MRC Chalk Talk

Simple "Machines" for Learning and Predicting: Classification Algorithms

Yoonsang Kim, PhD

Institute for Health Research and Policy

University of Illinois at Chicago

March 10, 2015

Focus of the Chalk Talk

- ♣ Terminology
- ♣ Data partition for learning and prediction
- ♣ Validation

- ♣ Machine learning classifiers:
  - Naive Bayes
  - Logistic regression
  - Regularized logistic regression
  - Support vector machine

Terminology

- **Supervised learning**
- **"train"** and **"learn"**
- **Classifier**: a model/algorithm that predicts y from x
- **Label**: the class of given data, similar to the response variable, **Y**
- **Feature**: similar to the independent variable, **X**,

  Example for text data: term indicator, term frequency, TF-IDF

  (term frequency-inverse document frequency)
- **Boolean variable**: 1/0 variable

SPAM Email Example

Features

	make	address	all	num3d	our	over	remove	internet	order	mail	receive	will	people	report	addresses	free	business
1	0.00	0.64	0.64	0	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.64	0.00	0.00	0.00	0.32	0.00
2	0.21	0.28	0.50	0	0.14	0.28	0.21	0.07	0.00	0.94	0.21	0.79	0.65	0.21	0.14	0.14	0.07
3	0.06	0.00	0.71	0	1.23	0.19	0.19	0.12	0.64	0.25	0.38	0.45	0.12	0.00	1.75	0.06	0.06
4	0.00	0.00	0.00	0	0.63	0.00	0.31	0.63	0.31	0.63	0.31	0.31	0.31	0.00	0.00	0.31	0.00
5	0.00	0.00	0.00	0	0.63	0.00	0.31	0.63	0.31	0.63	0.31	0.31	0.31	0.00	0.00	0.31	0.00
6	0.00	0.00	0.00	0	1.85	0.00	0.00	1.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	email	you	credit	your	font	num000	money	hp	hpl	george	num650	lab	labs	telnet	num857	data	num415	num85
1	1.29	1.93	0.00	0.96	0	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0
2	0.28	3.47	0.00	1.59	0	0.43	0.43	0	0	0	0	0	0	0	0	0	0	0
3	1.03	1.36	0.32	0.51	0	1.16	0.06	0	0	0	0	0	0	0	0	0	0	0
4	0.00	3.18	0.00	0.31	0	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0
5	0.00	3.18	0.00	0.31	0	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0
6	0.00	0.00	0.00	0.00	0	0.00	0.00	0	0	0	0	0	0	0	0	0	0	0

	technology	num1999	parts	pm	direct	cs	meeting	original	project	re	edu	table	conference	charSemicolon
1	0	0.00	0	0	0.00	0	0	0.00	0	0.00	0.00	0	0	0.00
2	0	0.07	0	0	0.00	0	0	0.00	0	0.00	0.00	0	0	0.00
3	0	0.00	0	0	0.06	0	0	0.12	0	0.06	0.06	0	0	0.01
4	0	0.00	0	0	0.00	0	0	0.00	0	0.00	0.00	0	0	0.00
5	0	0.00	0	0	0.00	0	0	0.00	0	0.00	0.00	0	0	0.00
6	0	0.00	0	0	0.00	0	0	0.00	0	0.00	0.00	0	0	0.00

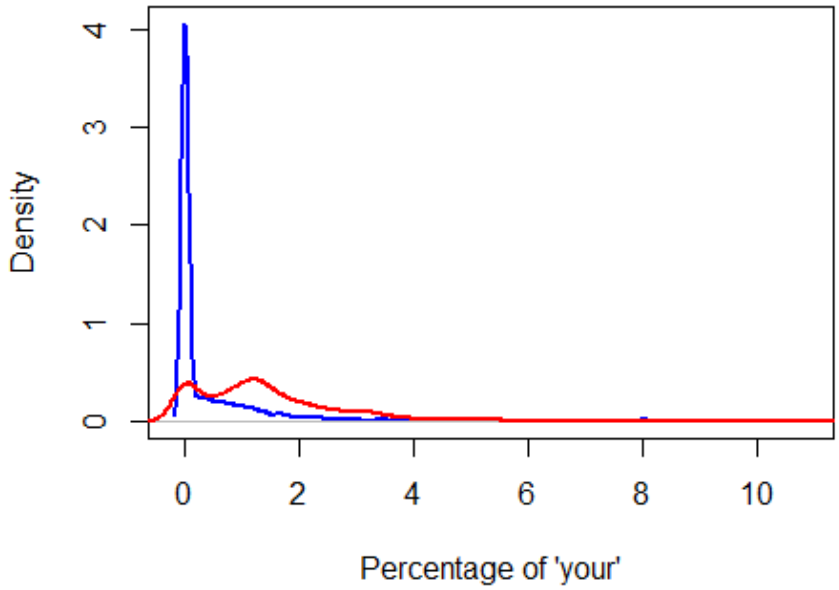
	charRoundbracket	charSquarebracket	charExclamation	charDollar	charHash	capitalAve	capitalLong	capitalTotal	type
1	0.000	0	0.778	0.000	0.000	3.756	61	278	spam
2	0.132	0	0.372	0.180	0.048	5.114	101	1028	spam
3	0.143	0	0.276	0.184	0.010	9.821	485	2259	spam
4	0.137	0	0.137	0.000	0.000	3.537	40	191	spam
5	0.135	0	0.135	0.000	0.000	3.537	40	191	spam
6	0.223	0	0.000	0.000	0.000	3.000	15	54	spam

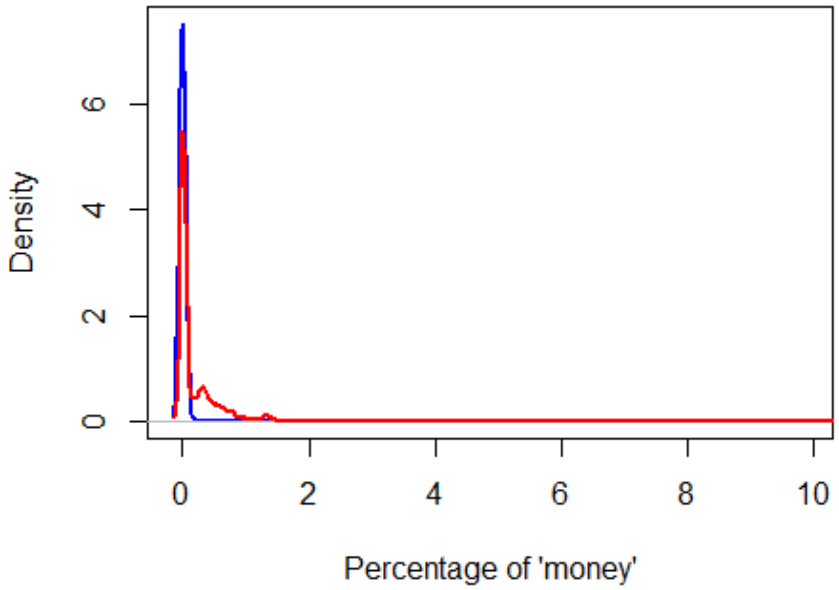
Spam data from R package kernlab

http://archive.ics.uci.edu/ml/datasets/Spambase

SPAM Email Example

BLUE = nonspam RED = spam





A **really** simple algorithm:

Classify as spam if freq of 'your' > threshold

Data Partition

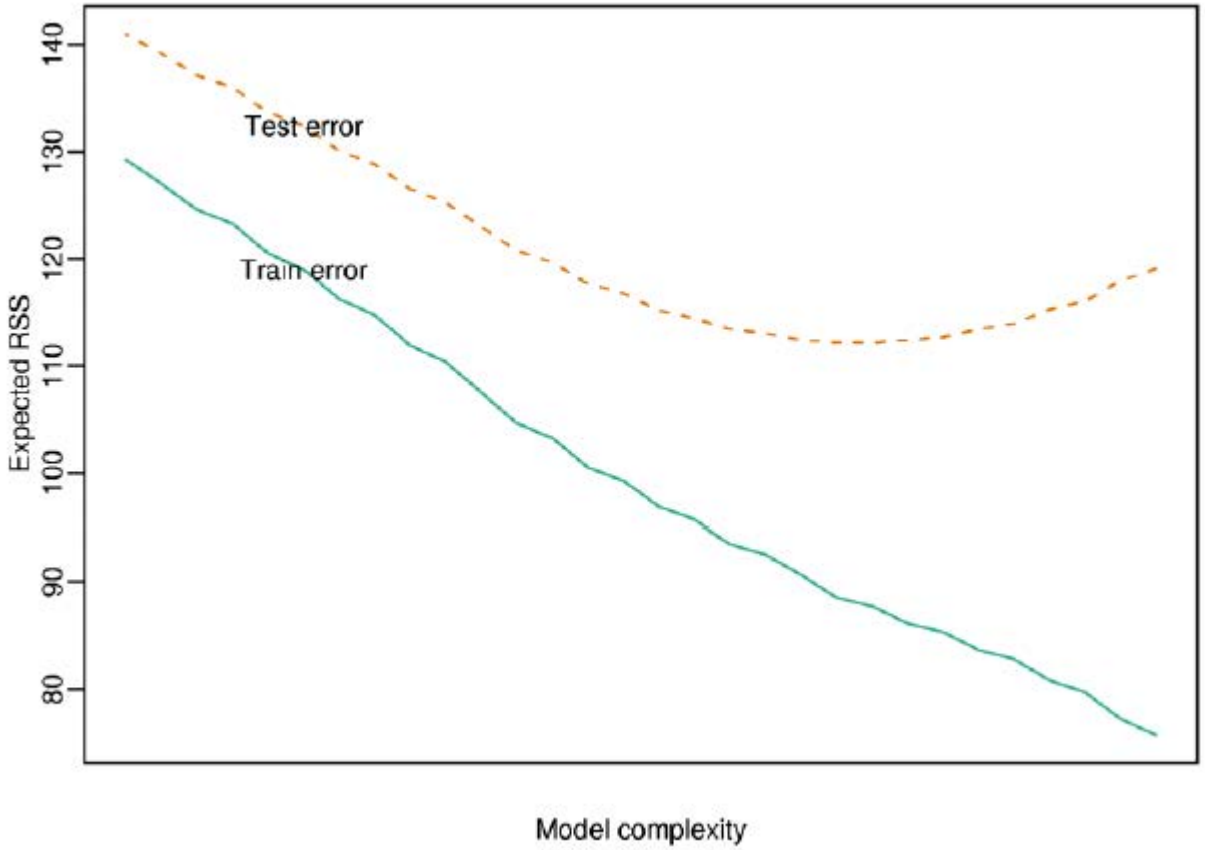
Train (to build a model)	**Test** (to evaluate)

Train (to build a model)	**Validation** (to refine)	**Test** (to evaluate)

- Build a prediction model on training set -> fitted model
- Testing set is held out at the beginning
  - To calculate generalization error
  - To prevent overfitting

Data Partition

- Test Error vs. Train Error



http://www.biostat.jhsph.edu/~ririzarr/Teaching/649

Validation

- Suppose you have a great fitted model, now what?
- Check prediction accuracy

SPAM example: Classified as spam if freq of 'your' > 0.4%

	Label	
Prediction	Spam	NonSpam
Spam	1395	731
NonSpam	418	2057

Percent agreement = 75%

Precision = 1395/(1395+731)*100 = 65.6%

Recall = 1395/(1395+418)*100 = 77%

Report accuracy on **testing** set.

Machine Learning Classifiers

Naive Bayes

- **Bayes rule**

  For Boolean variable Y, i.e. $y_i \in \{0,1\}$

$$P(Y = y_i | X_1, \dots, X_p) = \frac{P(X_1, \dots, X_p | Y = y_i) P(Y = y_i)}{\sum_i P(X_1, \dots, X_p | Y = y_i) P(Y = y_i)}$$

$$\propto P(X_1, \dots, X_p | Y = y_i) P(Y = y_i)$$

- **Conditional independence** assumption of X given Y:

$$P(X_1, \dots, X_p | Y) = P(X_1 | Y) P(X_2 | Y) \cdots P(X_p | Y)$$

e.g. P(your, money, credit|spam)

= P(your|spam) x P(money|spam) x P(credit|spam)

parameters = 3 instead of 2^3

suppose 'your', 'money', 'credit' are indicator variables

Allow to reduce # of parameters!

Naive Bayes

- Need to estimate two kinds of parameters:

$$\theta_{ij} = P(X_j = x_j | Y = y_i)$$

$$\pi_i = P(Y = y_i)$$

$\hat{\theta}_{ij}$ and $\hat{\pi}_{ij}$ are relative freq from training set or Bayesian estimates

- For each new document, calculate

$P(y_i = 1) \prod_j P(X_j | y_i = 1)$ and $P(y_i = 0) \prod_i P(X_j | y_i = 0)$

→ A larger value will determine $\hat{y}_i$

- Naive assumption (conditional independence) but it works well.

Logistic Regression

- Regression equation

  For Boolean variable Y, i.e. $y_i \in \{0,1\}$

  X = (your, money, credit)
  Y = spam or nonspam

$$\ln \frac{P(Y=0|X)}{P(Y=1|X)} = b + \sum \beta_j x_j$$

- How to classify?

$$\frac{P(Y=0|X)}{P(Y=1|X)} > 1 \quad \Leftrightarrow \quad P(Y=0|X) > P(Y=1|X)$$

$$\Leftrightarrow \exp\left(b + \sum \beta_j x_j\right) > 1$$

$$\Leftrightarrow b + \sum \beta_j x_j > 0$$

then $\hat{y} = 0$, otherwise $\hat{y} = 1$

Regularized Logistic Regression

- Parameter estimation for the *usual* logistic regression

  ML : Find β that maximizes the log-likelihood function

$$\ell(\beta) = \sum \ln P(y|x,\beta)$$

- Regularization:

$$\ell(\beta) = \sum ln\, P(y|x,\beta) - \frac{\lambda}{2}\|\beta\|^2$$

  To penalize large values of β_j

Regularized Logistic Regression

- Parameter estimation for the *usual* logistic regression

  ML : Find β that maximizes the log-likelihood function

$$\ell(\beta) = \sum \ln P(y|x,\beta)$$

- Regularization:

$$\ell(\beta) = \sum ln\, P(y|x,\beta) - \frac{\lambda}{2}\|\beta\|^2$$

$\|\beta\|^2 = \beta^T\beta$
$= \sum\beta_j^2$

To penalize large values of β_j

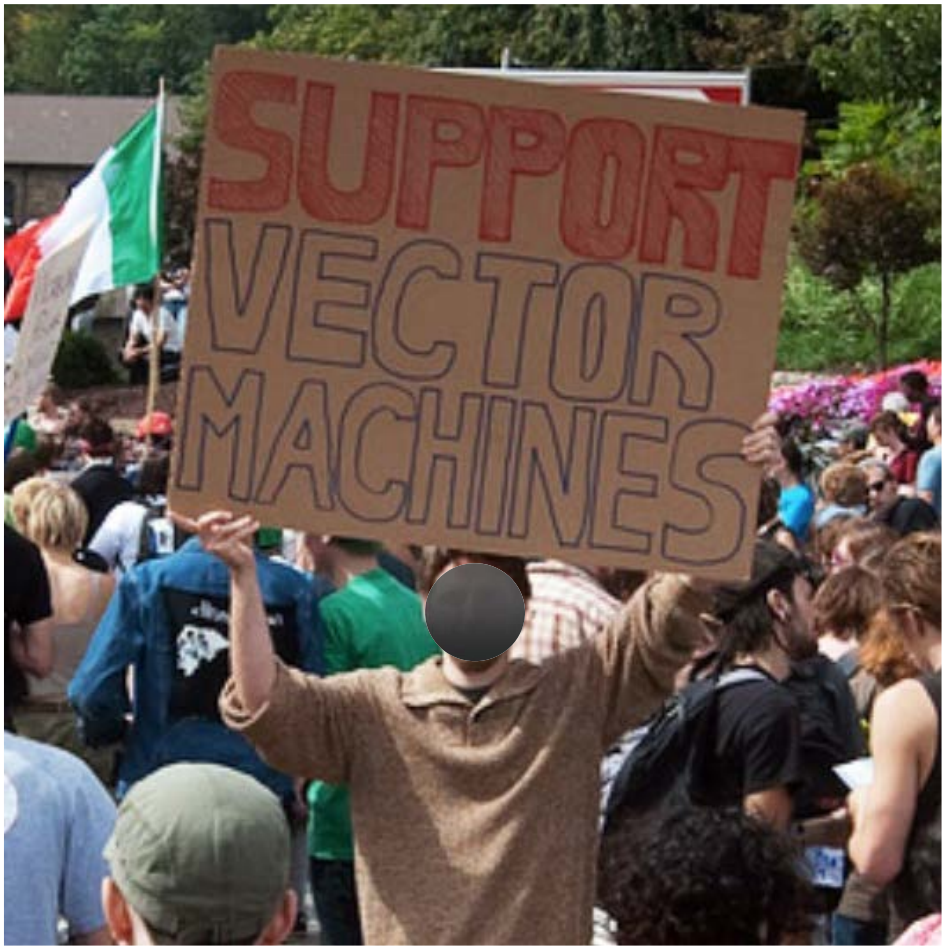
so-called "penalized log-likelihood"

λ = a constant determining the strength of penalty

Helps reduce overfitting!

esp. when data is high dimensional and training set is sparse

Support Vector Machines

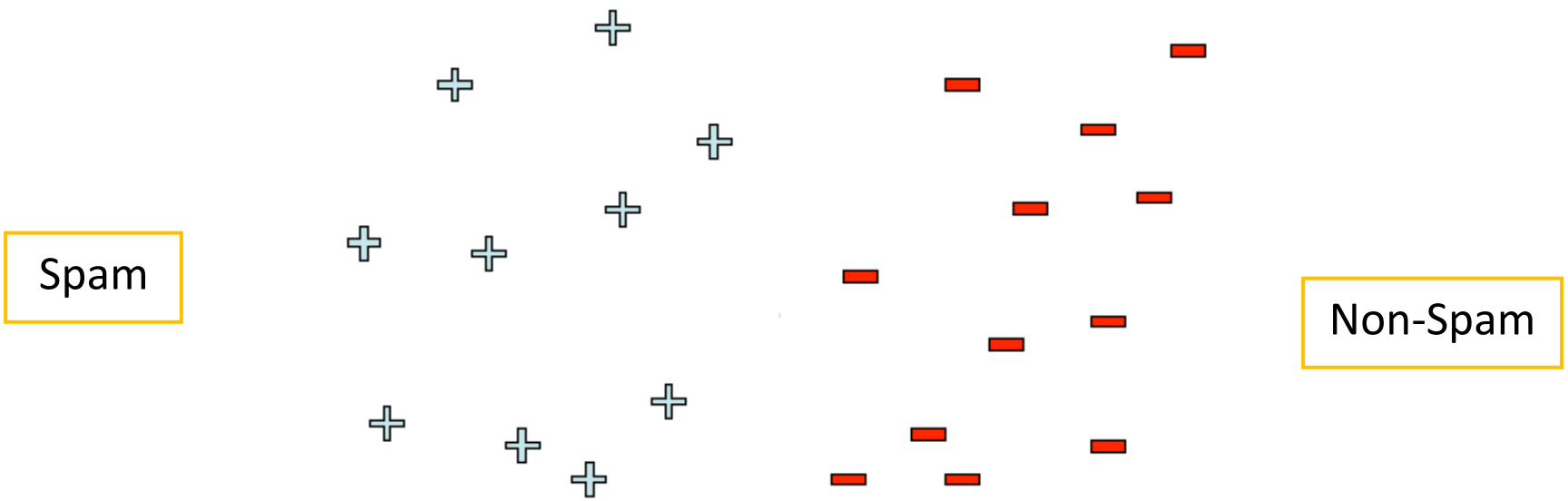


- Most widely used
- Maximize margins
- Kernel function for efficient computation

From Tom Michell's lecture slide. CMU machine learning dept protest.

Support Vector Machines

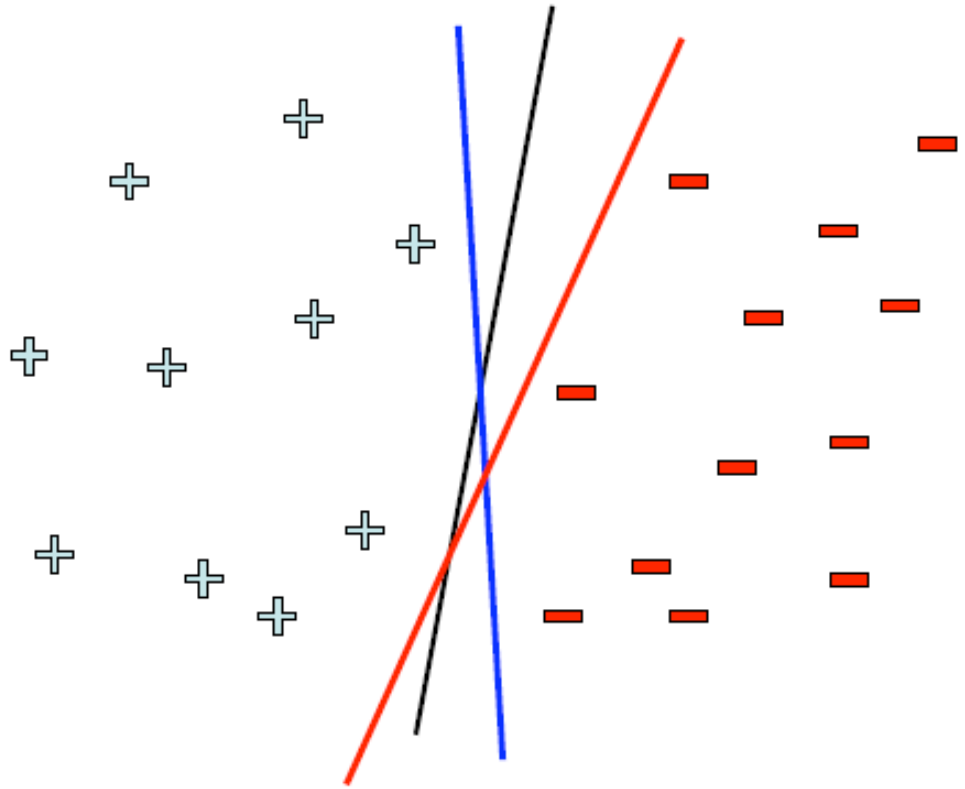
Draw a line to separate + and –



Each point is a vector of features/predictors

Support Vector Machines

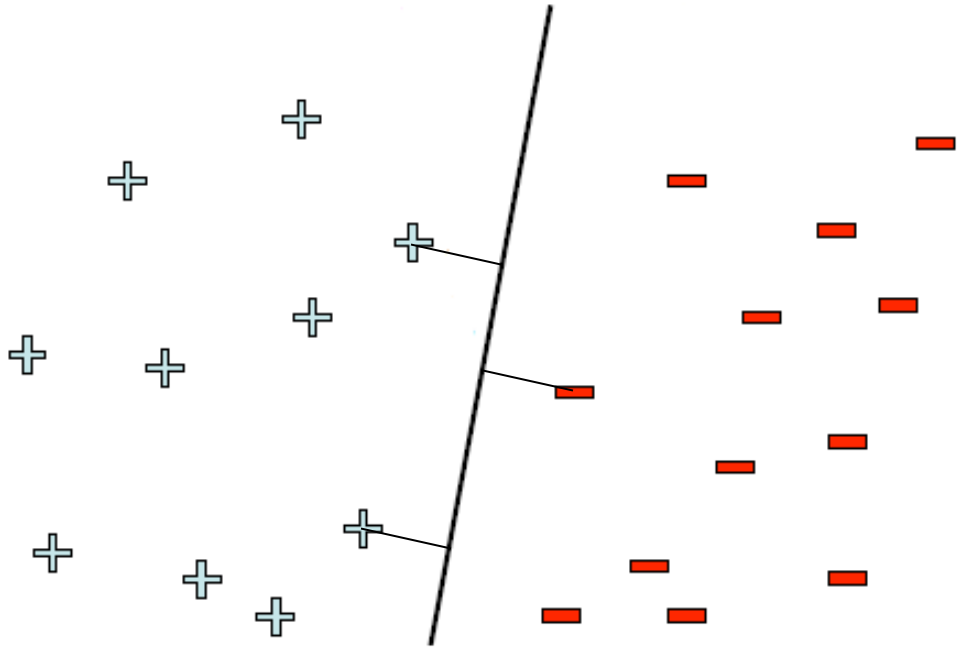
Which line is the best in separating + and – ?



Support Vector Machines

Margin:

Distance btwn the decision boundary and the closest points



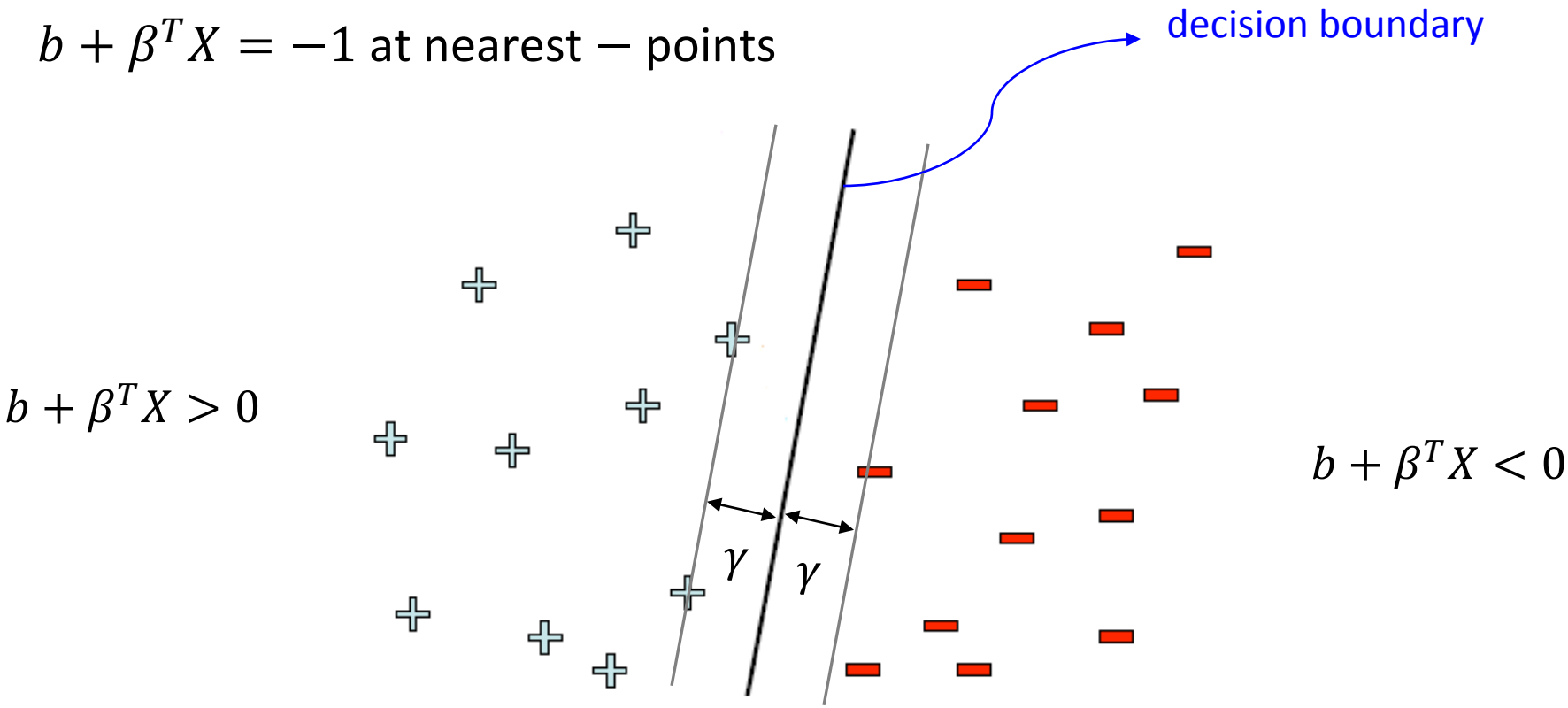
Class label $y \in \{-1, 1\}$

Support Vector Machines

$b + \beta^T X = 0$ at decision boundary

$b + \beta^T X = 1$ at nearest $+$ points

$b + \beta^T X = -1$ at nearest $-$ points

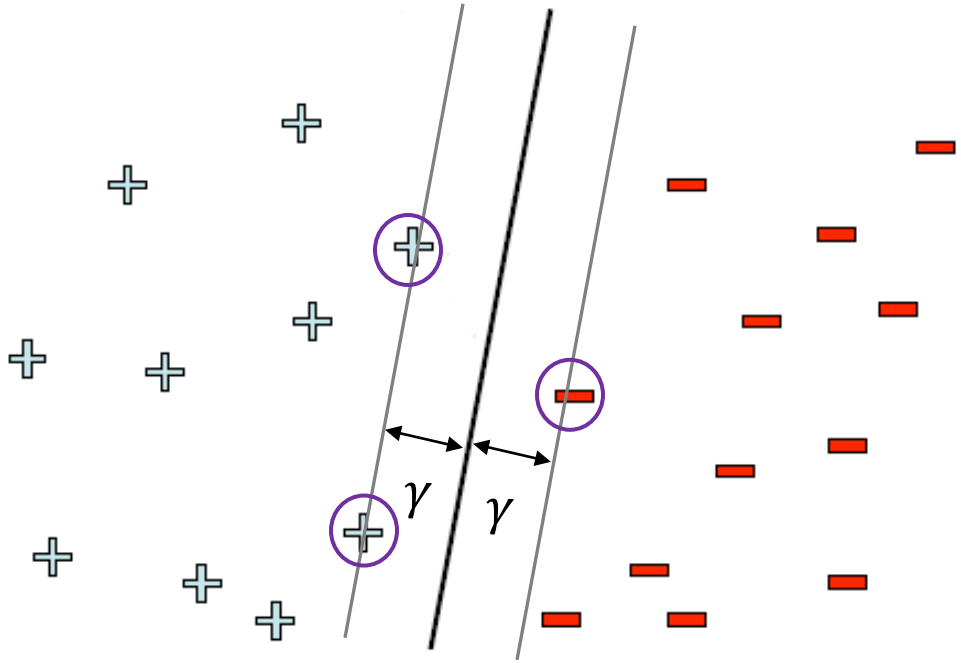


Maximize the margin $\gamma = 1/\|\beta\|$ $\Leftrightarrow$ Minimize $\|\beta\|$

Support Vector Machines

Support Vectors:

Data points nearest to the decision boundary

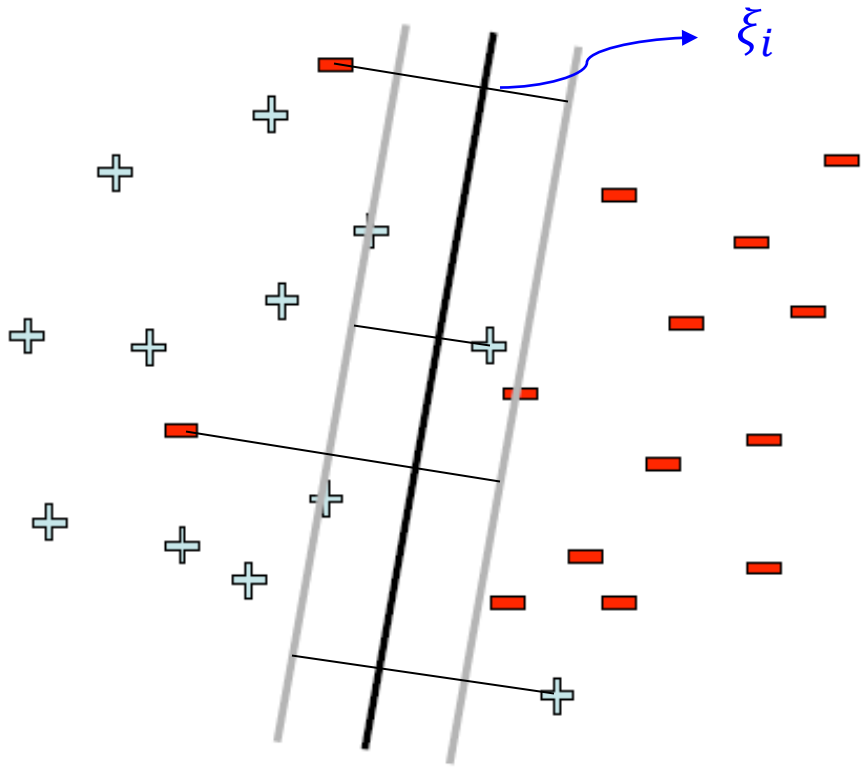


Minimize $\|\beta\|$ such that $(b + \beta^T X_i) y_i \geq 1$ where $y_i \in \{-1, 1\}$

Support Vector Machines

SVM with Soft Margin

For noisy data... allow some error in classification



Minimize $\|\beta\| + C \sum \xi_i$ such that $(b + \beta^T X_i) y_i \geq 1 - \xi_i , \xi_i \geq 0$

Support Vector Machines

Estimating algorithm:

Minimize $\|\beta\| + C \sum \xi_i$ such that $(b + \beta^T X_i) y_i \geq 1 - \xi_i$

$\xi_i =$ "slack" variable (not all mistakes are equally bad)

$$\xi_i = |y_i - (b + \beta^T X_i)| \geq 0$$

$C =$ regularization parameter, large value indicates high penalty for mistakes

What is Kernel function, but first dual form....

Dual form for prediction with new predictors $\mathbf{x}$:

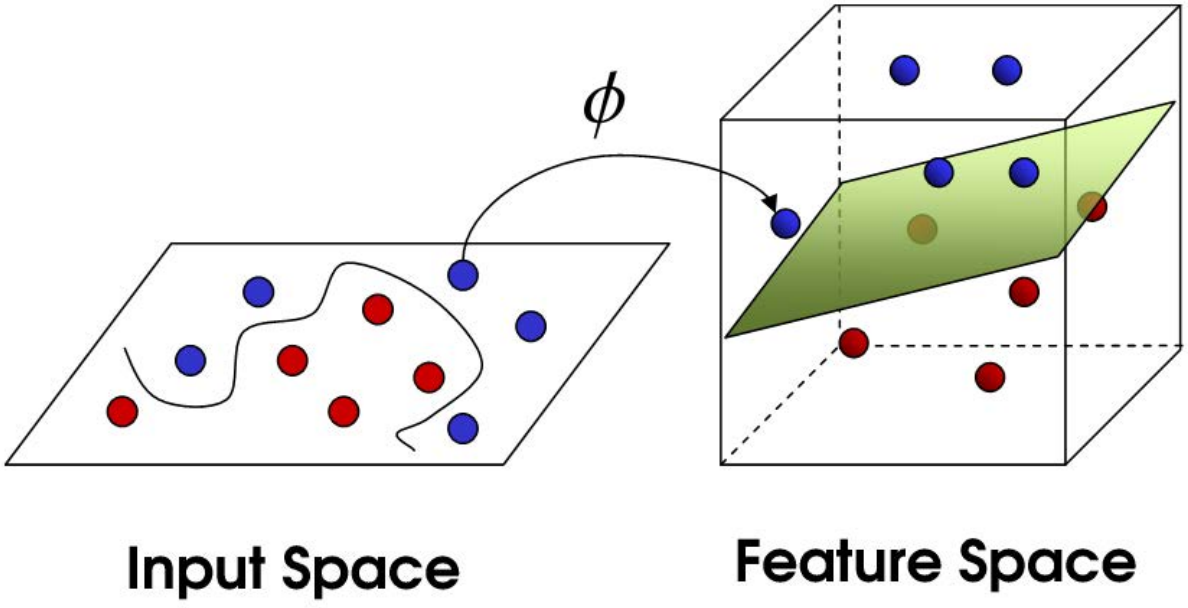
$\boldsymbol{\beta}^T \mathbf{x} = \sum_i \alpha_i \mathbf{x}^T \mathbf{x_i}$ for a scalar $\alpha_i, i =$ index of training set

$= \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i)$ for a Kernel function *K*

Support Vector Machines

Kernel function:

- Projects low dimensional space to high dimensional space.
- Complex in low dimension but simpler in higher dimensions
- Computation is performed in low dim space → efficient computation!
- Useful for strings of text or image data



ϕ: mapping function from low to high dimension feature space.

http://i.imgur.com/WuxyO.png

Support Vector Machines

Kernel functions (pretend to) project $\mathbf{x}^T\mathbf{x_i}$ in original space to $\phi(\mathbf{x})^T\phi(\mathbf{x})$ in high dim space.

Key points of SVM

- Kernel trick
- Sparsity (support vectors)
- Maximized margin

References

Kevin Murphy. Machine Learning: A Proablistic Prespective (adaptive computation and machine learning series.) The MIT Press, 2012

Bing Liu. Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies, 2012

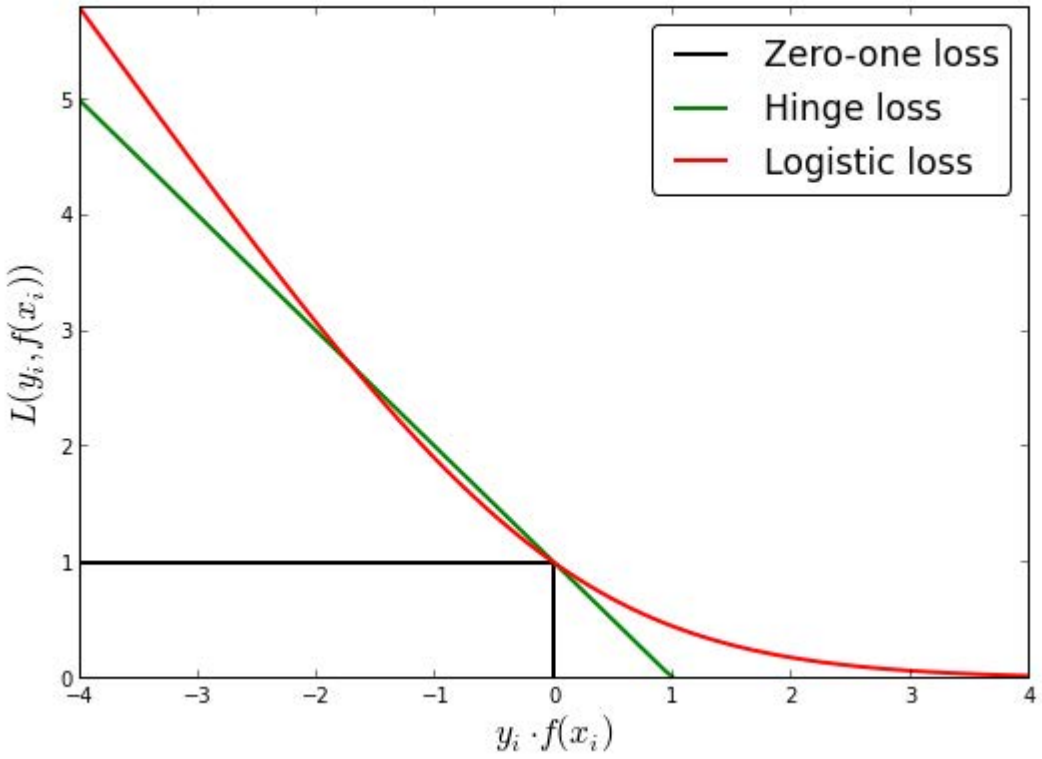
Tom Mitchell's lecture on Machine Learning
http://www.cs.cmu.edu/~tom/10701_sp11/lectures.shtml
http://www.cs.cmu.edu/~tom/NewChapters.html

Rafael Irizarry's lecture on Statistical Learning: Algorithmic and Nonparametric Approaches
http://www.biostat.jhsph.edu/~ririzarr/Teaching/649/

APPENDIX

SVM and Regularized Logistic Regression

Similar performance - why?



http://fa.bianp.net/blog/static/images/2013/loss_functions.png