From Regression to Classification with Logistic Regression

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

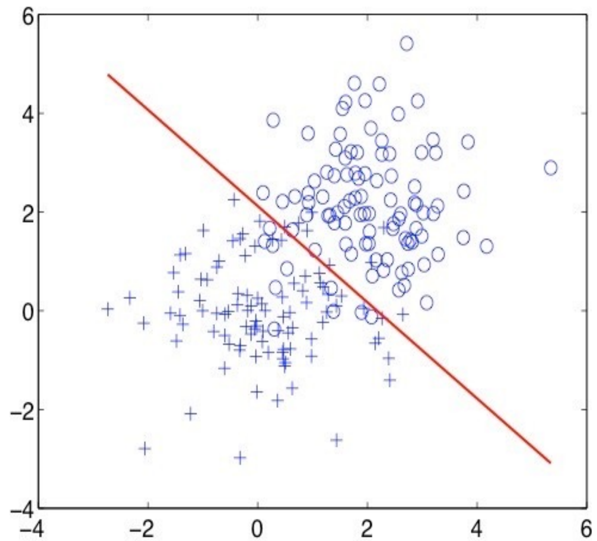
- Least squares regression
  - Outputs real number for each example
- It seems that classification should be easier!
- Let's use regression for classification
- Learn least squares regression model $f_w(x) = y$

$$f_w(x) = w^T \cdot x$$

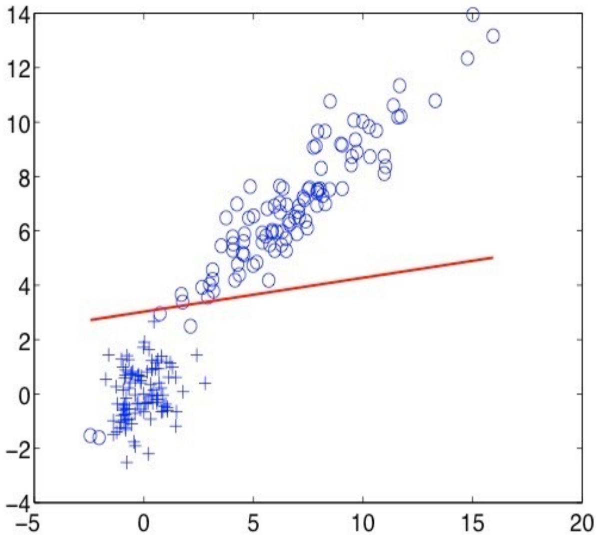
- If y>0, predict "True (1)"
- If y≤0 predict "False (0)"

Regression for Classification

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



Good



Bad

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

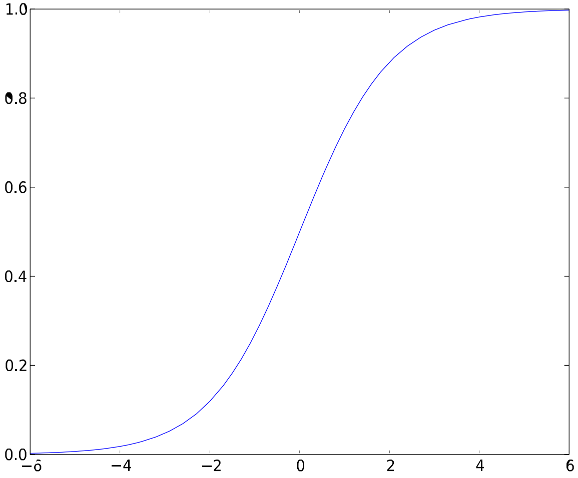
Machine Learning

- **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

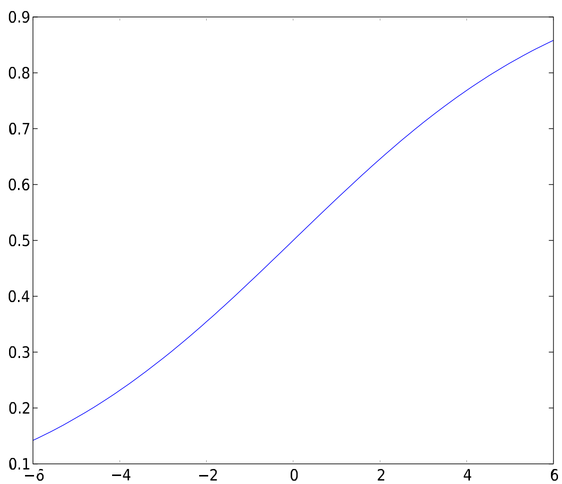
- Quick fix: apply a function to the output of regression that gives desired valued
- Logistic function
  - $g_\alpha(x) = \frac{1}{1+e^{-\alpha x}}$
  - Outputs between 0 and 1
  - Scaling parameter α
  - Most outputs are close to 1 or 0



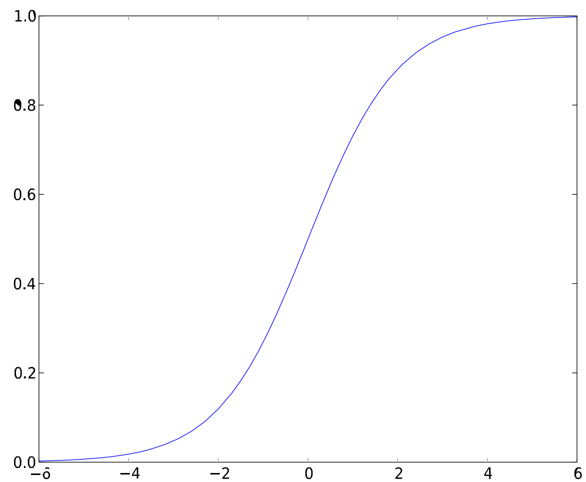
$\alpha = 1$

Logistic Function

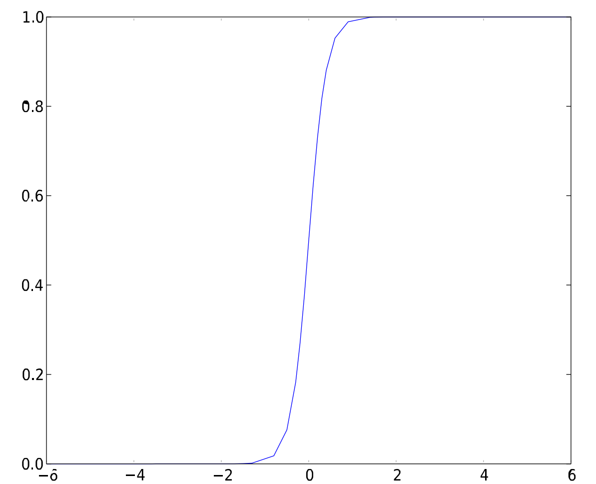
$$g_\alpha(x) = \frac{1}{1 + e^{-\alpha x}}$$



$\alpha = 0.3$



$\alpha = 1$



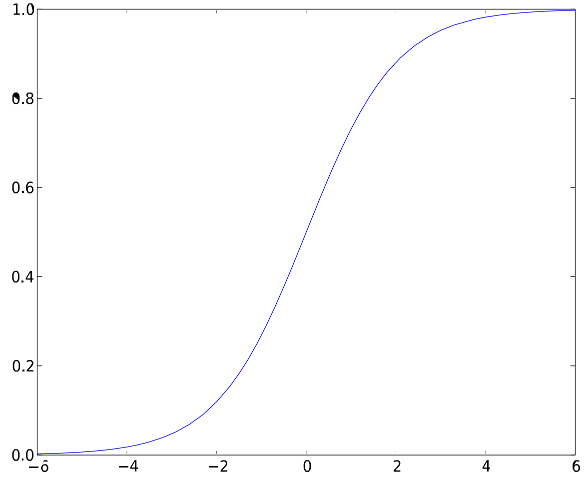
$\alpha = 5$

Logistic Regression

- We can combine the logistic function and our regression model

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

- Notice: as
  - Large- output closer to 1
  - Small- output closer to 0



Probabilistic View

- In regression we modeled probability of the output
- Probability of the example and classification?
  - p(x,y)?
  - p(x,y) = p(x|y) p(y)
  - Since we know x, we want to maximize y
  - p(x|y)p(y) = p(y|x)p(x)
  - Since p(x) is fixed:

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

Why?

- We can now write the distribution as

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

- Which implies that

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

- The odds of the event is then

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

- And the log-odds are

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

Generalized Linear Models

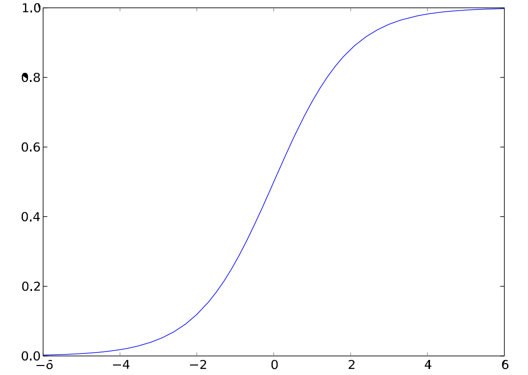
- Decision boundary/surface
  - An n-1 dimensional hyper-plane that separates the data into two groups
  - These are linear functions of x, even though logistic is not linear
- Generalized linear models
  - A linear model whose output is passed through non-linear function
- Hypothesis class
  - Linear decision boundaries

Logistic Regression Decisions

- Given parameters w, how do we make predictions?

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

- 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

Objective Function: Likelihood

- Conditional data likelihood

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

Conditional Log Likelihood

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

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

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

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

Logistic Regression

- **Fitting a function to data**

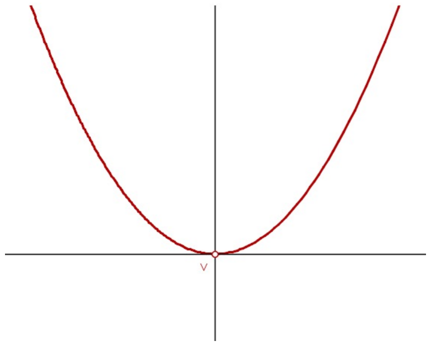
- **Fitting: Solve for w given y and x**
- Function:
  - Generalized linear function: logistic over regression: conditional likelihood
- Data: assume dependent variable linear combination of independent variables

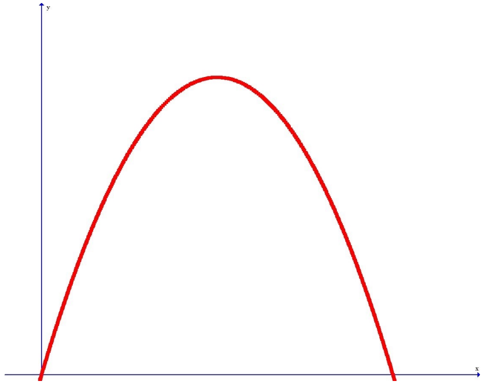
Function Optimization

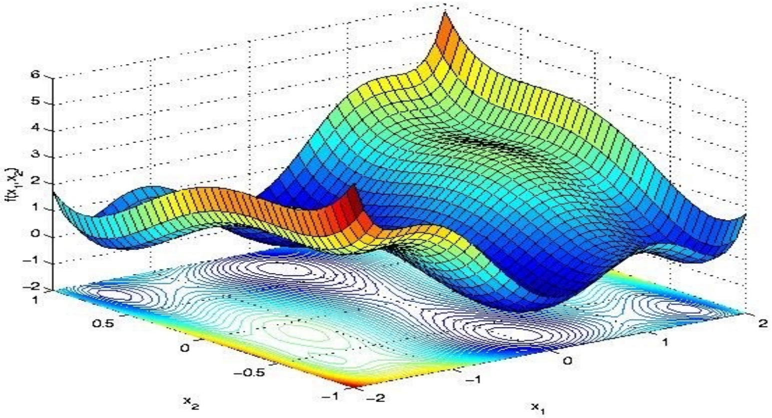
- We have a function and want to maximize/minimize it
- How do we find the point at which the function reaches its max/min?

Function Optimization

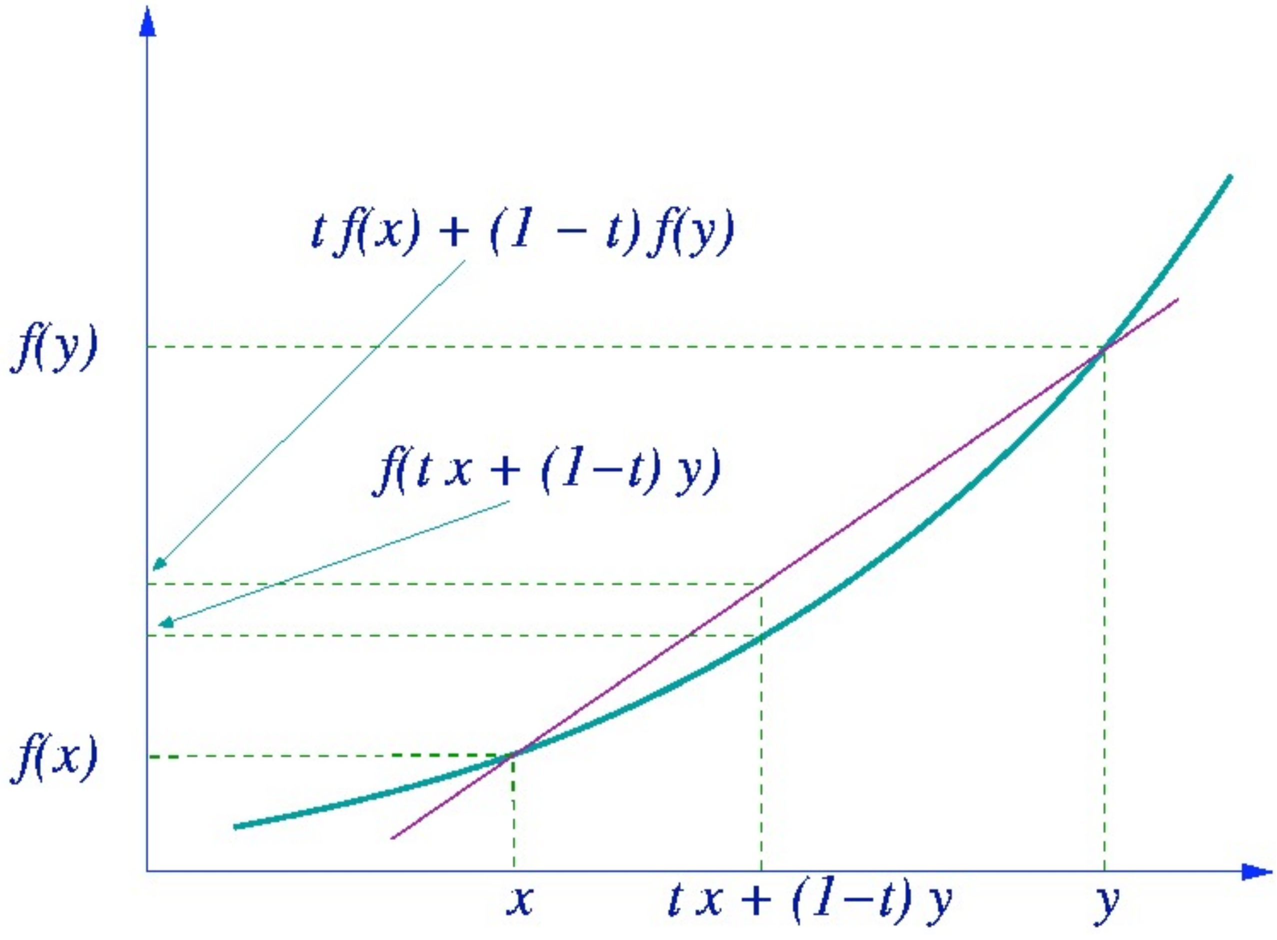
- Take the derivative, set it equal to 0, solve!
- Will this work for every function?







Convex Functions



Maximum Likelihood Estimation

- MLE: Find the value at which the likelihood is maximized
- 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

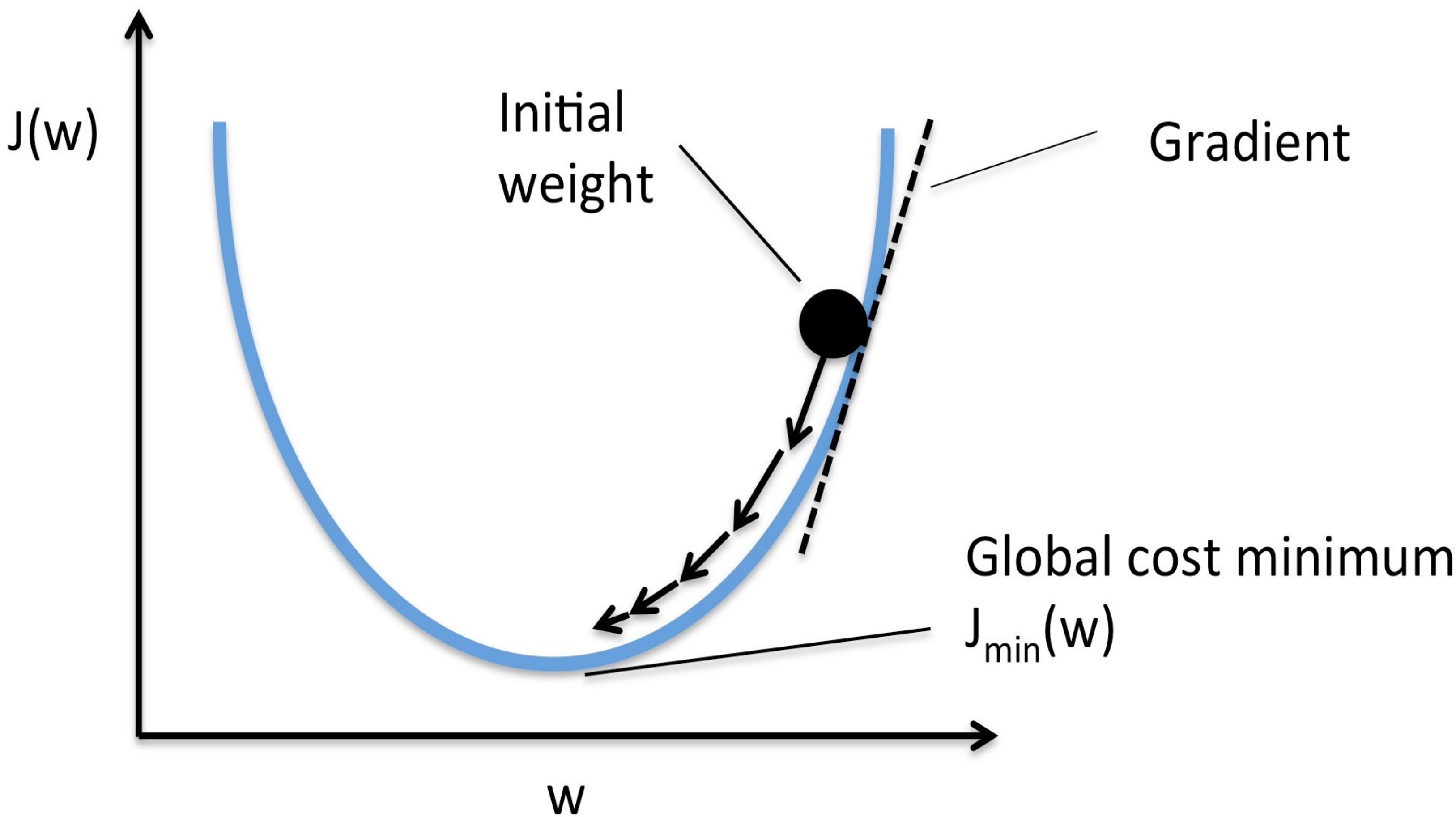
Convex Optimization

- 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

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

- The derivative is 0 when y_i = $p(y_i=1|xi,w)$
  - Minimize the prediction error

Gradient Descent Solution

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

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

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

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)

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

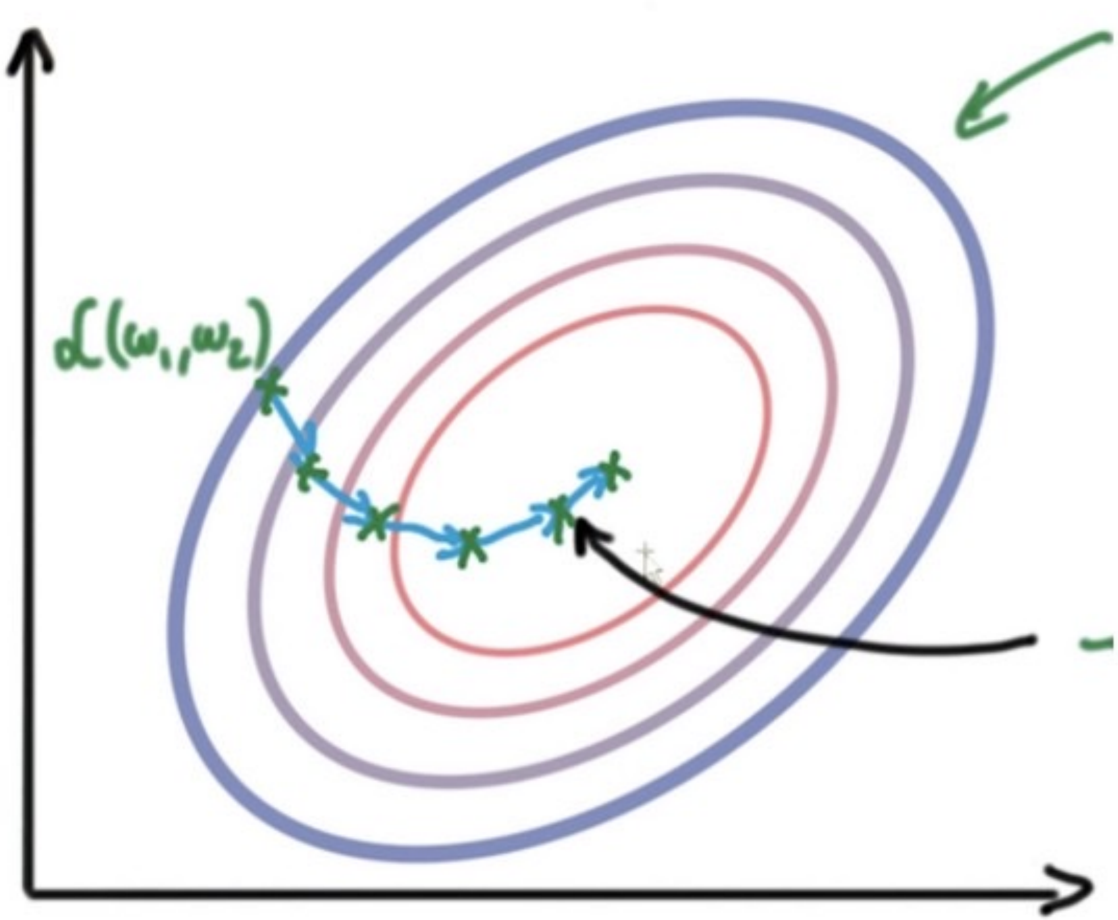
Gradient Based Optimization

- 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



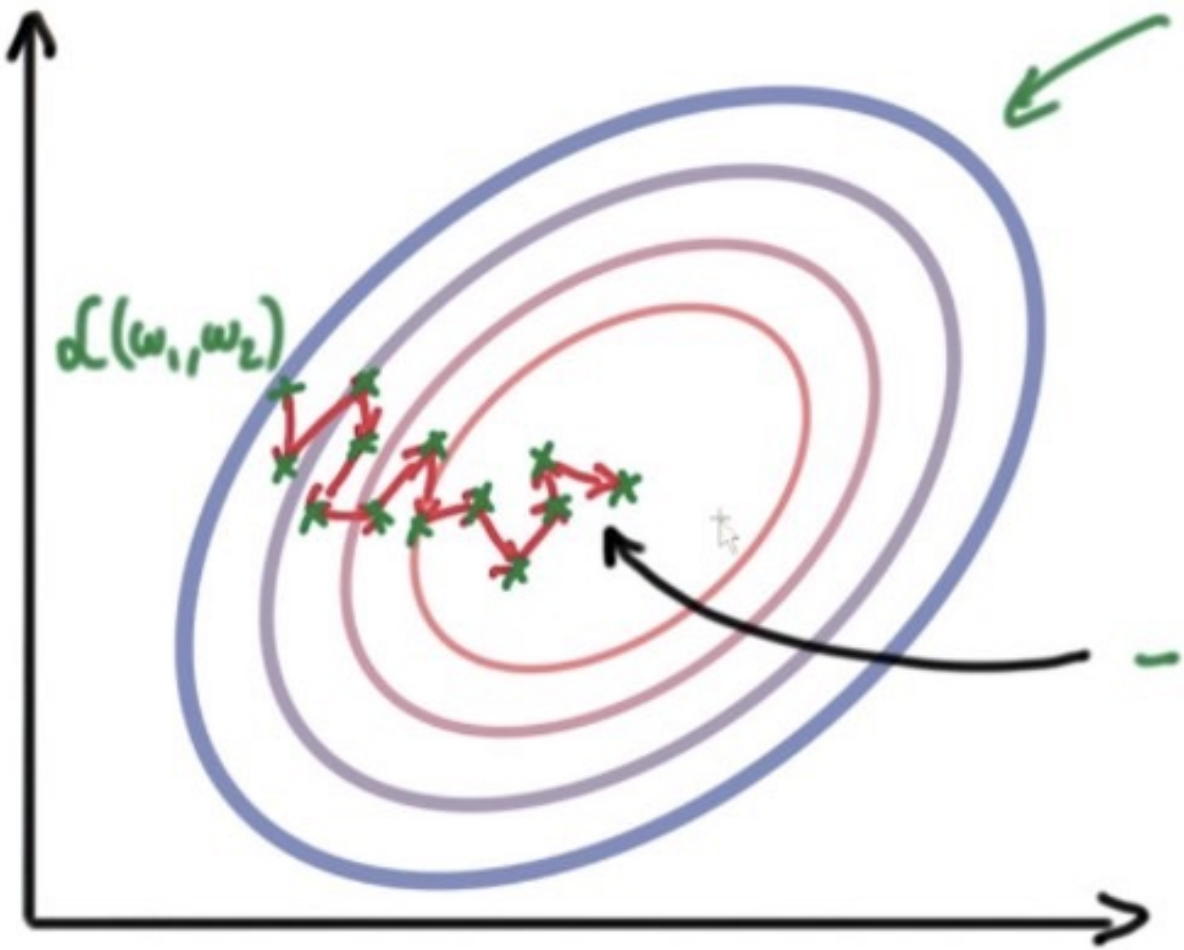
Stochastic Updates

- Compute the gradient on a single example at a time
- $w^{t+1} = w^t + \gamma \sum_{i=1}^{N} (y_i - p(y_i = 1 | x_i, w)) x_i$
- Instead of
- $w^{t+1} = w^t + \gamma (y_1 - p(y_1 = 1 | x_1, w)) x_1$

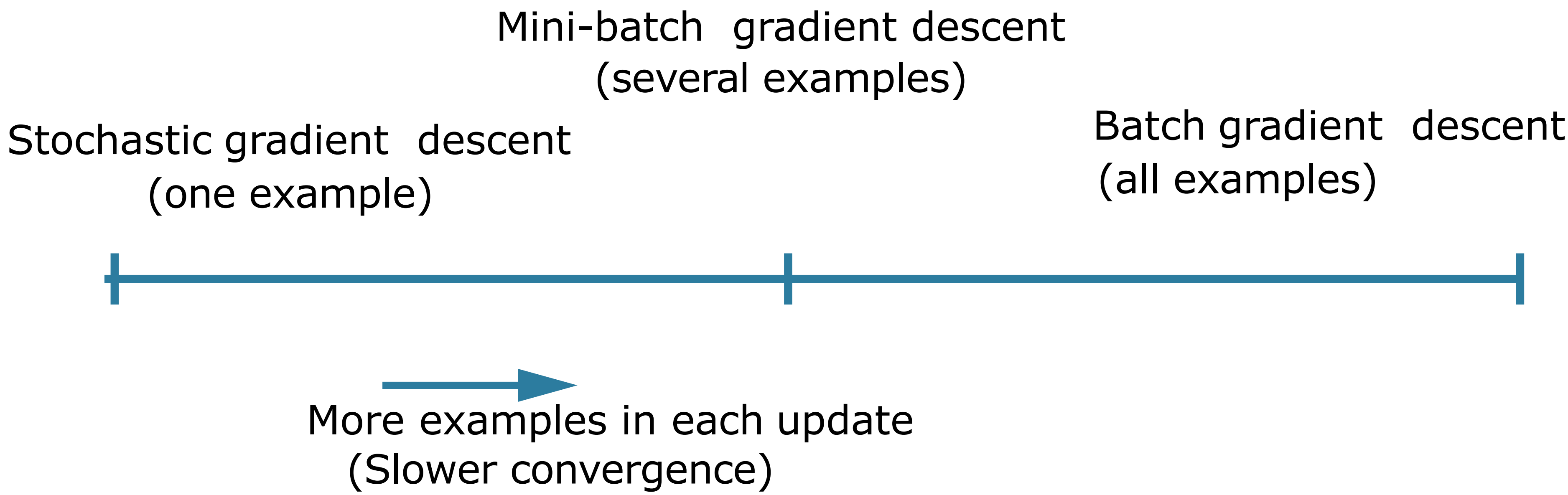
$$+ \gamma (y_2 - p(y_2 = 1 | x_2, w)) x_2$$

$$+ \gamma (y_3 - p(y_3 = 1 | x_3, w)) x_3 + \cdots$$

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