \mathbf{x}_n \right) \mathbf{x}_n$$

Properties:
1. (1) Step is in the direction of the gradient's steepest descent (negative of tangent slope)
2. (2) Magnitude of the update is proportional to the error term and scaled by α

The "algorithm"

repeat
$\quad \mathbf{w} := \mathbf{w} - \alpha \sum_{n=1}^{N} \left(t_n - \mathbf{w}^\top \mathbf{x}_n \right) \mathbf{x}_n$
until *convergence*;

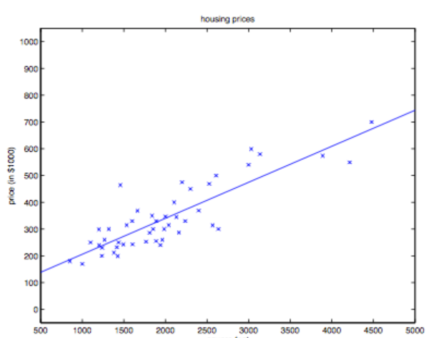


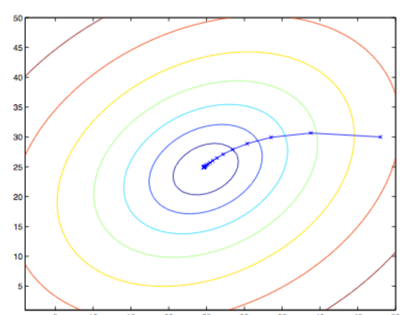
Version with design matrix computations (equivalent)

repeat
$\quad \mathbf{w} := \mathbf{w} - \alpha \left(\mathbf{X}^\top \mathbf{X} \mathbf{w} - \mathbf{X}^\top \mathbf{t} \right)$
until *convergence*;

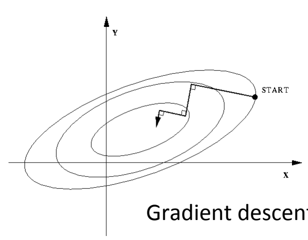
17

Gradient Descent with line in 2D





Plot of Loss gradient in space of Parameters: w_0 and w_1



Gradient descent path is not always so "smooth"

18

Exploring the Landscape

- **Local Maxima**: peaks that aren't the highest point in the space
- **Plateaus:** the space has a broad flat region that gives the search algorithm no direction (random walk)
- **Ridges:** flat like a plateau, but with drop-offs to the sides; steps to the North, East, South and West may go down, but a step to the NW may go up.

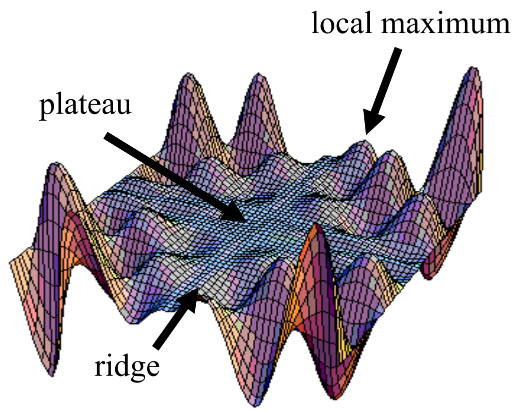


Image from: http://classes.yale.edu/fractals/CA/GA/Fitness/Fitness.html

19

Batch vs. Incremental (Stochastic)

- Batch gradient descent has to scan through the entire training set before taking a single step.
- Costly if *N* is large
- ***Stochastic* gradient descent** can start making progress right away, and continues to make progress with each example it looks at.

20

The parameter update rule (Widrow-Hoff) for ***<u>Stochastic</u>*** Gradient Descent

- The single instance version (for stochastic g.d.)

$$\begin{aligned} \mathbf{w} &:= \mathbf{w} - \alpha \frac{\partial \mathcal{L}_n}{\partial \mathbf{w}} \\ &:= \mathbf{w} - \alpha \left(t_n - \mathbf{w}^\top \mathbf{x}_n\right) \mathbf{x}_n \end{aligned}$$

(Update per **w** vector element:

$$w_j := w_j - \alpha \left(t_n - \mathbf{w}^\top \mathbf{x}_n\right) x_{n,j}$$)

- The "algorithm":

repeat
  for $n = 1$ *to* N **do**
    $\mathbf{w} := \mathbf{w} - \alpha \left(t_n - \mathbf{w}^\top \mathbf{x}_n\right) \mathbf{x}_n$
  end
until *convergence*;

21

Incremental / Stochastic Gradient Descent

- Often, stochastic gradient descent gets **w** "close" to the minimum much faster than batch gradient descent.
- NOTE: however, it may never "converge" to the minimum, as the parameters **w** may oscillate around the minimum of the *Loss*
- But in practice, most of the values near the minimum will be reasonably good approximations of optimal.
- It is more common to run stochastic gradient descent with a fixed learning rate (alpha). However, theoretically, by slowly decreasing the learning rate to zero at the right rate, it is possible to ensure that the parameters will converge to the global minimum rather than merely oscillate around the minimum.

22