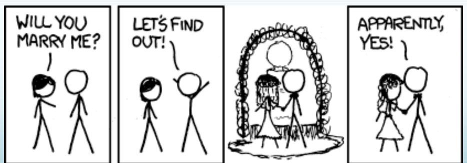
From Regression to Classification with Logistic Regression

Mark Dredze

Machine Learning
CS 601.475



Classification

- Data $\{(x_i, y_i)\}_{i=1}^{N}$ $x_i \in \Re^M$ $y_i \in L$
- Learn: a mapping from x to discrete value y
  - f(x) = y
- Examples
  - Spam classification
  - Document topic classification
  - Identifying faces in images

Binary Classification

- We'll focus on binary classification
  - $y_i \in \{0,1\}$
- Usually easy to generalize to multi-class classification

Different Definition

Fitting a function to data

- Fitting: Optimization, what parameters can we change?
- **Function: Model, loss function**
- Data: Data/model assumptions? How we use data?

- ML Algorithms: minimize a function on some data

Evaluation

- Accuracy

$$\frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

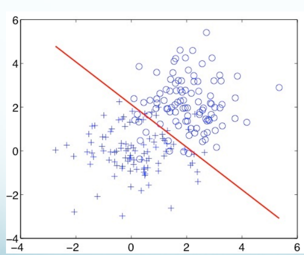
- Other measurements appropriate for some tasks
  - Ex. we care more about certain types of mistakes

Regression

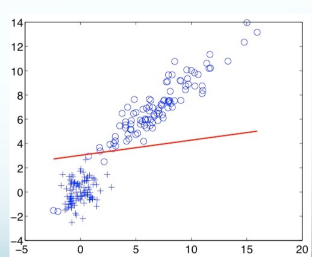
- Least squares regression
  - Outputs real number for each example
- It seems that classification should be easier!
- Let's use regression for classification
  - Learn least squares regression model $f_w(x)=y$
    - $f_w(x)=w^Tx$
  - If $y>0$, predict "True (1)"
  - If $y\leq 0$ predict "False (0)"

Regression for Classification

- $f_w(x)=0$ partitions the input space into two class specific regions
  - Linear decision boundary



Good



Bad

Figures by Tommi Jaakkola

Regression for Classification

- Mismatch between regression loss and classification
  - Classification: accuracy
  - We don't care about large vs. small values of output
- Outliers problematic
  - Prediction of 42 for example is fine for classification, bad for regression
- We need output to be either 1 or 0

Machine Learning

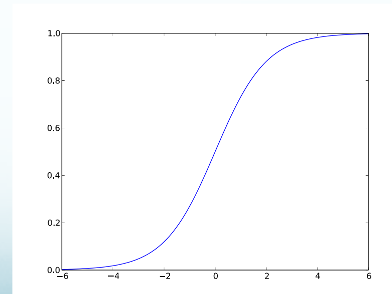
Fitting a function to data

- Fitting: Solve for w given y and x
- Function: Regression uses squared loss
  - Bad match for our task!
- Data: assume dependent variable linear combination of independent variables
- Our loss function doesn't match classification goals

Logistic Function

- Quick fix: apply a function to the output of regression that gives desired valued
- Logistic function
  - Outputs between 0 and 1
  - Scaling parameter α
  - Most outputs are close to 1 or 0

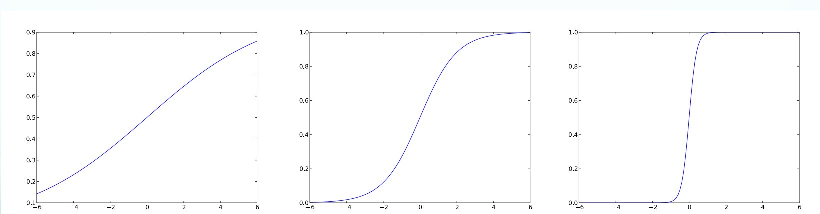
$$g_{\alpha}(x) = \frac{1}{1+e^{-\alpha x}}$$



α = 1

Logistic Function

$$g_{\alpha}(x) = \frac{1}{1+e^{-\alpha x}}$$



α = 0.3 α = 1 α = 5

Logistic Regression

- We can combine the logistic function and our regression model

$$g(w^T \cdot x_i) = \frac{1}{1+e^{-w^T \cdot x_i}}$$

- Notice: as $w^T \cdot x_i$ becomes:
  - Large- output closer to 1
  - Small- output closer to 0

Probabilistic View

- We want to model the probability of a label given the example
- Conditional likelihood $p(y \mid x)$
- Consider
  - We could maximize the joint: $p(y, x)$
  - Which can be factored as: $p(x \mid y)p(y)$
  - However, the best label y is the same under both:

$$\arg\max_{y=0,1} p(x \mid y)p(y) = \arg\max_{y=0,1} p(y \mid x)$$

  - Because x is fixed/given

Probabilistic View

- We can now write the distribution as

$$p_w(y=1 \mid x) = \frac{1}{1+e^{-w^T \cdot x}}$$

- Which implies that

$$p_w(y=0 \mid x) = \frac{e^{-w^T \cdot x}}{1+e^{-w^T \cdot x}}$$

- The odds of the event is then $\dfrac{p_w(y=1 \mid x)}{p_w(y=0 \mid x)} = e^{w^T \cdot x}$
- And the log-odds are

$$\log \frac{p_w(y=1 \mid x)}{p_w(y=0 \mid x)} = w^T \cdot x$$

Generalized Linear Models

- Generalized linear models
  - A linear model whose output is passed through non-linear function
- Decision boundary/surface
  - An n-1 dimensional hyper-plane that separates the data into two groups
  - The non-linearity gives a classification boundary
- We still have
  - Convex model
  - Hypothesis class: linear decision boundaries

Logistic Regression Decisions

- Given parameters w, how do we make predictions?

$$p_w(y=1 \mid x) = \frac{1}{1+e^{-w^T \cdot x}}$$

- If output > .5, predict 1, else predict 0
- In addition to prediction, we have confidence in prediction
  - Confidence is the probability of the prediction

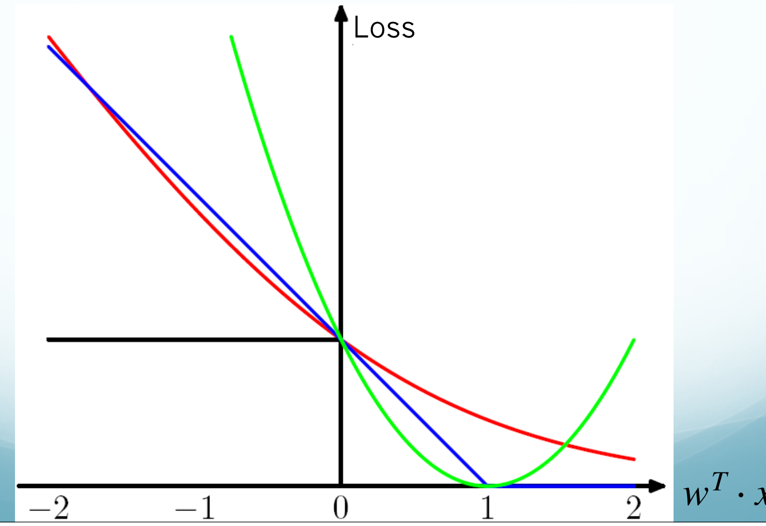
Logistic Regression

Fitting a function to data

- Fitting: Solve for w given y and x
- Function: Generalized linear function: logistic over regression
- Data: assume dependent variable linear combination of independent variables

Squared loss (green): we told liner regression to predict 1 for positive class
Logistic loss (red): loss is suffered whenever p(y=1|x) < 1

The loss suffered (y-axis) by predicting wx (x-axis) when positive is correct answer



Objective Function: Likelihood

- Conditional data likelihood

$$p(Y \mid X, w) = \prod_{i=1}^{n} p(y_i \mid x_i, w)$$

Conditional Log Likelihood

$$p(Y \mid X, w) = \prod_{i=1}^{n} p(y_i \mid x_i, w)$$

$$\ell(Y, X, w) = \log p(Y \mid X, w) = \sum_{i=1}^{n} \log p(y_i \mid x_i, w)$$

$$p\,(y=1 \mid x, w) = \frac{1}{1+e^{-w^T \cdot x}} \qquad p\,(y=0 \mid x, w) = \frac{e^{-w^T \cdot x}}{1+e^{-w^T \cdot x}}$$

Logistic Regression

Fitting a function to data

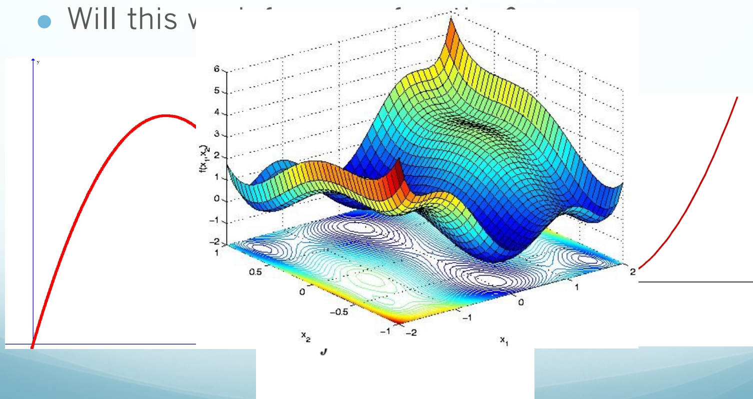
- **Fitting: Solve for w given y and x**
- Function: Generalized linear function: logistic over regression: conditional log likelihood
- Data: assume dependent variable linear combination of independent variables

Function Optimization

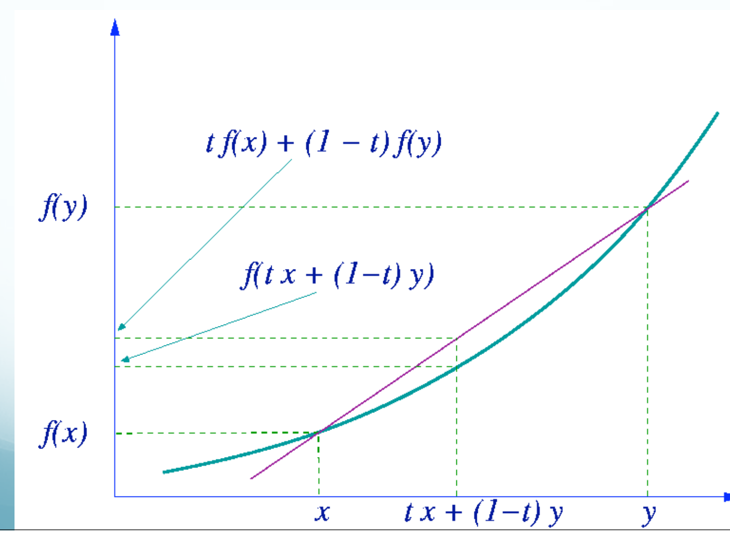
- We have a function and want to maximize/minimize it
- How do we find the point at which the function reaches its max/min?

Function Optimization

- Take the derivative, set it equal to 0, solve!
- Will this



Convex Functions



Maximum Likelihood Estimation

- MLE: Find the value at which the likelihood is maximized
  - We'll talk about other options later in the semester
- Given the conditional log likelihood
  - Take the derivatives for parameters w
  - Set each derivative to 0
  - M equations and M variables
  - Solve for w
- Problem
  - No closed form (analytical) solution for w

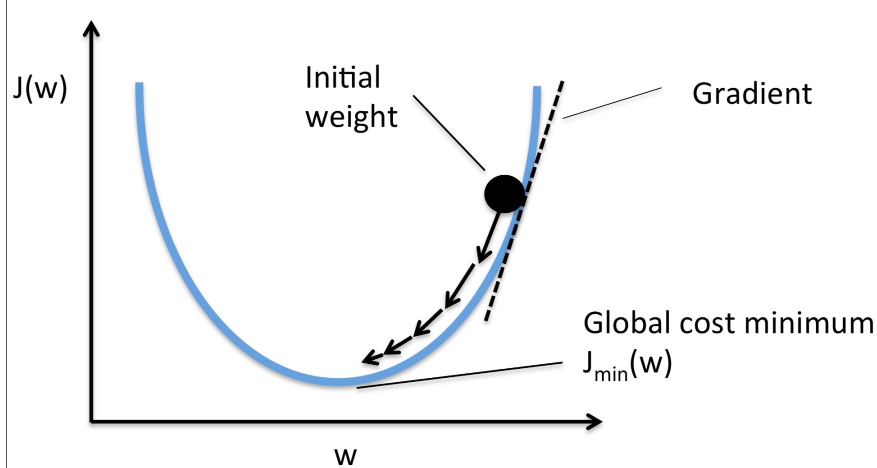
Convex Optimization

- The conditional maximum likelihood is concave
  - There is a single maximal solution
- We can maximize using convex optimization techniques
  - Its easy to optimize convex functions
  - There are **many** convex optimization algorithms

Gradient Descent

- First order method: needs first order derivatives
- Assuming F(x) is defined and differentiable, then F(x) decreases fastest if we go from x in the direction of the gradient of F
  - $-\nabla F(x)$- vector of partial derivatives of F
  - $x' = x - \gamma \nabla F(x)$ - Update
  - For sufficiently small values of γ, the value of the function will get smaller

Gradient Descent



Derivatives

Objective:
conditional log likelihood $\ell(Y,X,w) = \log p(Y \mid X,w) = \sum_{i=1}^{n} \log p(y_i \mid x_i, w)$

Given the sigmoid as $h_w(x) = \dfrac{1}{1+e^{-w^T \cdot x}}$

We can rewrite compactly $p(y \mid x) = (h_w(x))^y (1-h_w(x))^{1-y}$

New objective $\ell(Y,X,w) = \sum_{i=1}^{N} \log\{(h_w(x))^y (1-h_w(x))^{1-y}\}$

Derivative

$$\frac{\partial \ell(Y,X,w)}{\partial w} = \sum_{i=1}^{N} (y_i - h_w(x_i))x_i$$

- The derivative is 0 when $y_i = p(y_i \mid x_i, w)$
- Maximizing likelihood = minimize logistic error

Gradient Descent Solution

$$w^{(t+1)} = w^t + \gamma \frac{\partial \ell(Y,X,w)}{\partial w}$$

$$w^{t+1} = w^t + \gamma \sum_{i=1}^{N} (y_i - h_w(x_i))x_i$$

Algorithm: Logistic Regression

- Train: given data X and Y
  - Initialize w to starting value
  - Repeat until convergence
    - Compute the value of the derivative for X,Y and w
    - Update w by taking a gradient step
- Predict: given an example x
  - Using the learned w, compute p(y|x,w)

$$p(y=1 \mid x,w) = \frac{1}{1+e^{-w^T \cdot x}}$$

- Note: many other optimization routines available

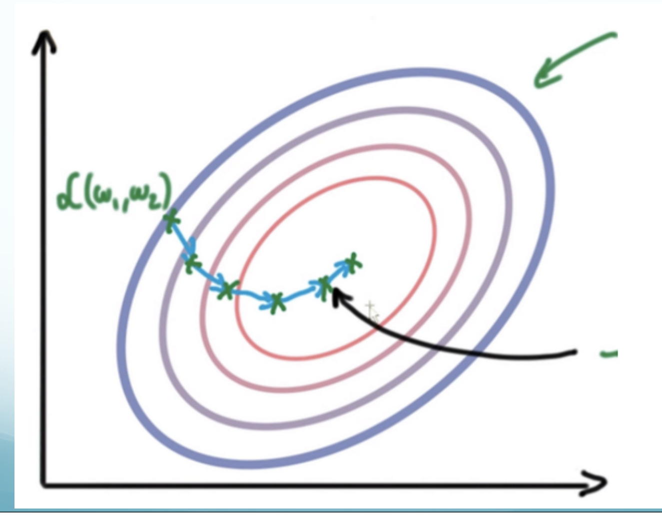
Gradient Based Optimization

- Multiple methods available for optimizing the same objective function
  - First order methods
  - Second order methods
  - Adaptive methods
  - …

Alternate Methods

- Batch gradient descent
  - Utilize the gradient of all the data
  - Slow: need to consider all the data before making a single update

Gradient Descent

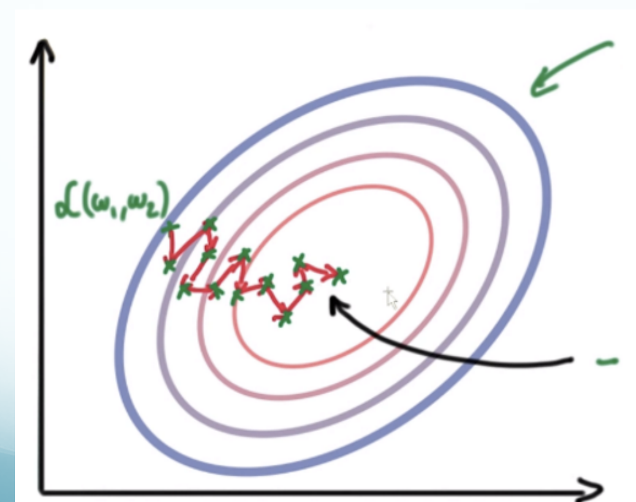


Stochastic Updates

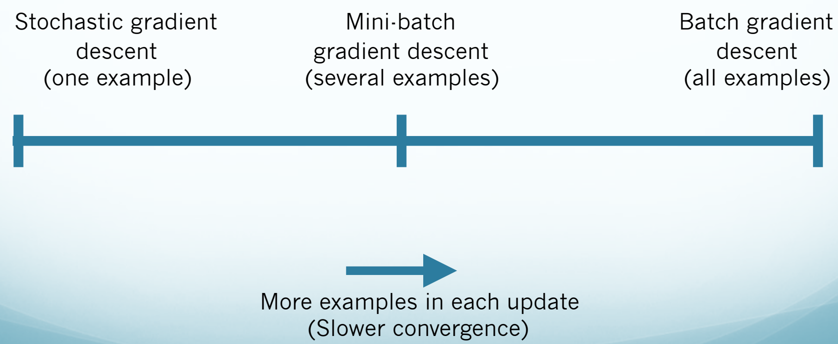
- Compute the gradient on a single example at a time

$$w^{t+1} = w^t + \gamma(y_i - h_w(x_i))x_i$$

Gradient Descent



Update Frequency



Regularization

- Same over-fitting problems as least squares
- Add regularization term to objective to favor different considerations
- Similar options
  - Quadratic regularization (L2)
  - L1 regularization (sparse solutions)
  - For each regularization optimize new objective function

Summary

- Logistic regression
  - Learn p(y|x) directly with functional form of distribution
  - Maximize the data conditional log-likelihood
  - Equivalent to linear prediction
    - Decision rule is a hyper-plane
  - Regularization to prevent over-fitting