Caltech

Machine Learning & Data Mining

CS/CNS/EE 155

Lecture 2:

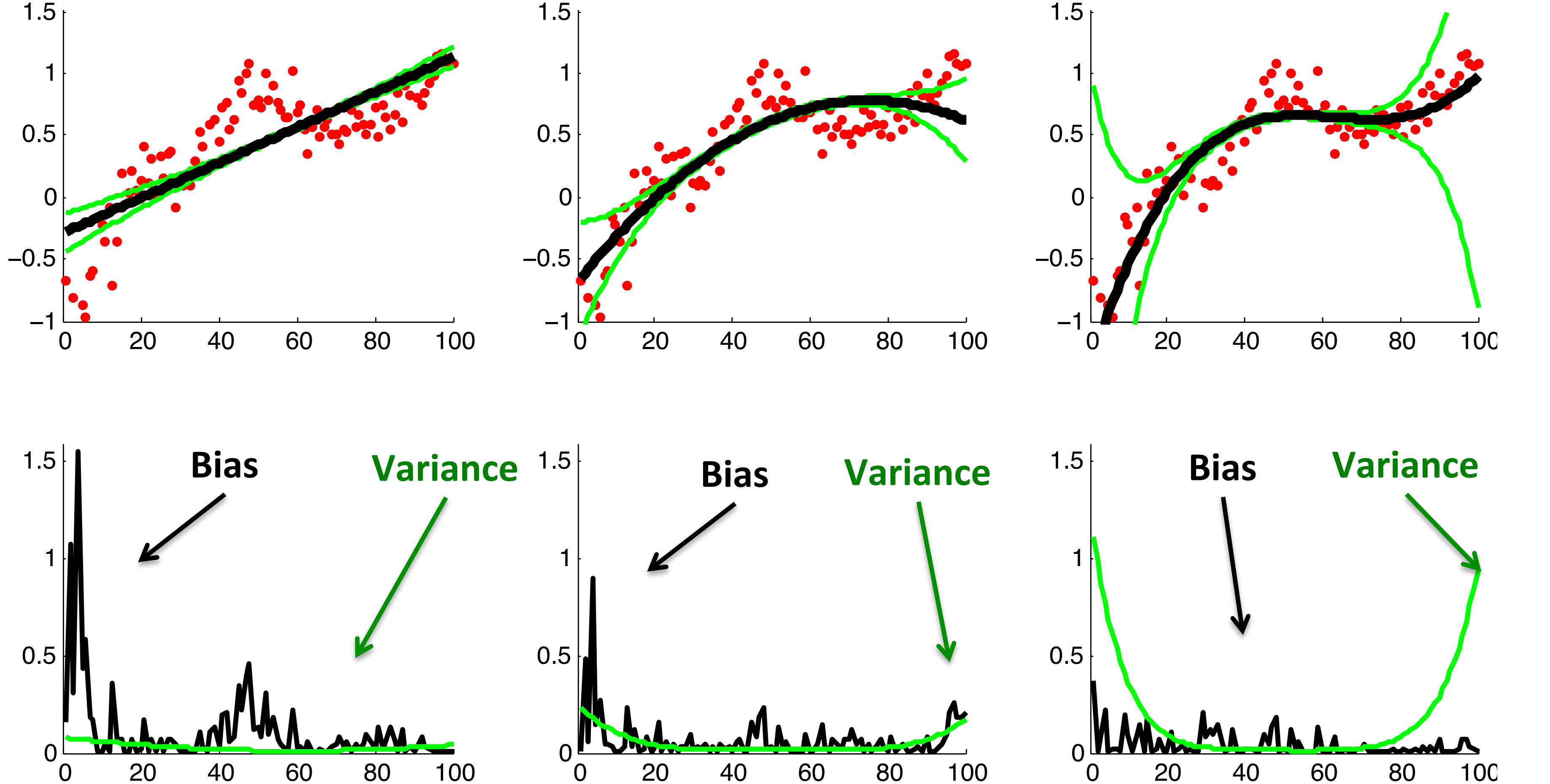
Review Part 2

Recap: Basic Recipe

- Training Data: $S = \{(x_i, y_i)\}_{i=1}^{N}$ $\quad x \in R^D$, $y \in \{-1,+1\}$
- Model Class: $f(x \mid w,b) = w^T x - b$ **Linear Models**
- Loss Function: $L(a,b) = (a-b)^2$ **Squared Loss**
- Learning Objective: $\underset{w,b}{\operatorname{argmin}} \sum_{i=1}^{N} L\big(y_i, f(x_i \mid w,b)\big)$

Optimization Problem

Recap: Bias-Variance Trade-off



Recap: Complete Pipeline

$$S = \{(x_i, y_i)\}_{i=1}^{N}$$

Training Data

$$f(x \mid w, b) = w^T x - b$$

Model Class(es)

$$L(a,b) = (a-b)^2$$

Loss Function

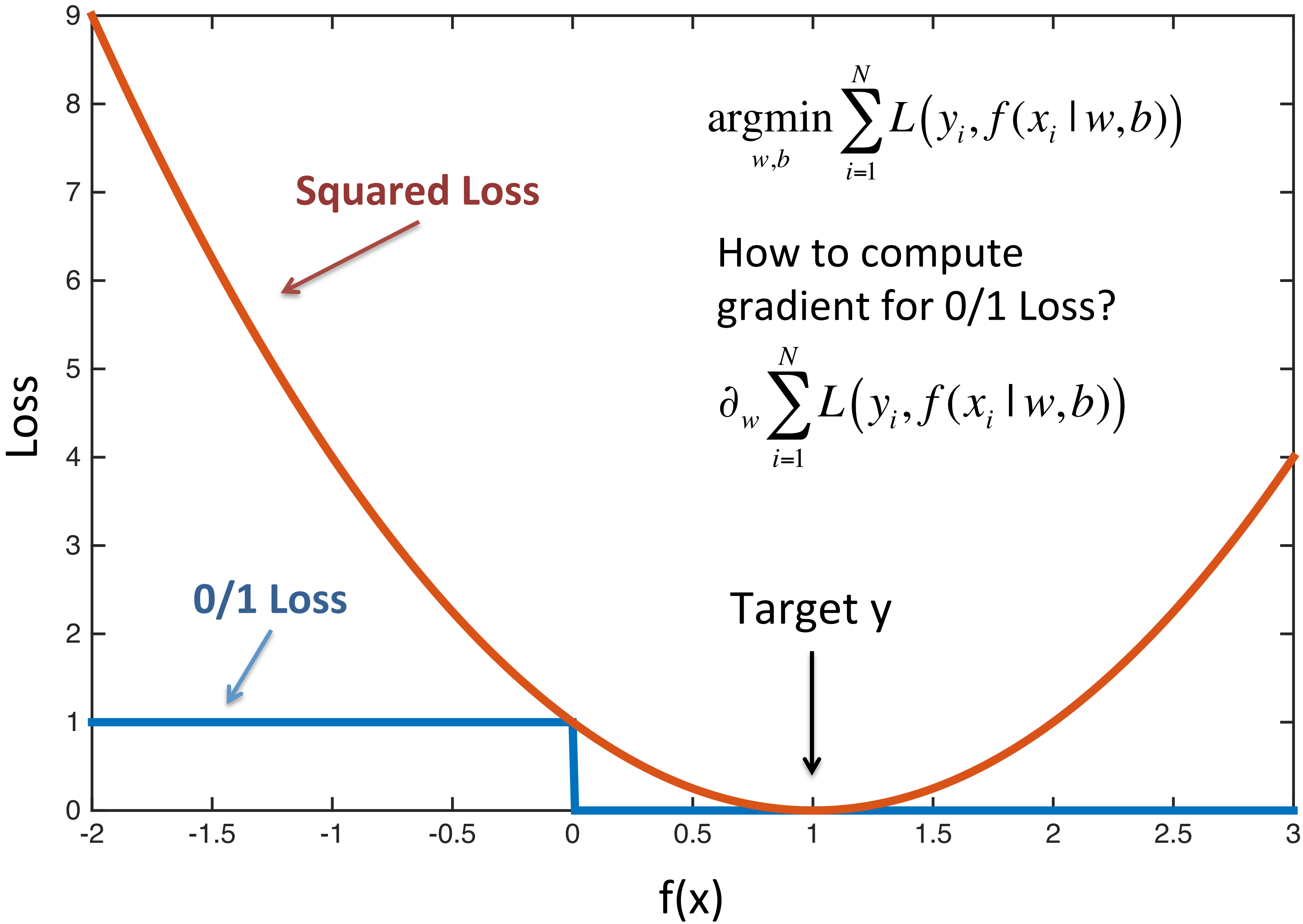
$$\underset{w,b}{\operatorname{argmin}} \sum_{i=1}^{N} L\big(y_i, f(x_i \mid w, b)\big)$$

Cross Validation & Model Selection

Profit!

Today

- **Beyond Linear Basic Linear Models**
  - **Support Vector Machines**
  - **Logistic Regression**
  - **Feed-forward Neural Networks**
  - **Different ways to interpret models**
- Different Evaluation Metrics
- Hypothesis Testing



$\underset{w,b}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, f(x_i \mid w, b)\right)$
How to compute gradient for 0/1 Loss?
$\partial_w \sum_{i=1}^{N} L\left(y_i, f(x_i \mid w, b)\right)$
Squared Loss
0/1 Loss
Target y
Loss
f(x)
9
8
7
6
5
4
3
2
1
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

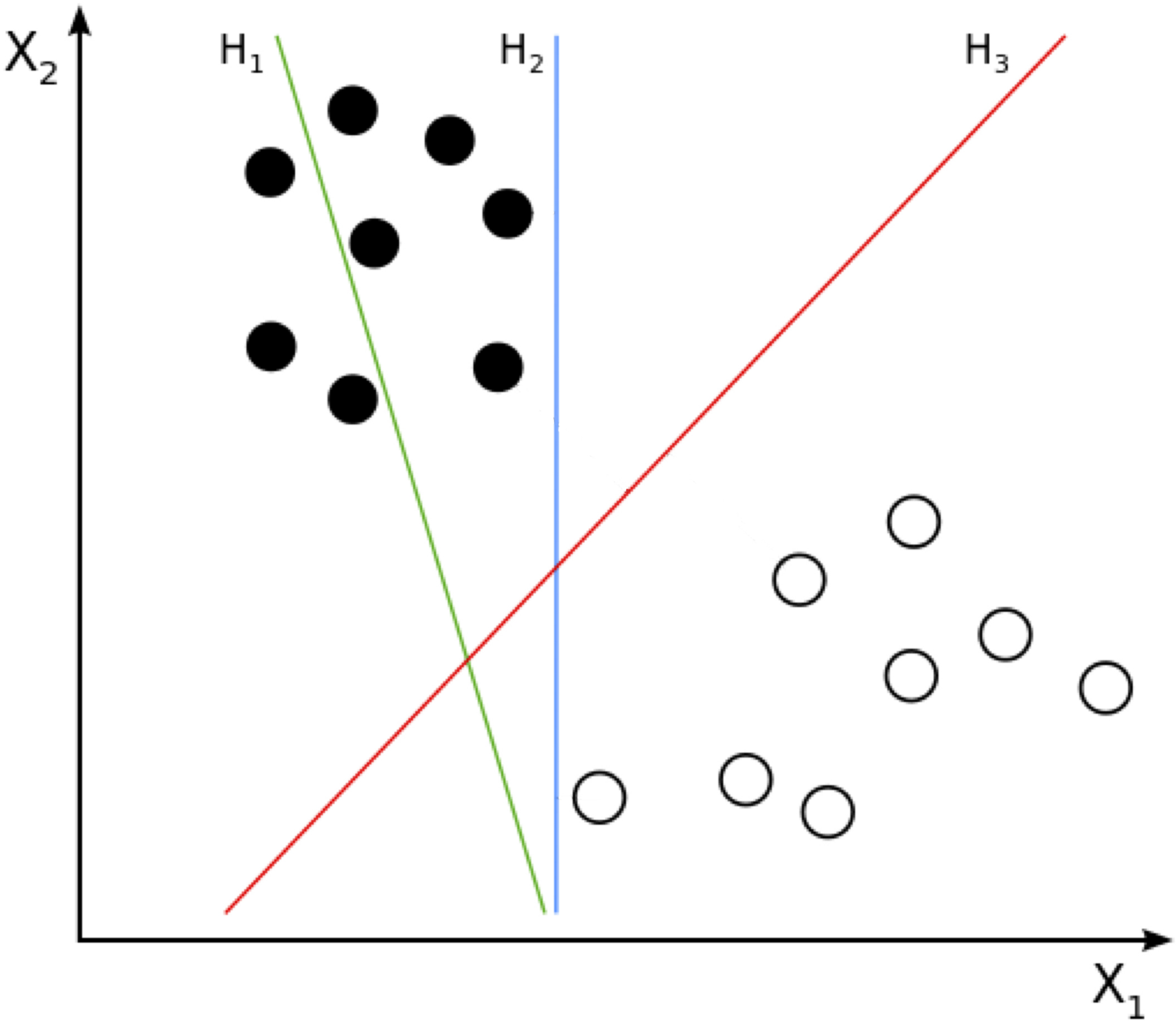
Recap: 0/1 Loss is Intractable

- 0/1 Loss is flat or discontinuous everywhere
- VERY difficult to optimize using gradient descent
- **Solution:** Optimize smooth surrogate Loss
  - **Today: Hinge Loss (…eventually)**

Support Vector Machines

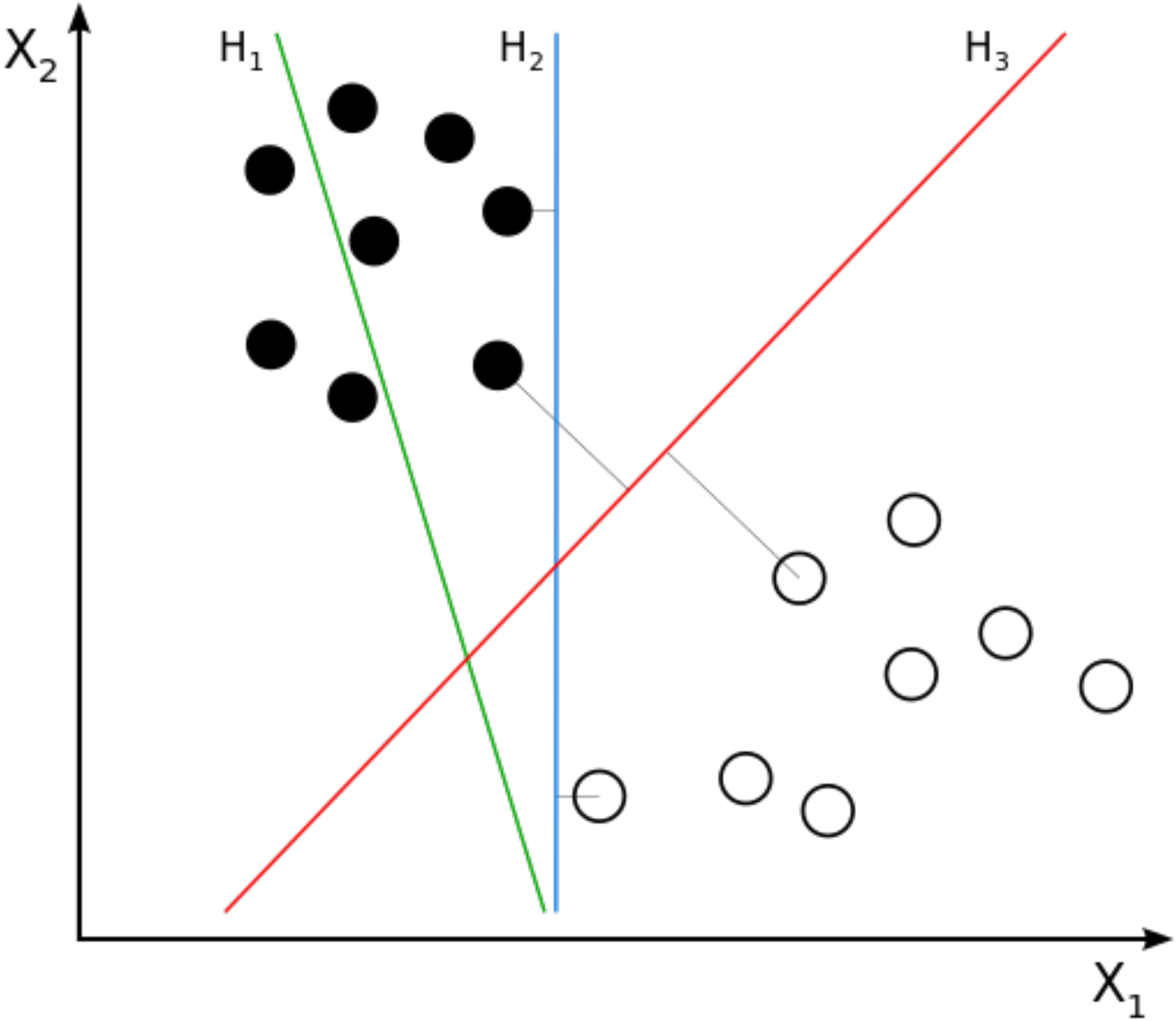
aka Max-Margin Classifiers

Which Line is the Best Classifier?



Source: http://en.wikipedia.org/wiki/Support_vector_machine

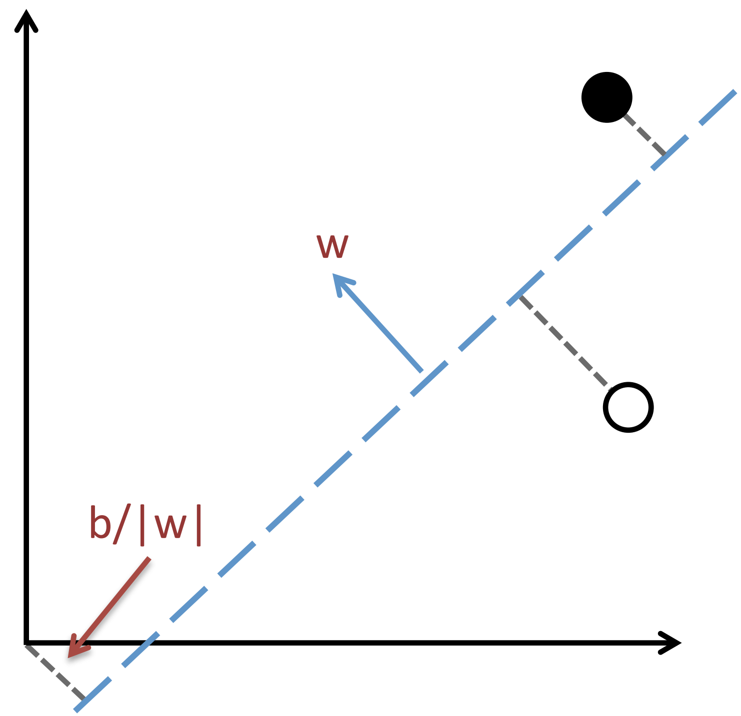
Which Line is the Best Classifier?



Source: http://en.wikipedia.org/wiki/Support_vector_machine

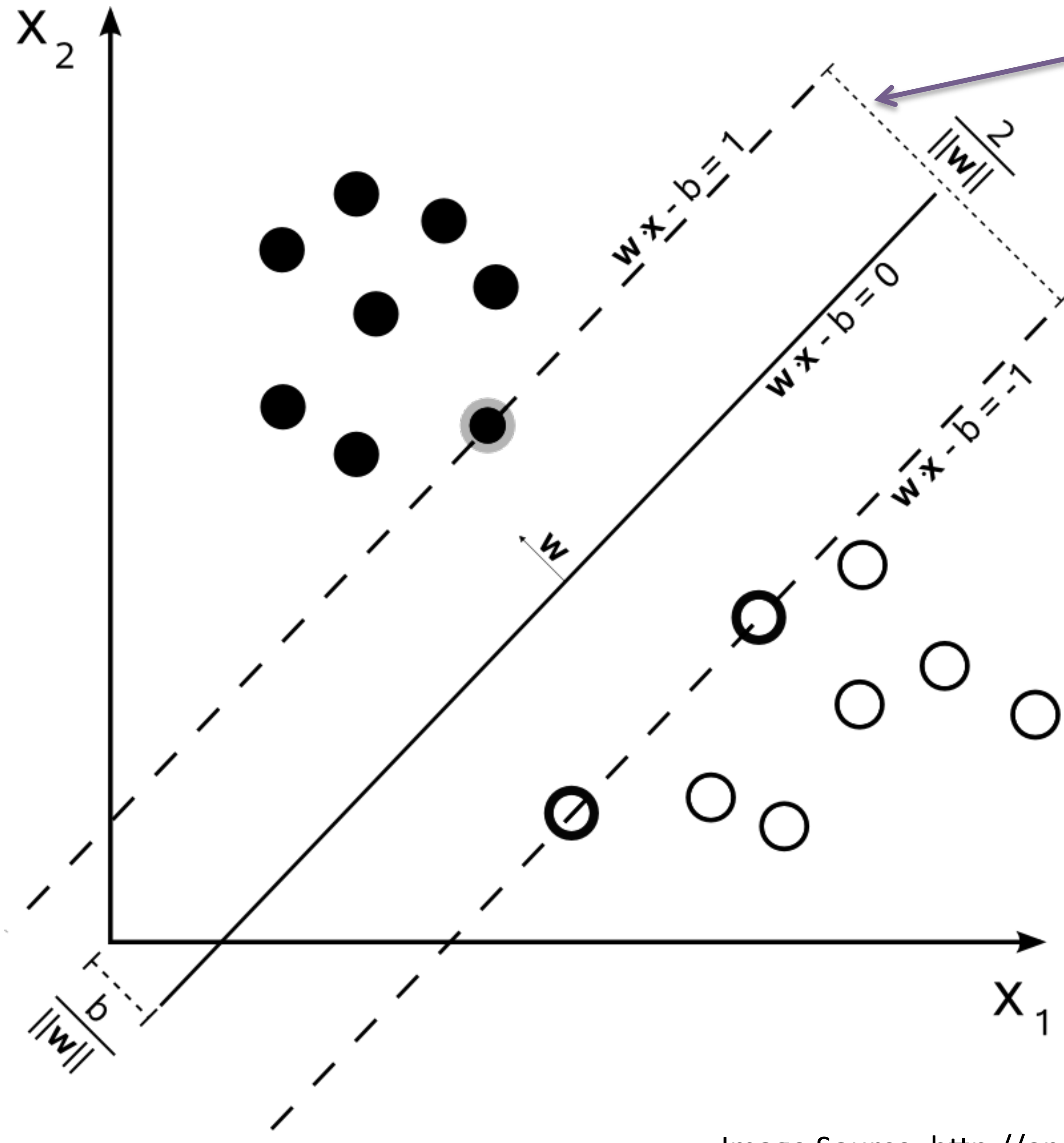
Hyperplane Distance

- Line is a 1D, Plane is 2D
- Hyperplane is many D
  - Includes Line and Plane
- Defined by (w,b)
- Distance: $$\frac{|w^T x - b|}{\|w\|}$$
- Signed Distance: $$\frac{w^T x - b}{\|w\|}$$



Linear Model = un-normalized signed distance!

Max Margin Classifier (Support Vector Machine)



"Margin"

$$\underset{w,b}{\operatorname{argmin}} \frac{1}{2} w^T w \equiv \frac{1}{2} \|w\|^2$$

$$\forall i : y_i \left(w^T x_i - b \right) \geq 1$$

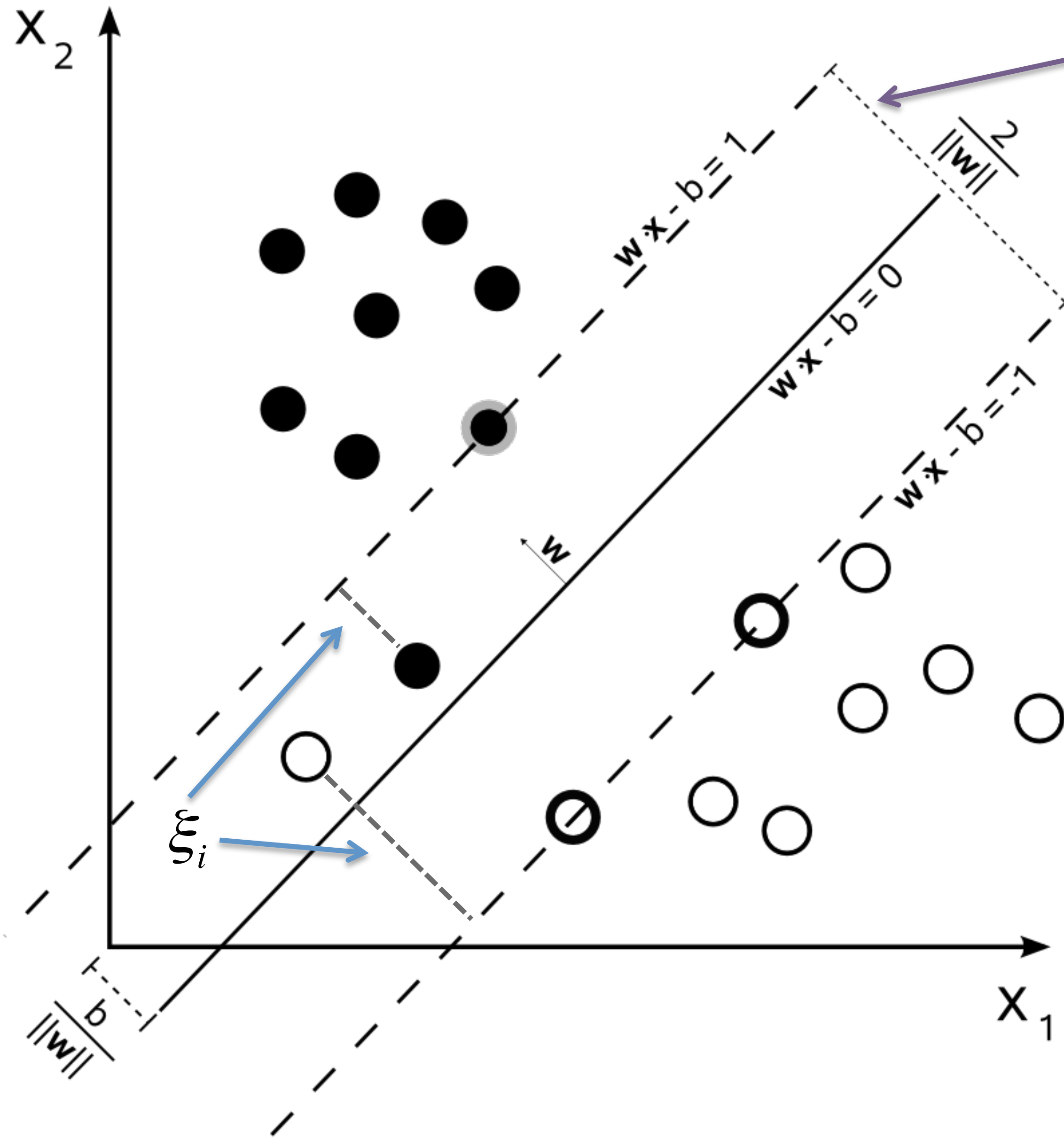
Better generalization
to unseen test examples
(beyond scope of course*)

"Linearly Separable"

*http://olivier.chapelle.cc/pub/span_lmc.pdf

Image Source: http://en.wikipedia.org/wiki/Support_vector_machine

Soft-Margin Support Vector Machine



$$\underset{w,b,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

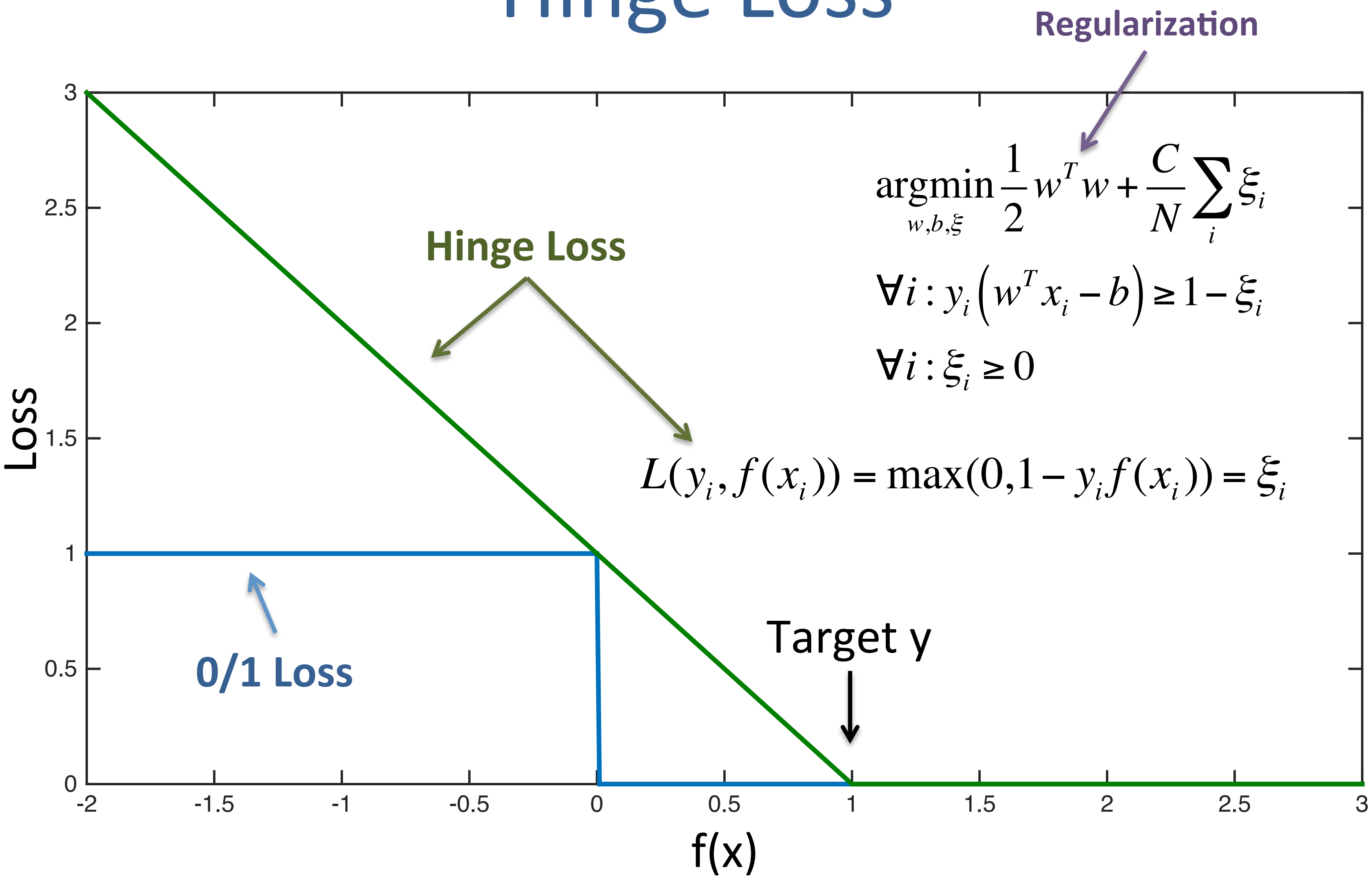
$$\forall i : y_i \left(w^T x_i - b \right) \geq 1 - \xi_i$$

$$\forall i : \xi_i \geq 0$$

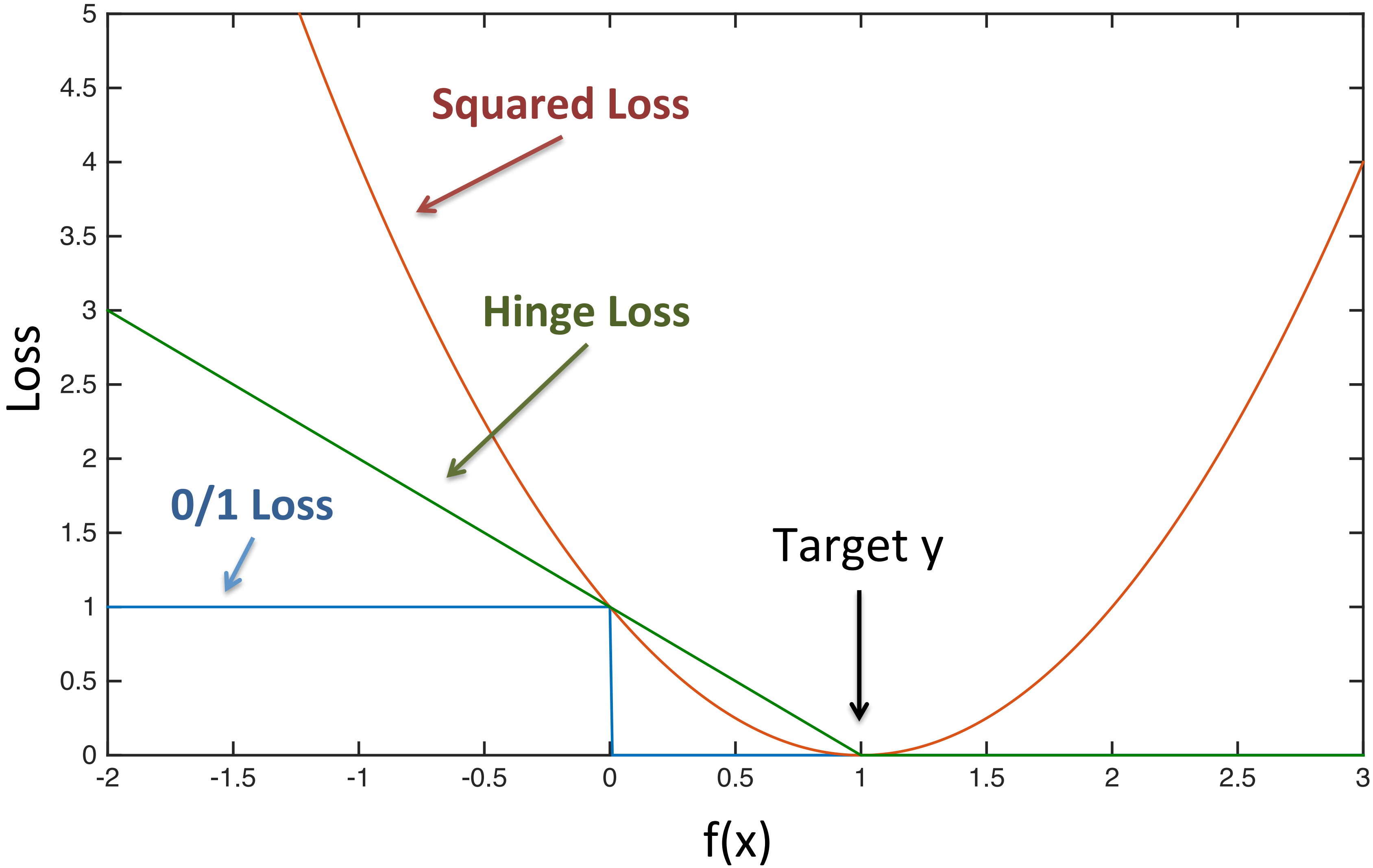
Size of Margin
vs
Size of Margin Violations
(C controls trade-off)

Image Source: http://en.wikipedia.org/wiki/Support_vector_machine

Hinge Loss


Regularization
$\underset{w,b,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$
$\forall i : y_i \left(w^T x_i - b \right) \geq 1 - \xi_i$
$\forall i : \xi_i \geq 0$
Hinge Loss
$L(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) = \xi_i$
0/1 Loss
Target y
Loss
f(x)
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

Hinge Loss vs Squared Loss



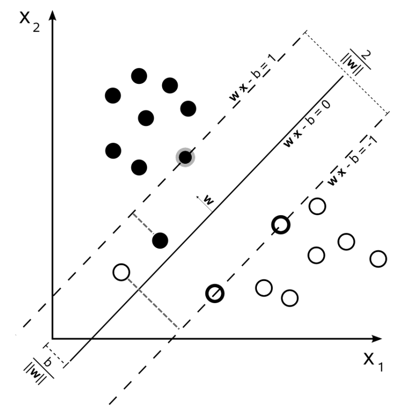
Support Vector Machine

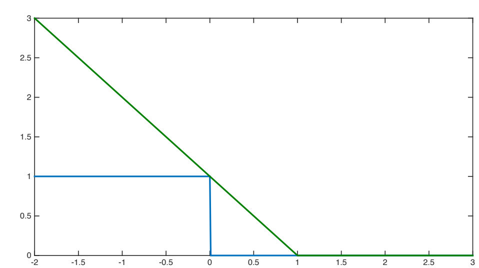
- 2 Interpretations
- Geometric
  - Margin vs Margin Violations
- Loss Minimization
  - Model complexity vs Hinge Loss
- **Equivalent!**

$$\underset{w,b,\xi}{\operatorname{argmin}} \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i$$

$$\forall i : y_i \left(w^T x_i - b \right) \geq 1 - \xi_i$$

$$\forall i : \xi_i \geq 0$$





Logistic Regression

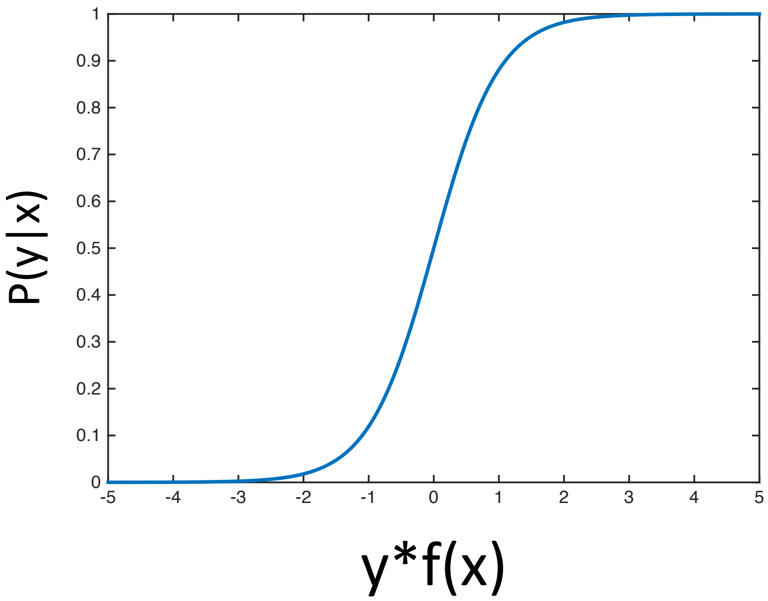
aka “Log-Linear” Models

Logistic Regression

$$P(y\,|\,x,w,b)=\frac{e^{y\left(w^T x-b\right)}}{e^{y\left(w^T x-b\right)}+e^{-y\left(w^T x-b\right)}}$$

$$P(y\,|\,x,w,b)\propto e^{y\left(w^T x-b\right)}\equiv e^{y*f(x|w,b)}$$

"Log-Linear" Model



Also known as sigmoid function: $\sigma(a)=\frac{e^a}{1+e^a}$

Maximum Likelihood Training

- Training set: $S = \{(x_i, y_i)\}_{i=1}^{N}$ $\quad x \in R^D$ $\quad y \in \{-1,+1\}$

- Maximum Likelihood: $\underset{w,b}{\operatorname{argmax}} \prod_i P(y_i \mid x_i, w, b)$
  - **(Why?)**

- Each (x,y) in S sampled independently!
  - See recitation next Wednesday!

Why Use Logistic Regression?

- SVMs often better at classification
  - At least if there is a margin…
- Calibrated Probabilities?
- Increase in SVM score….
  - ...similar increase in P(y=+1|x)?
  - **Not well calibrated!**
- **Logistic Regression!**

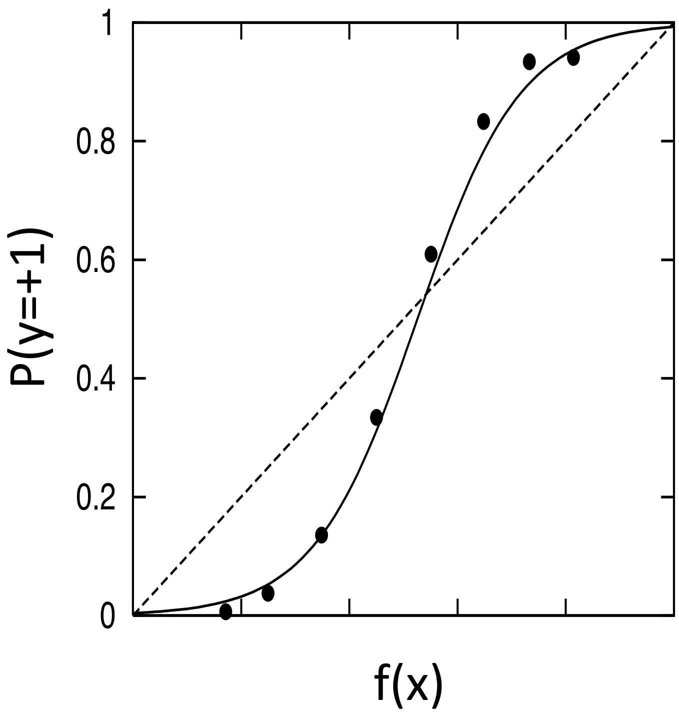


*Figure above is for
Boosted Decision Trees
(SVMs have similar effect)

Image Source: http://machinelearning.org/proceedings/icml2005/papers/079_GoodProbabilities_NiculescuMizilCaruana.pdf

Log Loss

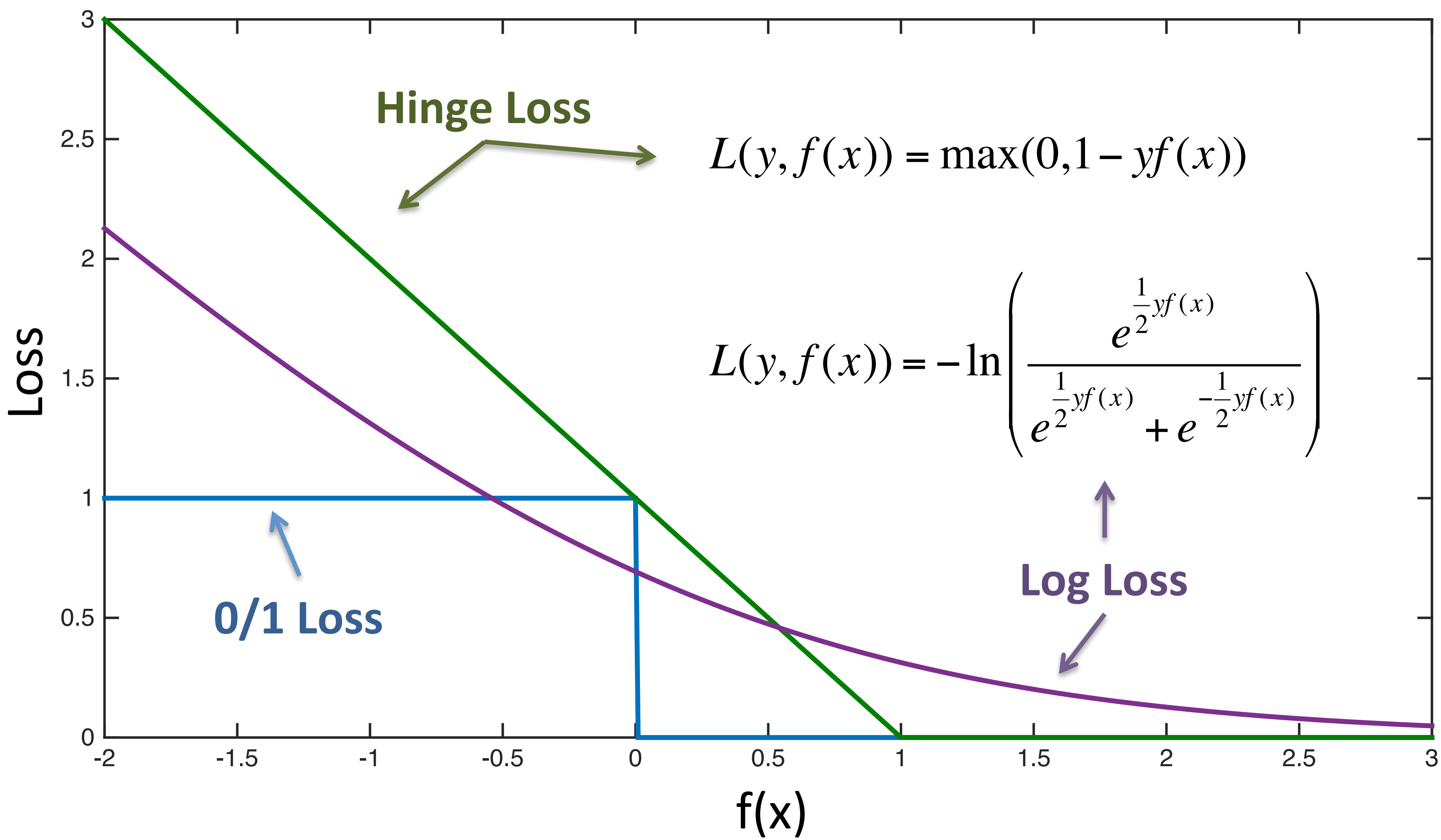
$$P(y \mid x,w,b) = \frac{e^{\frac{1}{2}y\left(w^T x - b\right)}}{e^{\frac{1}{2}y\left(w^T x - b\right)} + e^{-\frac{1}{2}y\left(w^T x - b\right)}} = \frac{e^{\frac{1}{2}yf(x|w,b)}}{e^{\frac{1}{2}yf(x|w,b)} + e^{-\frac{1}{2}yf(x|w,b)}}$$

$$\underset{w,b}{\operatorname{argmax}} \prod_i P(y_i \mid x_i,w,b) = \underset{w,b}{\operatorname{argmin}} \sum_i \underbrace{-\ln P(y_i \mid x_i,w,b)}_{\text{Log Loss}}$$

$$L(y,f(x)) = -\ln\left(\frac{e^{\frac{1}{2}yf(x)}}{e^{\frac{1}{2}yf(x)} + e^{-\frac{1}{2}yf(x)}}\right)$$

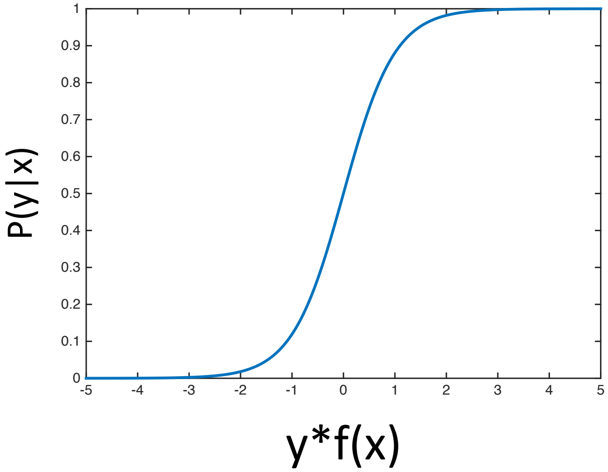
Solve using
Gradient Descent

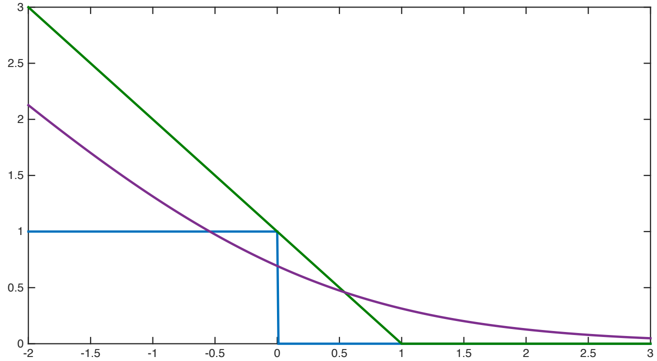
Log Loss vs Hinge Loss



Logistic Regression

- Two Interpretations
- Maximizing Likelihood
- Minimizing Log Loss
- **Equivalent!**



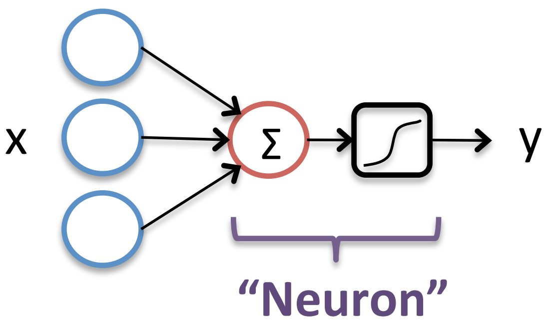


Feed-Forward Neural Networks

aka Not Quite Deep Learning

1 Layer Neural Network

- 1 Neuron
  - Takes input x
  - Outputs y

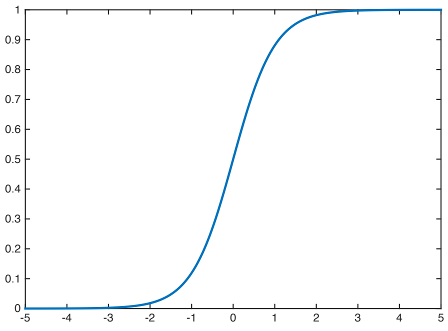


$f(x|w,b) = w^T x - b$

$= w_1 * x_1 + w_2 * x_2 + w_3 * x_3 - b$

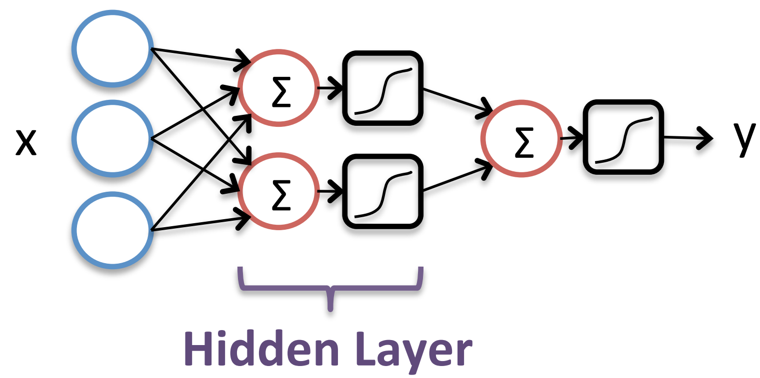
→ $y = \sigma(\ f(x)\ )$

- **~Logistic Regression!**
  - Gradient Descent



sigmoid
tanh
rectilinear

2 Layer Neural Network

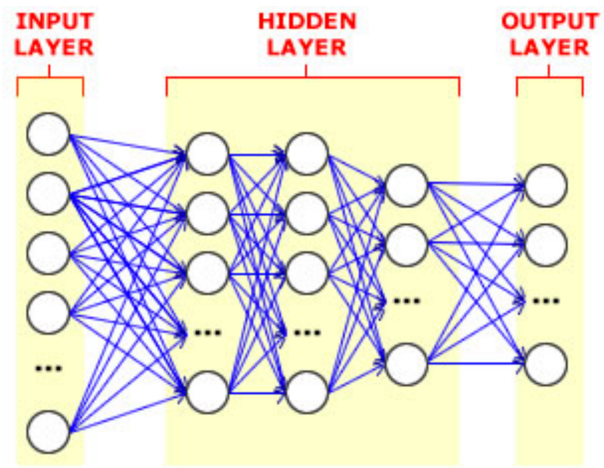


- 2 Layers of Neurons
  - 1st Layer takes input x
  - 2nd Layer takes output of 1st layer

Non-Linear!

- Can approximate arbitrary functions
  - Provided hidden layer is large enough
  - "fat" 2-Layer Network

Aside: Deep Neural Networks

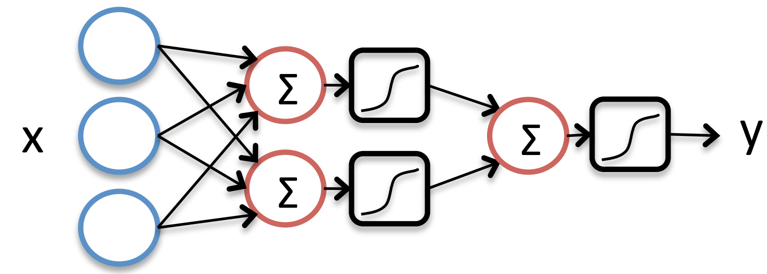


- Why prefer Deep over a "Fat" 2-Layer?
  - Compact Model
    - (exponentially large "fat" model)
  - Easier to train?

Image Source: http://blog.peltarion.com/2014/06/22/deep-learning-and-deep-neural-networks-in-synapse/

Training Neural Networks

- Gradient Descent!
  – Even for Deep Networks*
- Parameters:
  – $(w_{11},b_{11},w_{12},b_{12},w_2,b_2)$



$f(x|w,b) = w^T x - b \quad y = \sigma(\, f(x)\,)$

$$\partial_{w_2} \sum_{i=1}^{N} L(y_i,\sigma_2) = \sum_{i=1}^{N} \partial_{w_2} L(y_i,\sigma_2) = \sum_{i=1}^{N} \partial_{\sigma_2} L(y_i,\sigma_2) \partial_{w_2}\sigma_2 = \sum_{i=1}^{N} \partial_{\sigma_2} L(y_i,\sigma_2) \partial_{f_2}\sigma_2 \partial_{w_2} f_2$$

$$\partial_{w_{1m}} \sum_{i=1}^{N} L(y_i,\sigma_2) = \sum_{i=1}^{N} \partial_{\sigma_2} L(y_i,\sigma_2) \partial_{f_2}\sigma_2 \partial_{w_1} f_2 = \sum_{i=1}^{N} \partial_{\sigma_2} L(y_i,\sigma_2) \partial_{f_2}\sigma_2 \partial_{\sigma_{1m}} f_2 \partial_{f_{1m}} \sigma_{1m} \partial_{w_{1m}} f_{1m}$$

*more complicated

Backpropagation = Gradient Descent
(lots of chain rules)

Today

- Beyond Linear Basic Linear Models
  - Support Vector Machines
  - Logistic Regression
  - Feed-forward Neural Networks
  - Different ways to interpret models
- **Different Evaluation Metrics**
- Hypothesis Testing

Evaluation

- 0/1 Loss (Classification)
- Squared Loss (Regression)
- Anything Else?

Example: Cancer Prediction

Patient

Model

Loss Function	Has Cancer	Doesn't Have Cancer
Predicts Cancer	Low	Medium
Predicts No Cancer	**OMG Panic!**	Low

- Value Positives & Negatives Differently
  - Care much more about positives
- "Cost Matrix"
  - 0/1 Loss is Special Case

Precision & Recall

- **Precision** = TP/(TP + FP)
- **Recall** = TP/(TP + FN)

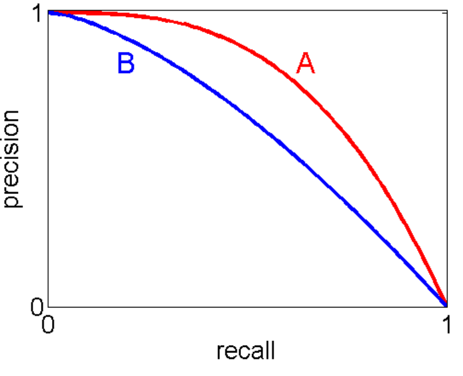
F1 = 2/(1/P+ 1/R)

Care More About Positives!

Patient

Model

Counts	Has Cancer	Doesn't Have Cancer
Predicts Cancer	20	30
Predicts No Cancer	5	70

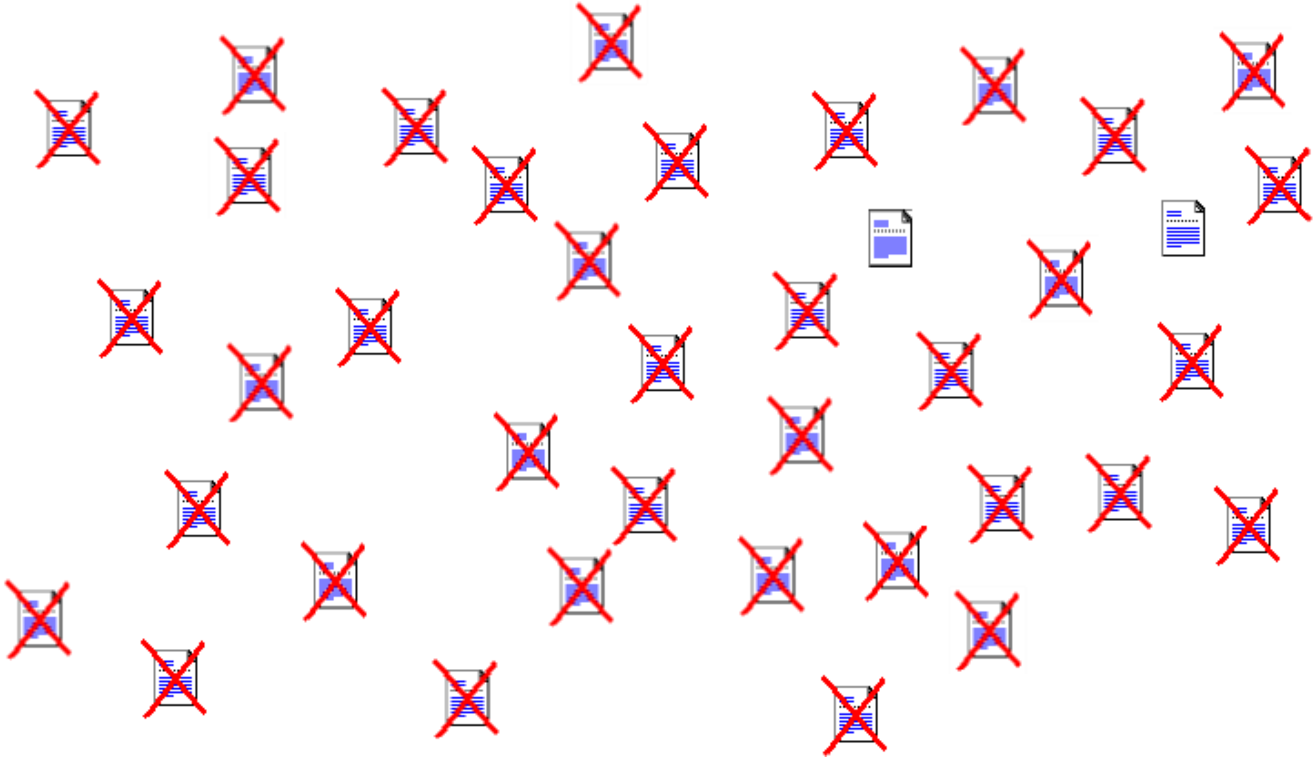


- TP = True Positive, TN = True Negative
- FP = False Positive, FN = False Negative

Image Source: http://pmtk3.googlecode.com/svn-history/r785/trunk/docs/demos/Decision_theory/PRhand.html

Example: Search Query

- Rank webpages by relevance

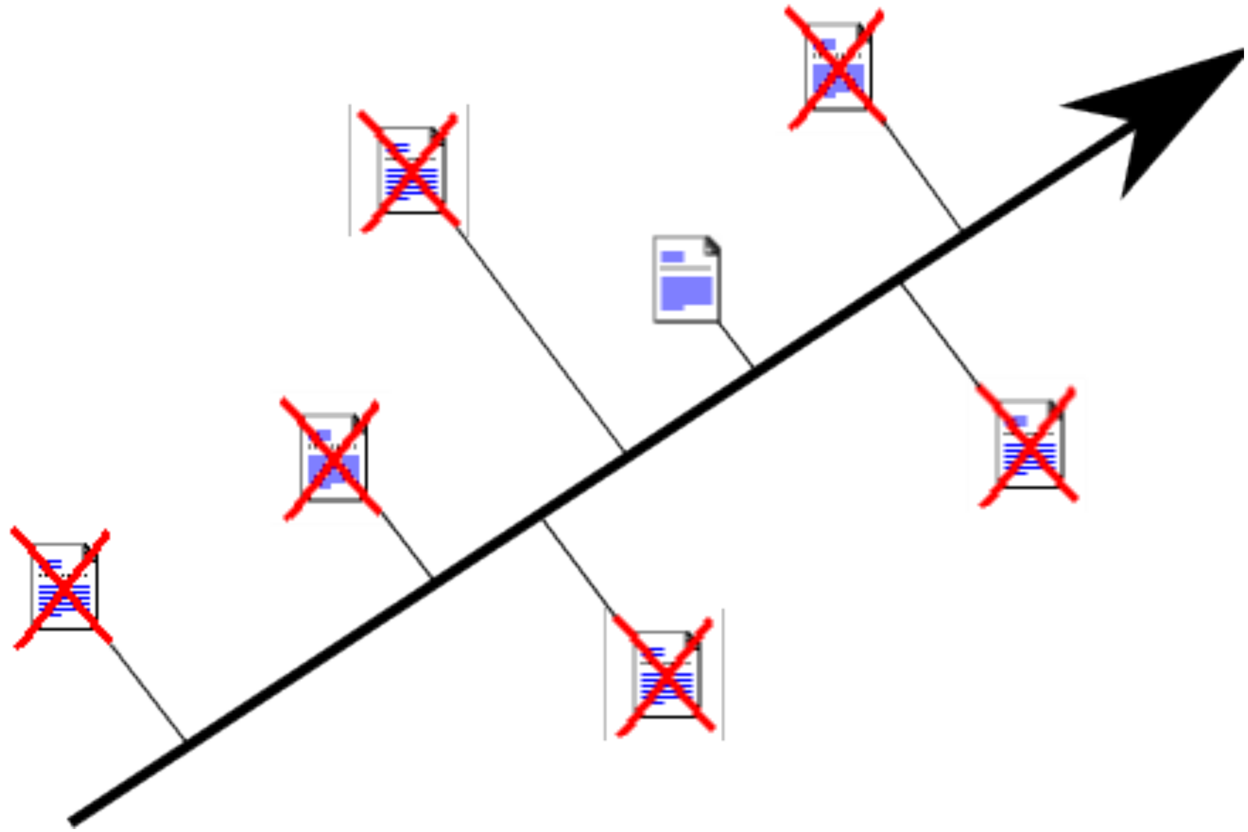


Ranking Measures

- Predict a Ranking (of webpages)
  - Users only look at top 4
  - Sort by f(x|w,b)
- Precision @4
  - Fraction of top 4 relevant
- Recall @4
  - Fraction of relevant in top 4
- Top of Ranking Only!

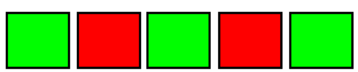
Image Source: http://pmtk3.googlecode.com/svn-history/r785/trunk/docs/demos/Decision_theory/PRhand.html

Pairwise Preferences

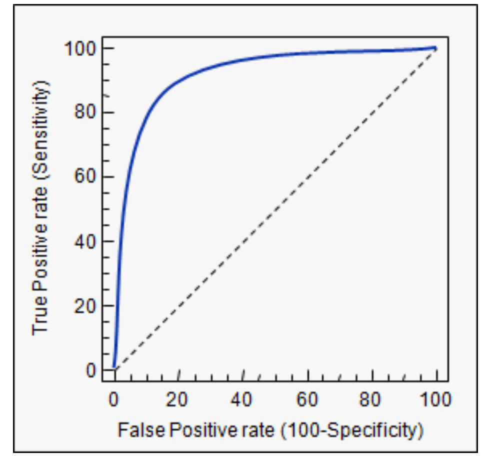


2 Pairwise Disagreements
4 Pairwise Agreements

ROC-Area & Average Precision

- ROC-Area
  - Area under ROC Curve
  - Fraction pairwise agreements
- Average Precision
  - Area under P-R Curve
  - P@K for each positive
- Example: 

ROC-Area: 0.5 AP: $\frac{1}{3}\cdot\left(\frac{1}{1}+\frac{2}{3}+\frac{3}{5}\right)\approx 0.76$



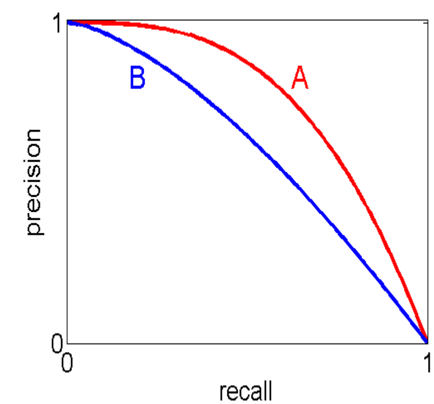


Image Source: http://pmtk3.googlecode.com/svn-history/r785/trunk/docs/demos/Decision_theory/PRhand.html
http://www.medcalc.org/manual/roc-curves.php

Summary: Evaluation Measures

- Different Evaluations Measures
  – Different Scenarios

- Large focus on getting positives
  – Large cost of mis-predicting cancer
  – Relevant webpages are rare

Today

- Beyond Linear Basic Linear Models
  - Support Vector Machines
  - Logistic Regression
  - Feed-forward Neural Networks
  - Different ways to interpret models
- Different Evaluation Metrics
- **Hypothesis Testing**

Uncertainty of Evaluation

- Model 1: 0.22 Loss on Cross Validation
- Model 2: 0.25 Loss on Cross Validation

- **Which is better?**
  - What does "better" mean?
    - True Loss on unseen test examples
  - Model 1 might be better...
  - ...or not enough data to distinguish

Uncertainty of Evaluation

- Model 1: 0.22 Loss on Cross Validation
- Model 2: 0.25 Loss on Cross Validation

- Validation set is finite
  – Sampled from "true" P(x,y)
- So there is uncertainty

Uncertainty of Evaluation

- Model 1: 0.22 Loss on Cross Validation
- Model 2: 0.25 Loss on Cross Validation

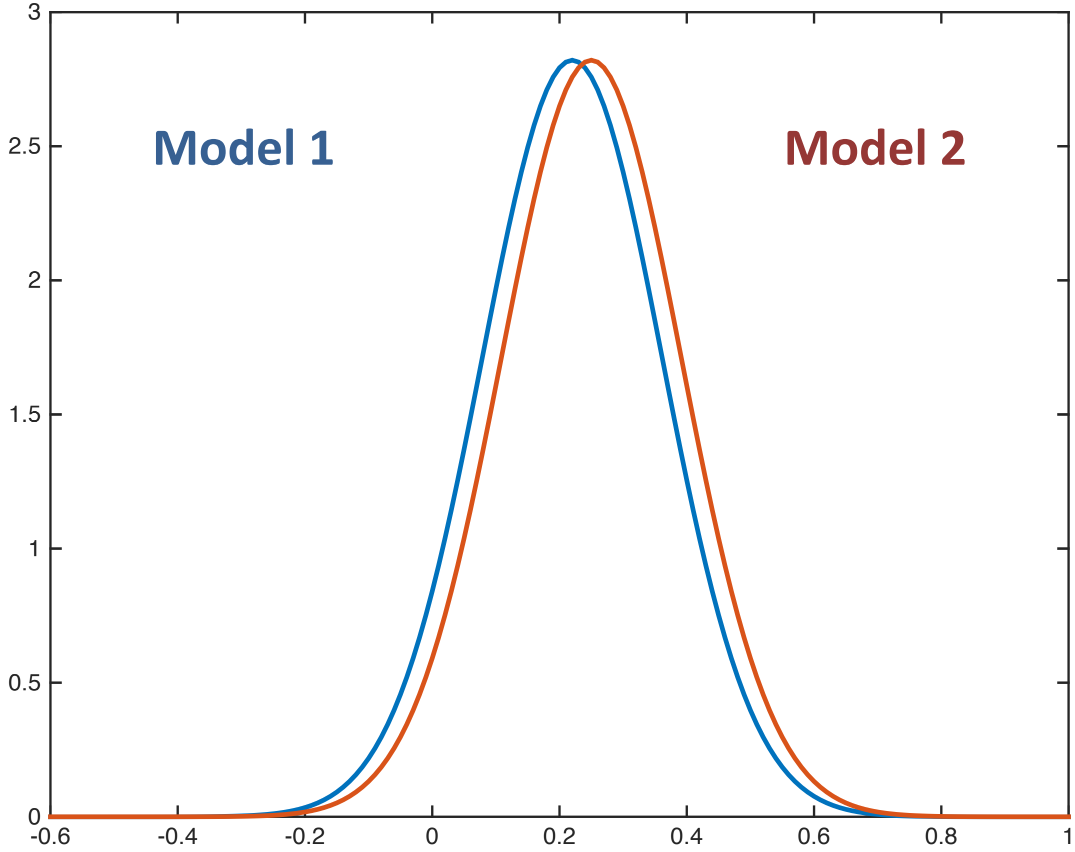
Model 1 Loss:

-0.6279	-0.9001	2.7460	1.8755	0.5275	-1.0371	-0.6455	0.0435	1.0114	-1.1120
-2.1099	-1.2291	0.5535	0.6114	0.6717	0.0897	0.4037	-0.2562	1.0820	-1.1417
0.6750	-0.6287	-0.1149	0.7728	1.2591	-0.8976	1.4807	0.8801	0.1521	0.0248
0.0024	-0.0831	0.2430	0.2713	1.0461	1.7470	0.6869	0.0103	0.8452	0.4032
-0.8098	1.1692	0.5271	0.3552	0.7352	0.4814	-0.7215	0.0577	0.0739	-0.2000

Model 2 Loss:

0.1251	1.7290	-0.6108	1.0347	0.5586	0.0161	-0.8070	-0.0341	0.1633	-1.2194
0.4422	-0.5723	0.1558	0.5862	-0.6547	-0.0383	0.6001	-1.5859	1.2860	2.6745
1.2094	-0.0658	0.6786	-0.7860	2.1279	1.1907	1.0373	-0.6259	0.5699	-0.3083
-0.0614	-0.3200	-0.7757	-0.6587	0.0401	-1.4489	0.8576	0.1322	0.9492	0.5196
0.7443	-1.2331	-0.7703	-0.1970	0.3597	1.3787	-0.0400	1.5116	0.9504	1.6843

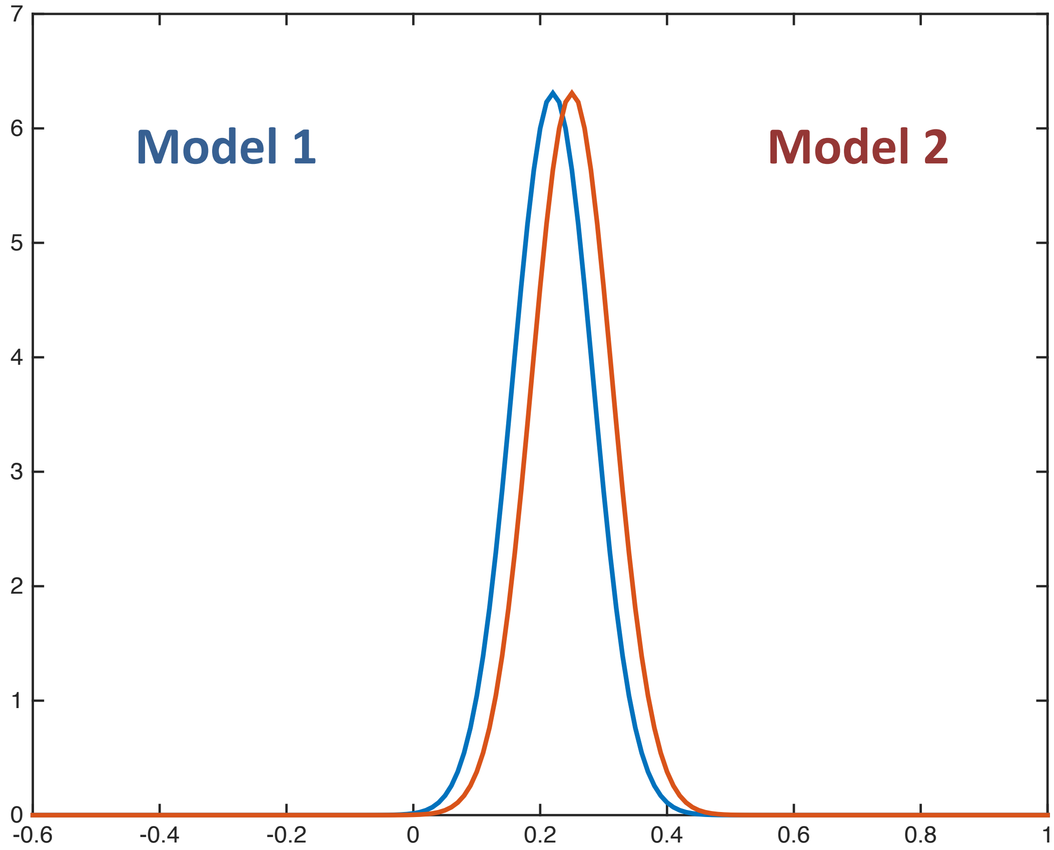
Gaussian Confidence Intervals



50 Points

See Recitation Next Wednesday!

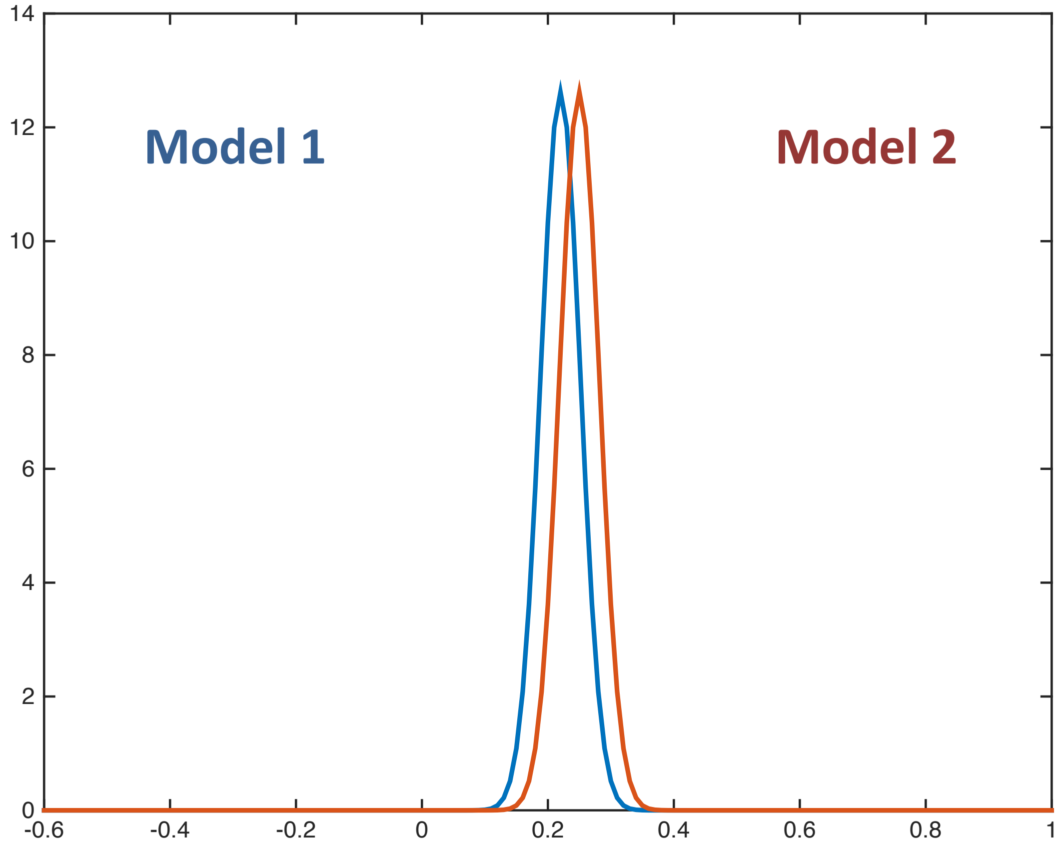
Gaussian Confidence Intervals



50 Points
250 Points

See Recitation Next Wednesday!

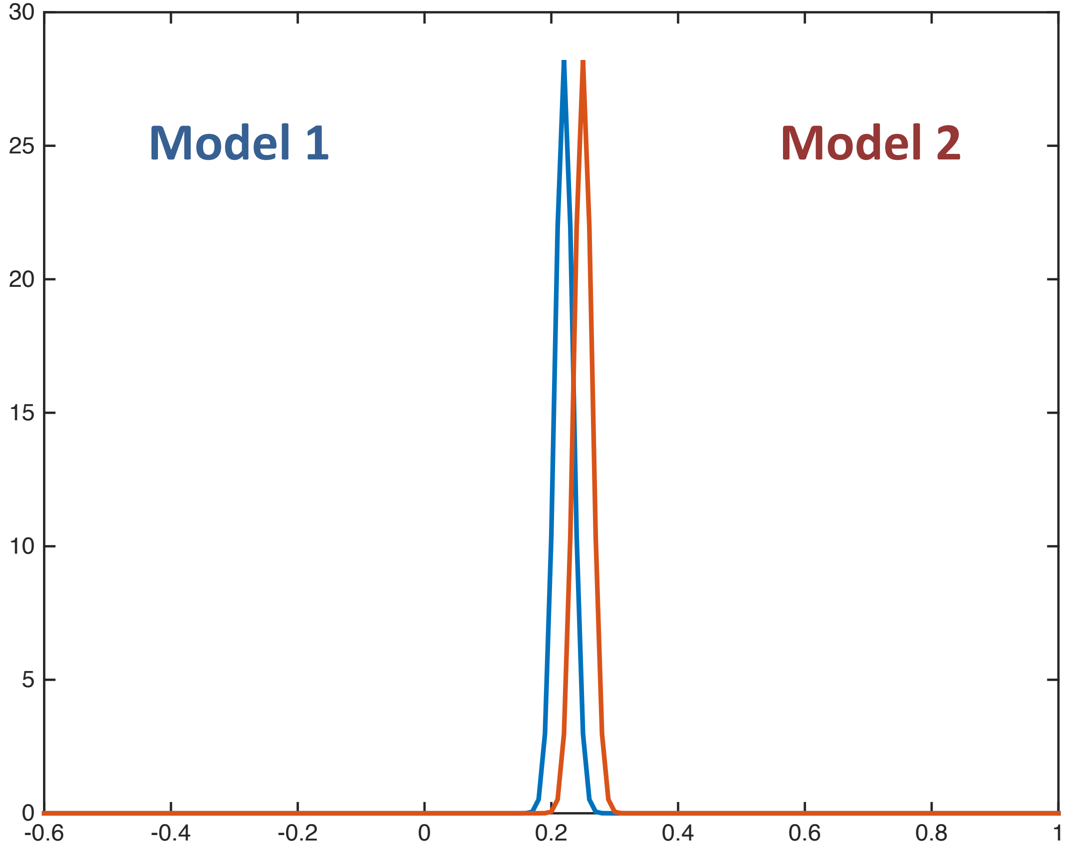
Gaussian Confidence Intervals



50 Points
250 Points
1000 Points

See Recitation Next Wednesday!

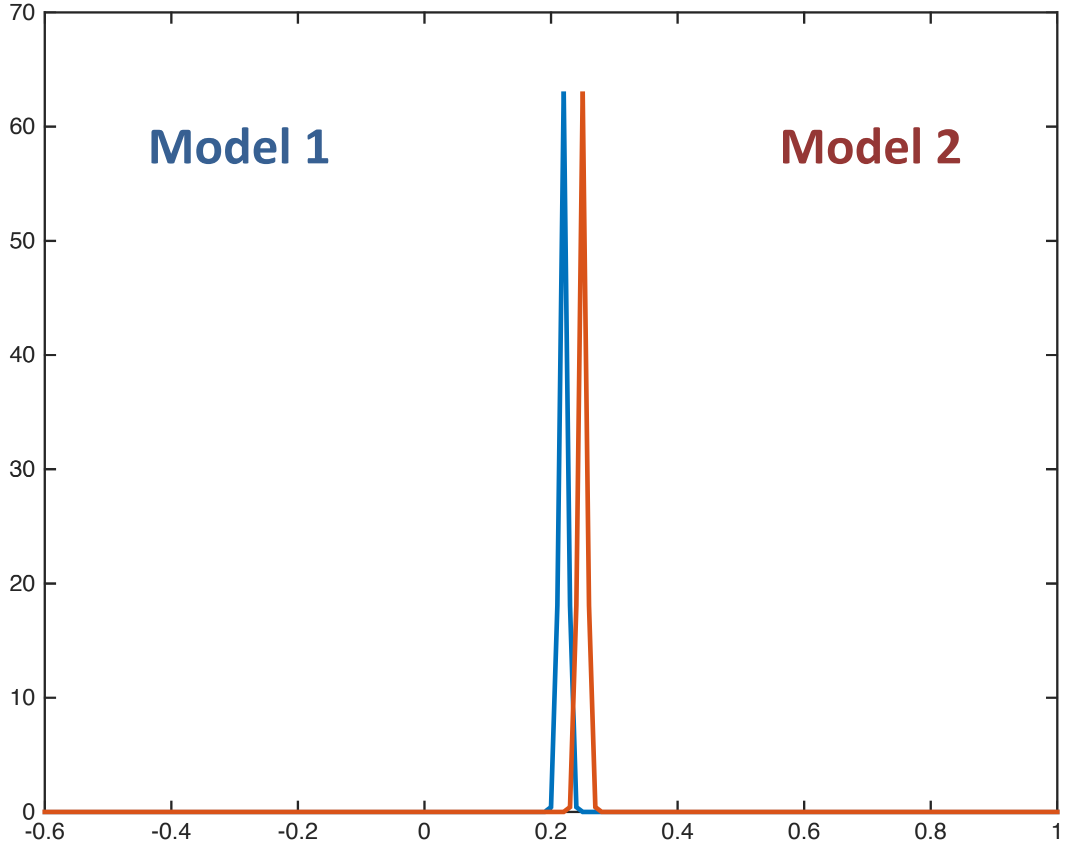
Gaussian Confidence Intervals



50 Points
250 Points
1000 Points
5000 Points

See Recitation Next Wednesday!

Gaussian Confidence Intervals



50 Points
250 Points
1000 Points
5000 Points
25000 Points

See Recitation Next Wednesday!

Next Week

- Regularization
- Lasso
- Recent Applications
- **Next Wednesday:**
  - **Recitation on Probability & Hypothesis Testing**