CS570: Introduction to Data Mining

Classification Advanced

Reading: Chapter 8 & 9 Han, Chapters 4 & 5 Tan

Anca Doloc-Mihu, Ph.D.

Slides courtesy of Li Xiong, Ph.D.,

September 19, 2013

Classification and Prediction

- Last lecture
  - Overview
  - Decision tree induction
  - Bayesian classification
- Today
  - Training (learning) Bayesian network
  - kNN classification and collaborative filtering
  - Support Vector Machines (SVM)
- Upcoming lectures
  - Rule based methods
  - Neural Networks
  - Regression
  - Ensemble methods
  - Model evaluation

Training Bayesian Networks

- Several scenarios:
  - Given both the network structure and all variables observable: *learn only the CPTs*
  - Network structure known, some hidden variables: *gradient descent* (greedy hill-climbing) method, analogous to neural network learning
  - Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
  - Unknown structure, all hidden variables: No good algorithms known for this purpose
- Ref. D. Heckerman: Bayesian networks for data mining

Training Bayesian Networks

- Scenario: Given both the network structure and all variables observable: *learn only the CPT* (similar to naive Bayesien)

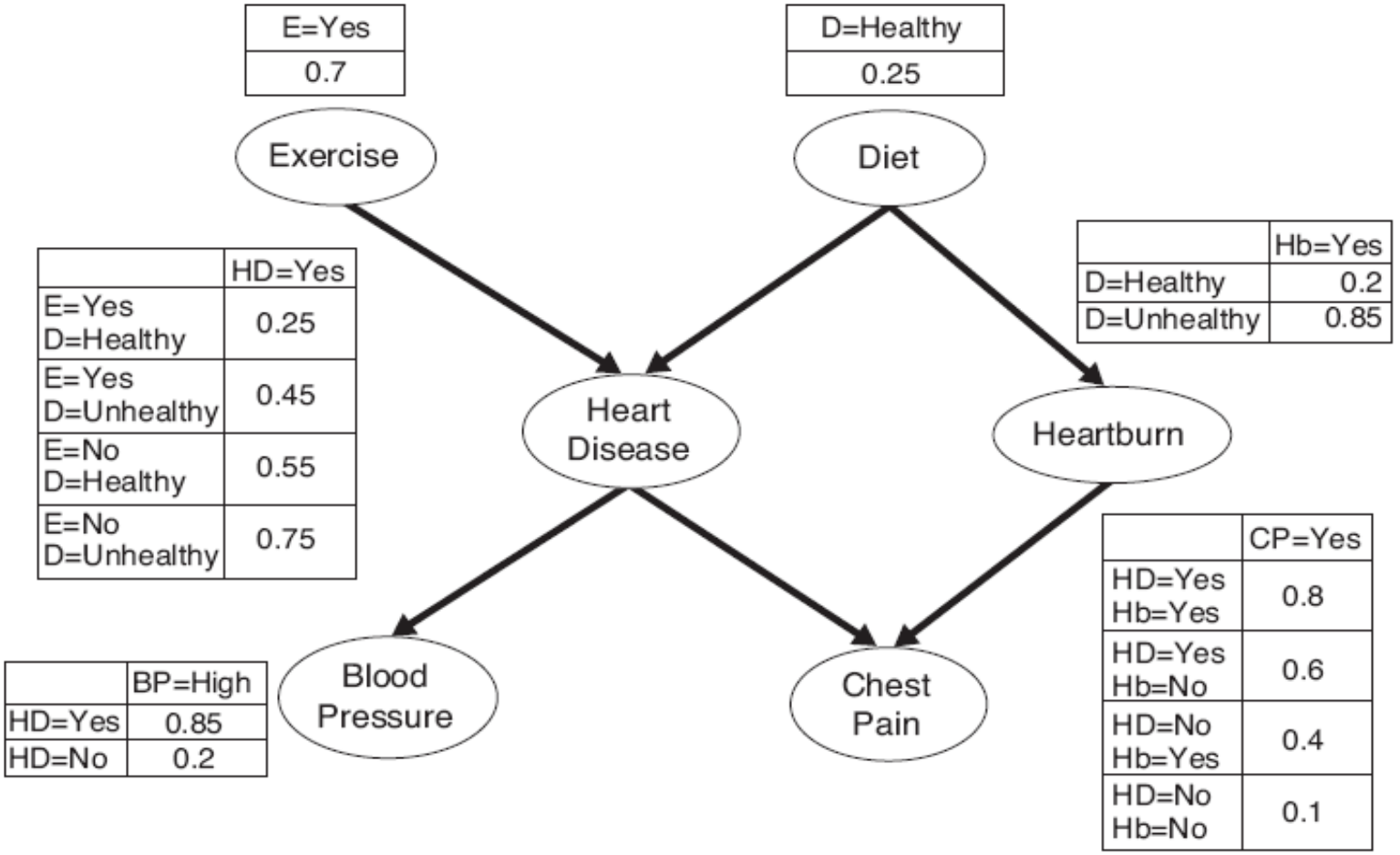


Figure 5.13. A Bayesian belief network for detecting heart disease and heartburn in patients.

Training Bayesian Networks

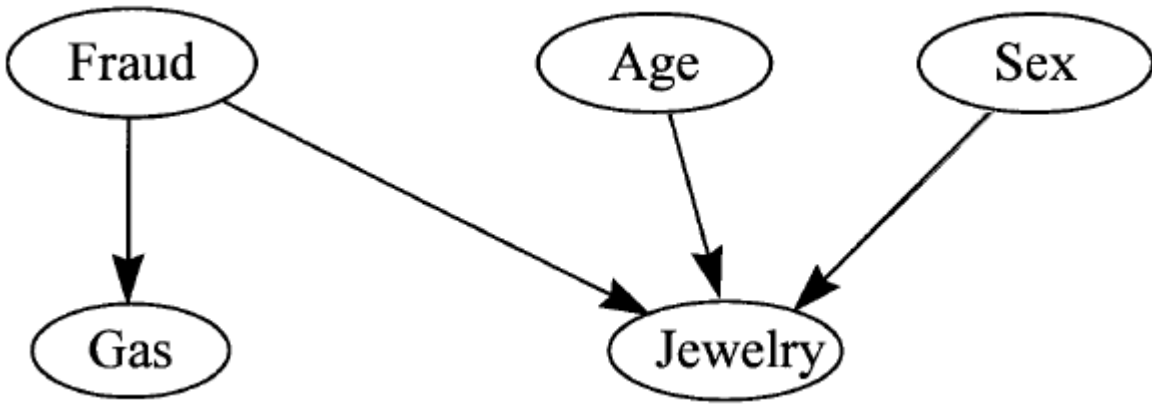
- Scenario: Network structure known, some variables hidden: *gradient descent* (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function (similar to neural network training)
  - Example optimization function: likelyhood of observing the data
  - Weights are initialized to random probability values
  - At each iteration, it moves towards what appears to be the best solution at the moment, w.o. backtracking
  - Weights are updated at each iteration & converge to local optimum

Training Bayesian Networks

- Scenario: Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
  - Define a total order of the variables
  - Construct sequences and for each sequence remove the variables that do not affect the current variable
  - Creating an arc using remaining dependencies

(F, A, S, G, J)

$$p(a \mid f) = p(a)$$
$$p(s \mid f, a) = p(s)$$
$$p(g \mid f, a, s) = p(g \mid f)$$
$$p(j \mid f, a, s, g) = p(j \mid f, a, s)$$



Classification and Prediction

- Last lecture
  - Overview
  - Decision tree induction
  - Bayesian classification
- Today
  - Training (learning) Bayesian network
  - kNN classification and collaborative filtering
  - Support Vector Machines (SVM)
- Upcoming lectures
  - Rule based methods
  - Neural Networks
  - Regression
  - Ensemble methods
  - Model evaluation

Lazy vs. Eager Learning

- Lazy vs. eager learning
  - Lazy learning (e.g. instance-based learning): stores training data (or only minor processing) and waits till receiving test data
  - Eager learning (e.g. decision tree, Bayesian): constructs a classification model before receiving test data

- Efficiency
  - Lazy learning: less time in training but more in predicting
  - Eager learning: more time in training but less in predicting

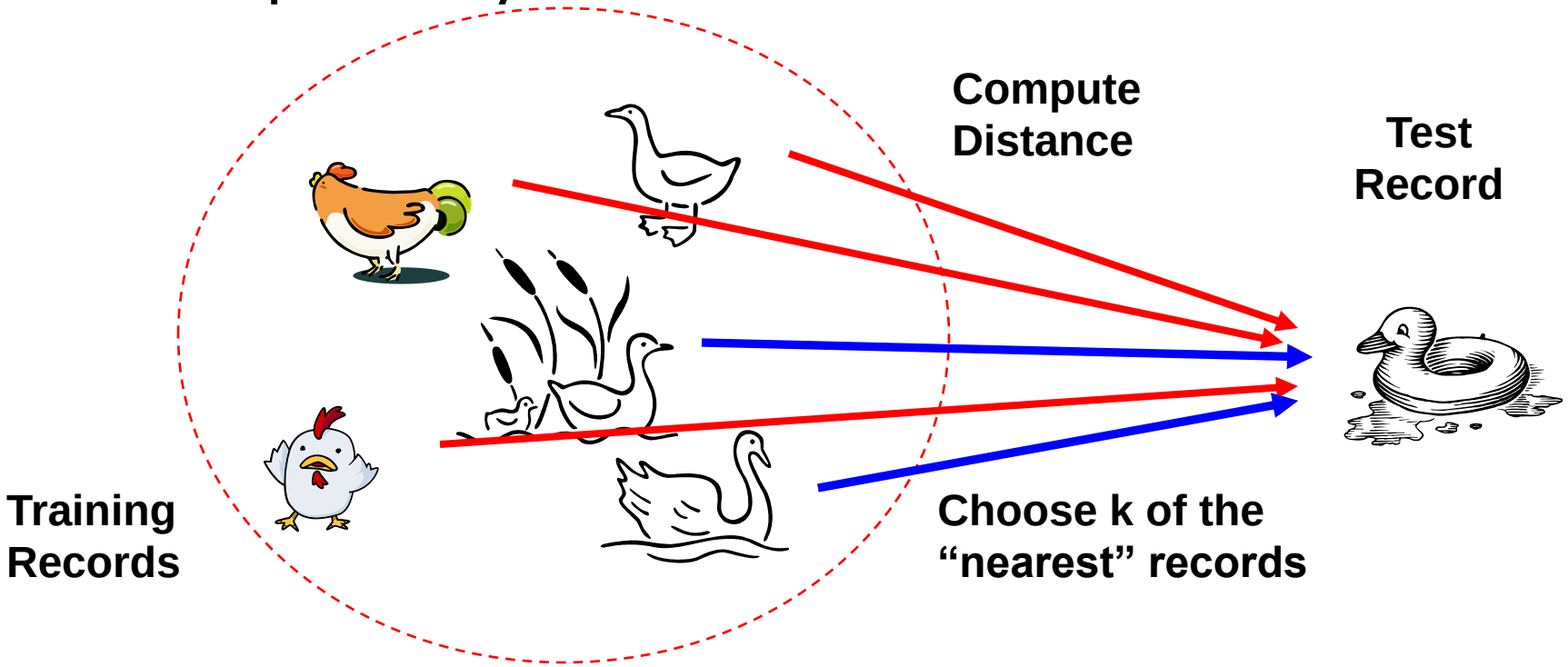
- Accuracy
  - Lazy learning: effectively uses a richer hypothesis space by using many local linear functions to form its global approximation to the target function
  - Eager learning: must commit to a single hypothesis that covers the entire instance space

Lazy Learner: Instance-Based Methods

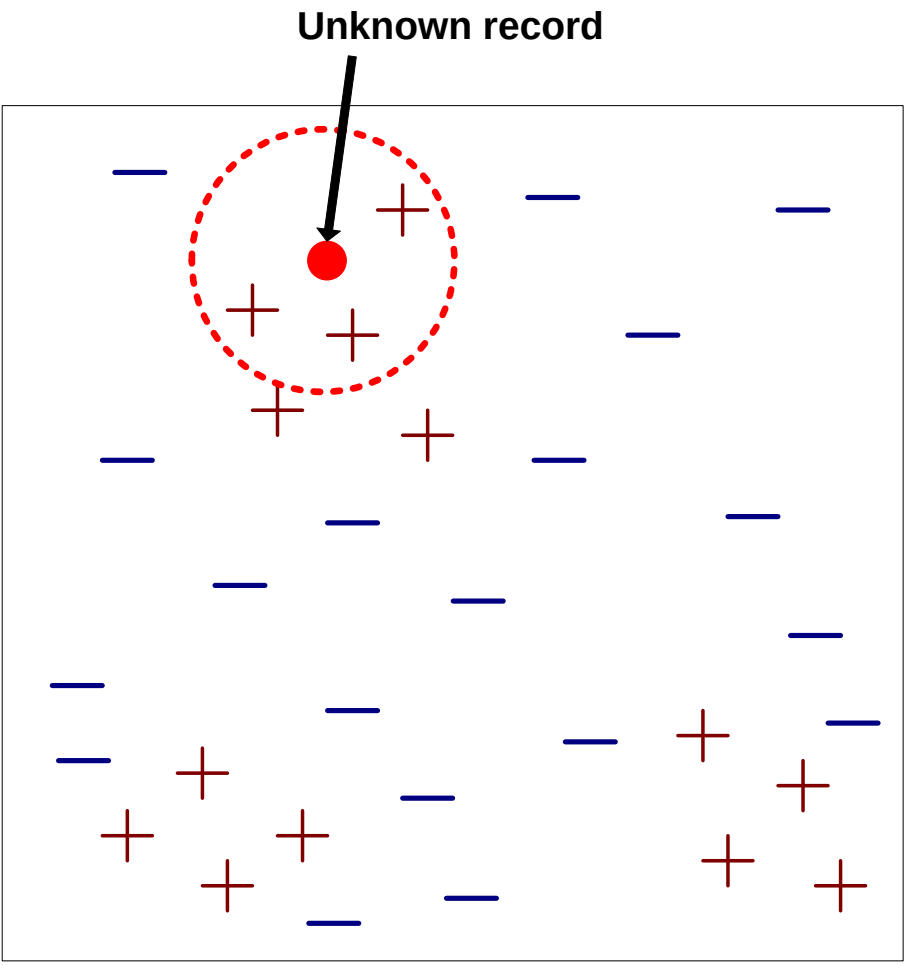
- Typical approaches
  - *k*-nearest neighbor approach (1950's)
    - Instances represented as points in a Euclidean space.
  - Locally weighted regression
    - Constructs local approximation

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest-Neighbor Classifiers



Algorithm

- Compute distance from (every) test record to (all) training records
- Identify *k* nearest neighbors
- Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

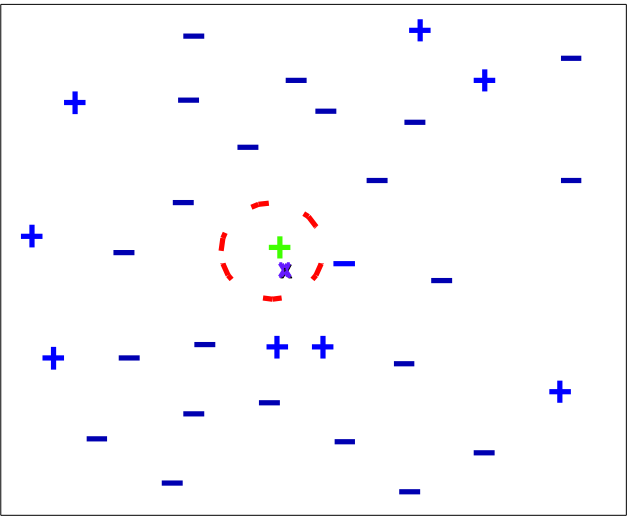
Nearest Neighbor Classification…

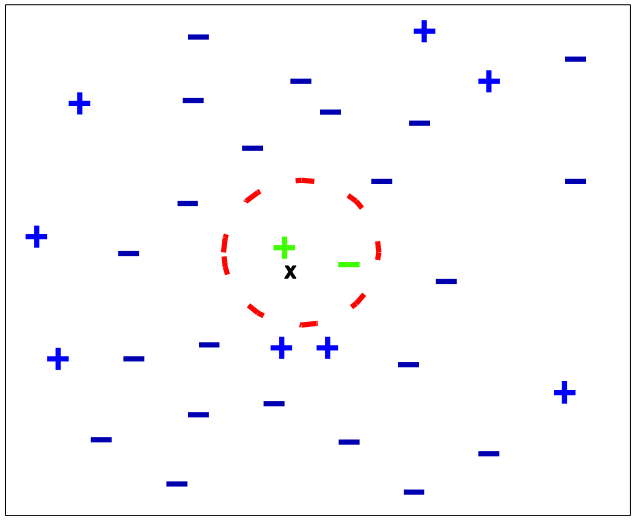
- Compute distance between two points:
  - Euclidean distance $$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$
- Problem with Euclidean measure:
  - High dimensional data
    - curse of dimensionality
  - Can produce counter-intuitive results

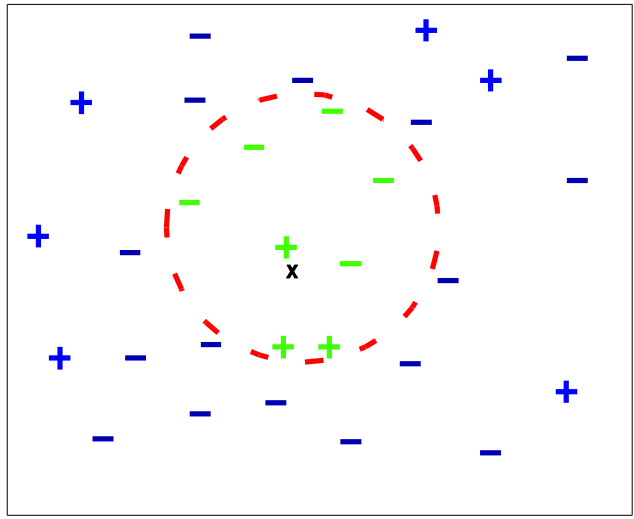
1 1 1 1 1 1 1 1 1 1 1 0		1 0 0 0 0 0 0 0 0 0 0 0
	vs	
0 1 1 1 1 1 1 1 1 1 1 1		0 0 0 0 0 0 0 0 0 0 0 1
d = 1.4142		d = 1.4142

  - Solution: Normalize the vectors to unit length

k Nearest Neighbor Classification



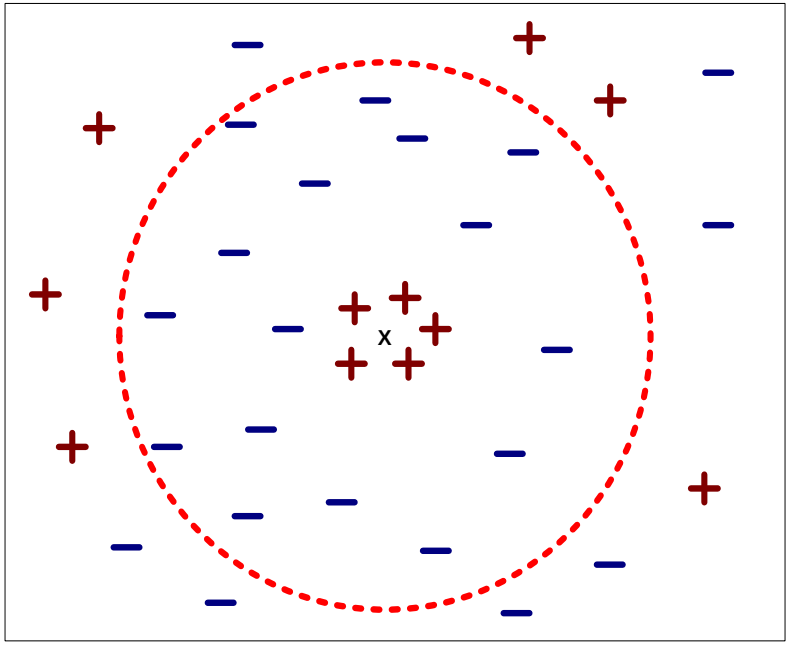




- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor Classification

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes

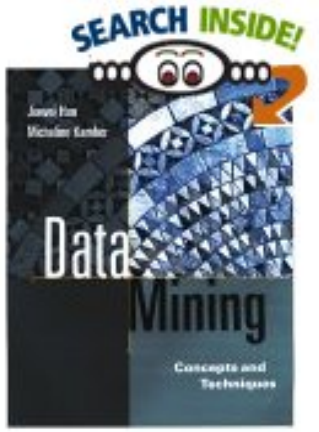


Nearest Neighbor Classification

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from $10K to $1M
  - Solution?
- Real-valued prediction for a given unknown tuple
  - Returns the mean values of the *k* nearest neighbors

Collaborative Filtering: kNN in action

Data Mining: Concepts and Techniques
by Jiawei Han (Author), Micheline Kamber (Author)



Search inside this book

List Price: $54.95
Price: **$54.95** & This item ships for **FREE with Super Saver Shipping**. See details.
Availability: Usually ships within 24 hours

Want it delivered tomorrow, March 4? Order it in the next 1 hour and 57 minutes, and choose **One-Day Shipping** at checkout. See details.

8 used & new from $44.95

Edition: Hardcover

Customers who bought this book also bought:

- Data Preparation for Data Mining: by Dorian Pyle (Author)
- The Elements of Statistical Learning: by T. Hastie, et al
- Data Mining: Introductory and Advanced Topics: by Margaret H. Dunham
- Mining the Web: Analysis of Hypertext and Semi Structured Data

Collaborative Filtering: kNN Classification/Prediction in Action

- User Perspective
  - Lots of online products, books, movies, etc.
  - Reduce my choices…please…

- Manager Perspective

"*if I have 3 million customers on the web, I should have 3 million stores on the web.*"

CEO of Amazon.com [SCH01]

Basic Approaches for Recommendation

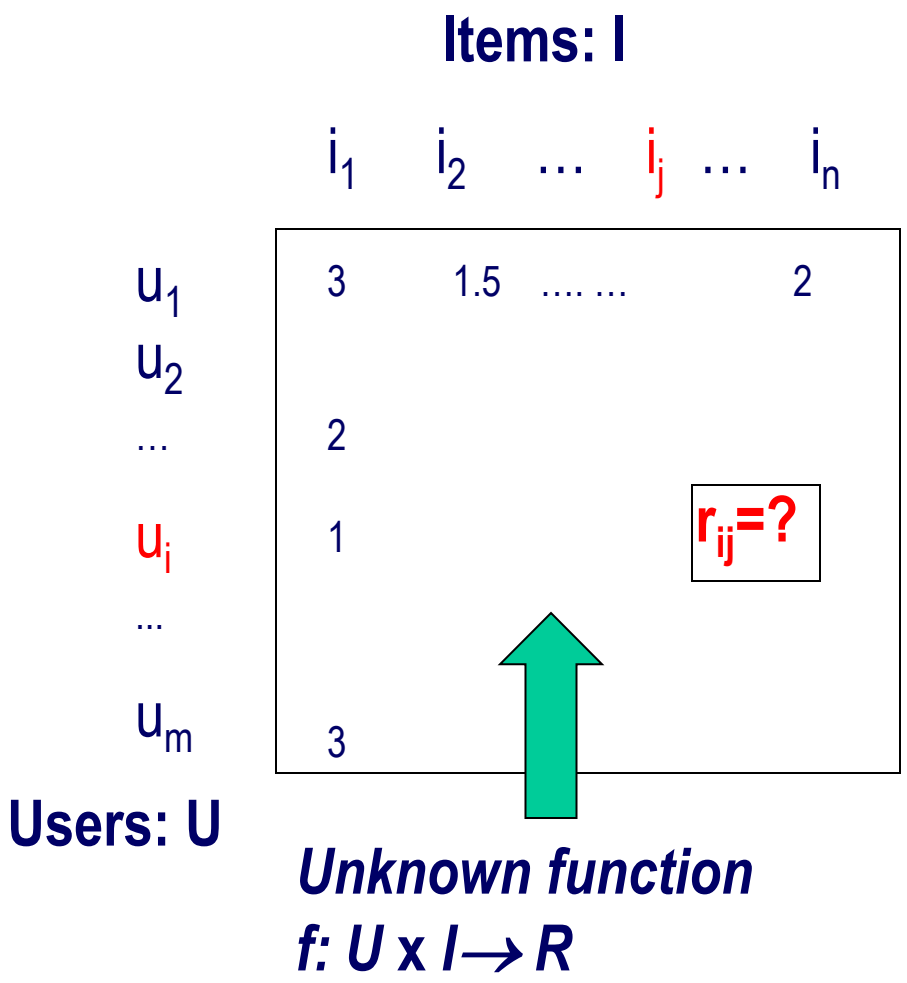
- Collaborative Filtering (CF)
  - Look at users collective behavior
  - Look at the active user history
  - Combine!

- Content-based Filtering
  - Recommend items based on key-words
  - More appropriate for information retrieval

How it Works?

- Each user has a profile
- Users rate items
  - Explicitly: score from 1..5
  - Implicitly: web usage mining
    - Time spent in viewing the item
    - Navigation path
    - Etc…
- System does the rest, How?
  - Collaborative filtering (based on kNN!)

Collaborative Filtering: A Framework



The task:

Q1: Find Unknown ratings?

Q2: Which items should we recommend to this user?

.

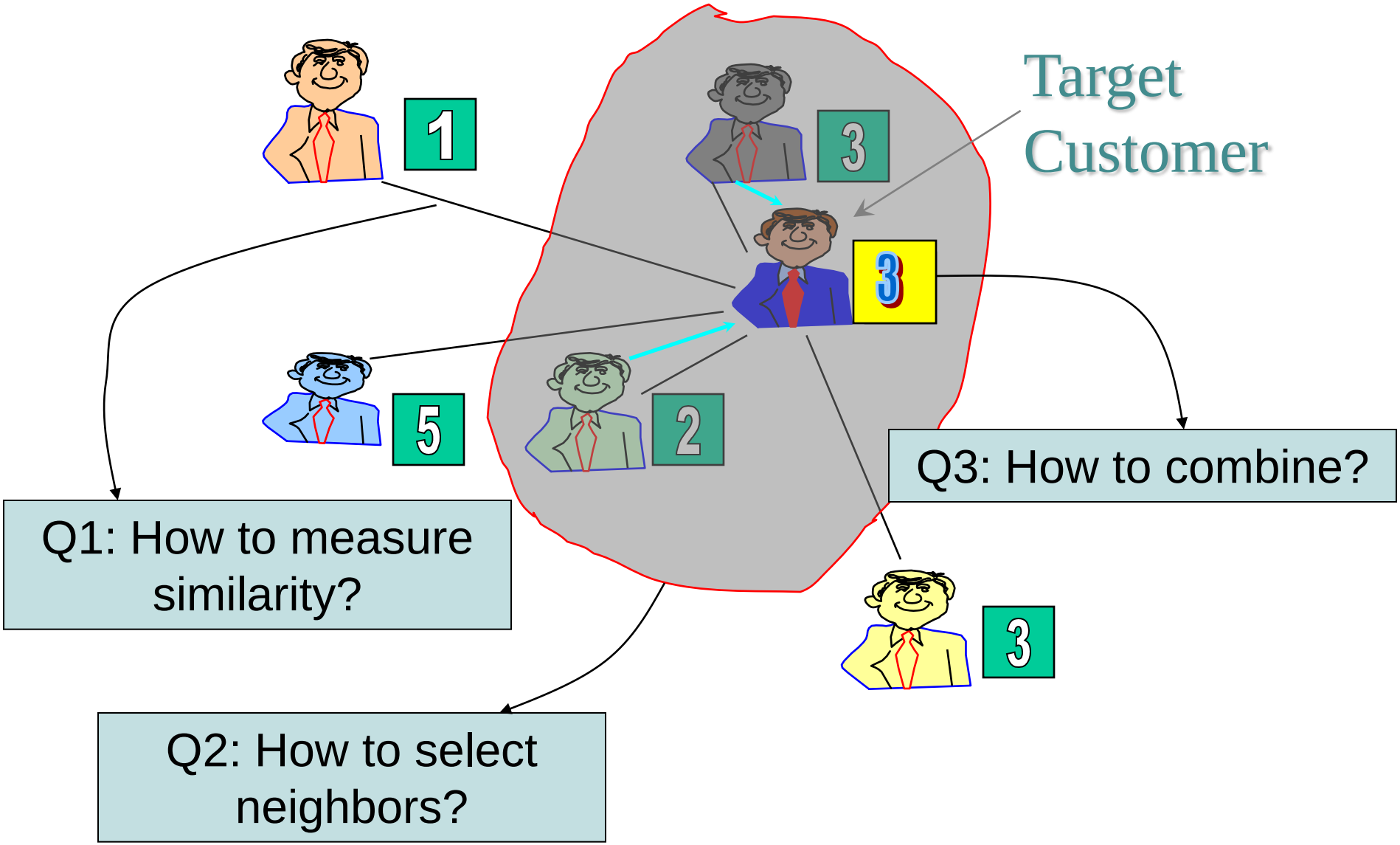
.

.

Collaborative Filtering

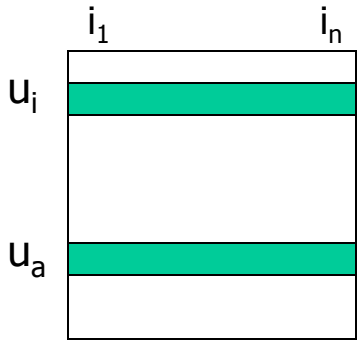
- User-User Methods
  - Identify like-minded users
  - Memory-based (heuristic): kNN
  - Model-based: Clustering, Bayesian networks
- Item-Item Method
  - Identify buying patterns
  - Correlation Analysis
  - Linear Regression
  - Belief Network
  - Association Rule Mining

User-User Similarity: Intuition



How to Measure Similarity?

- Pearson correlation coefficient



$$w_p(a,i) = \frac{\sum_{j \in \text{CommonlyRated Items}} (r_{aj} - \overline{r_a})(r_{ij} - \overline{r_i})}{\sqrt{\sum_{j \in \text{CommonlyRated Items}} (r_{aj} - \overline{r_a})^2 \sum_{j \in \text{CommonlyRated Items}} (r_{ij} - \overline{r_i})^2}}$$

- Cosine measure
  - Users are vectors in product-dimension space

$$w_c(a,i) = \frac{r_a . r_i}{\|r_a\|_2 * \|r_i\|_2}$$

Nearest Neighbor Approaches [SAR00a]

- Offline phase:
  - Do nothing…just store transactions
- Online phase:
  - Identify highly similar users to the active one
    - Best K ones
    - All with a measure greater than a threshold
- Prediction

$$r_{aj} = \bar{r}_a + \frac{\sum_i w(a,i)(r_{ij} - \bar{r}_i)}{\sum_i w(a,i)}$$

User a's neutral

User i's deviation

User a's estimated deviation

Challenges for Recommender Systems

Collaborative systems

- Scalability
- Quality of recommendations
- Dealing with new users (no history available)

Content-based systems

- False negatives
- False positives
- Limited by the features used to describe the items

Classification and Prediction

- Last lecture
  - Overview
  - Decision tree induction
  - Bayesian classification
- Today
  - Training (learning) Bayesian network
  - kNN classification and collaborative filtering
  - Support Vector Machines (SVM)
- Upcoming lectures
  - Rule based methods
  - Neural Networks
  - Regression
  - Ensemble methods
  - Model evaluation

Support Vector Machines: Overview

- A relatively new classification method for both separable and non-separable data
- Features
  - Sound mathematical foundation
  - Training time can be slow but efficient methods are being developed
  - Robust and accurate, less prone to overfitting
- Applications: handwritten digit recognition, speaker identification, …

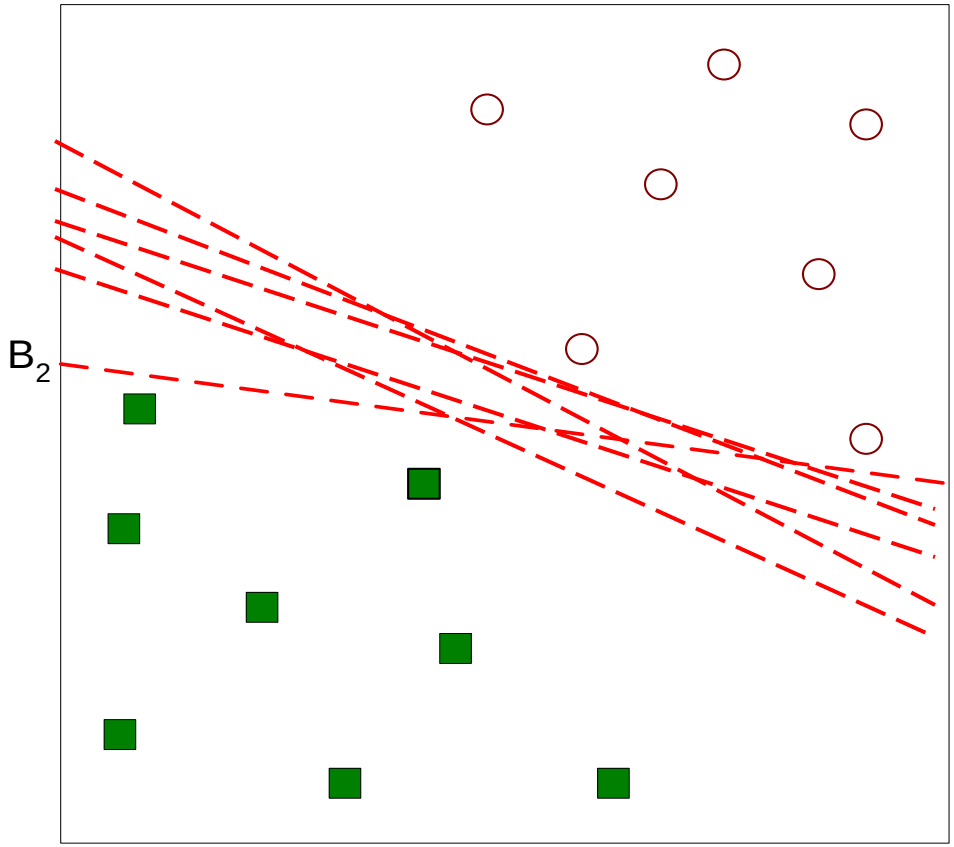
Support Vector Machines: History

- Vapnik and colleagues (1992)
  - Groundwork from Vapnik-Chervonenkis theory (1960 – 1990)
- Problems driving the initial development of SVM
  - Bias variance tradeoff, capacity control, overfitting
  - Basic idea: accuracy on the training set vs. capacity

- A Tutorial on Support Vector Machines for Pattern Recognition, Burges, Data Mining and Knowledge Discovery,1998

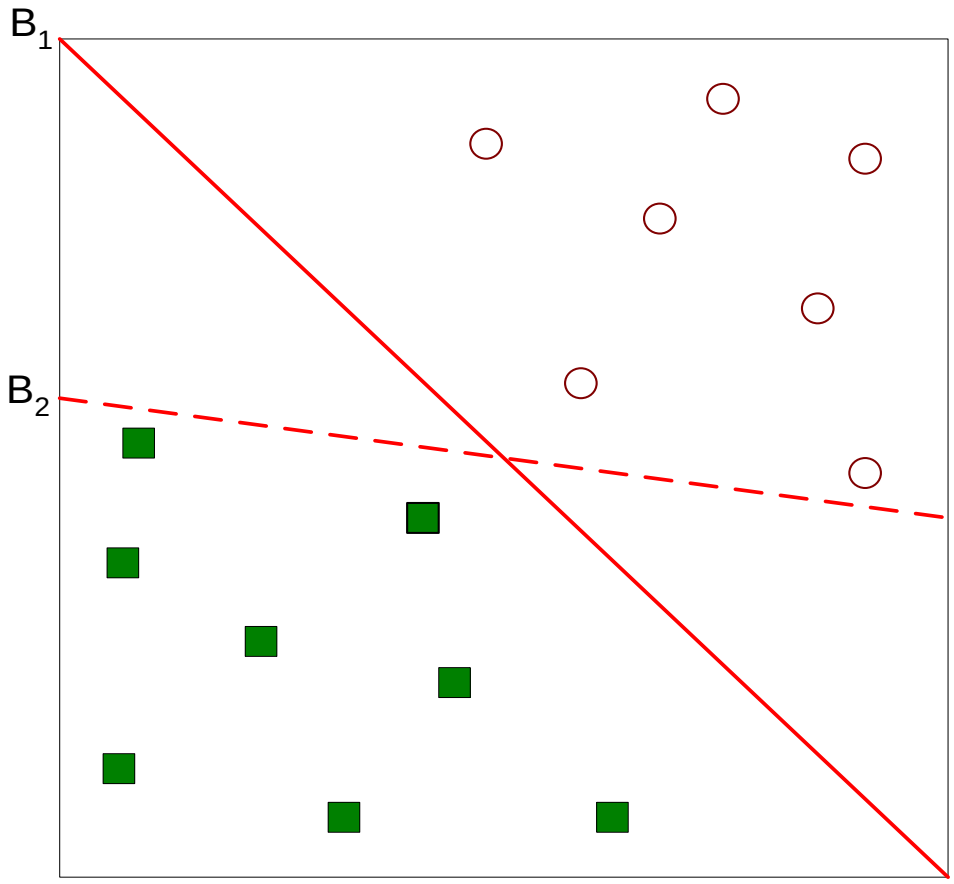
Linear Support Vector Machines

- Problem: find a linear hyperplane (decision boundary) that best separate the data

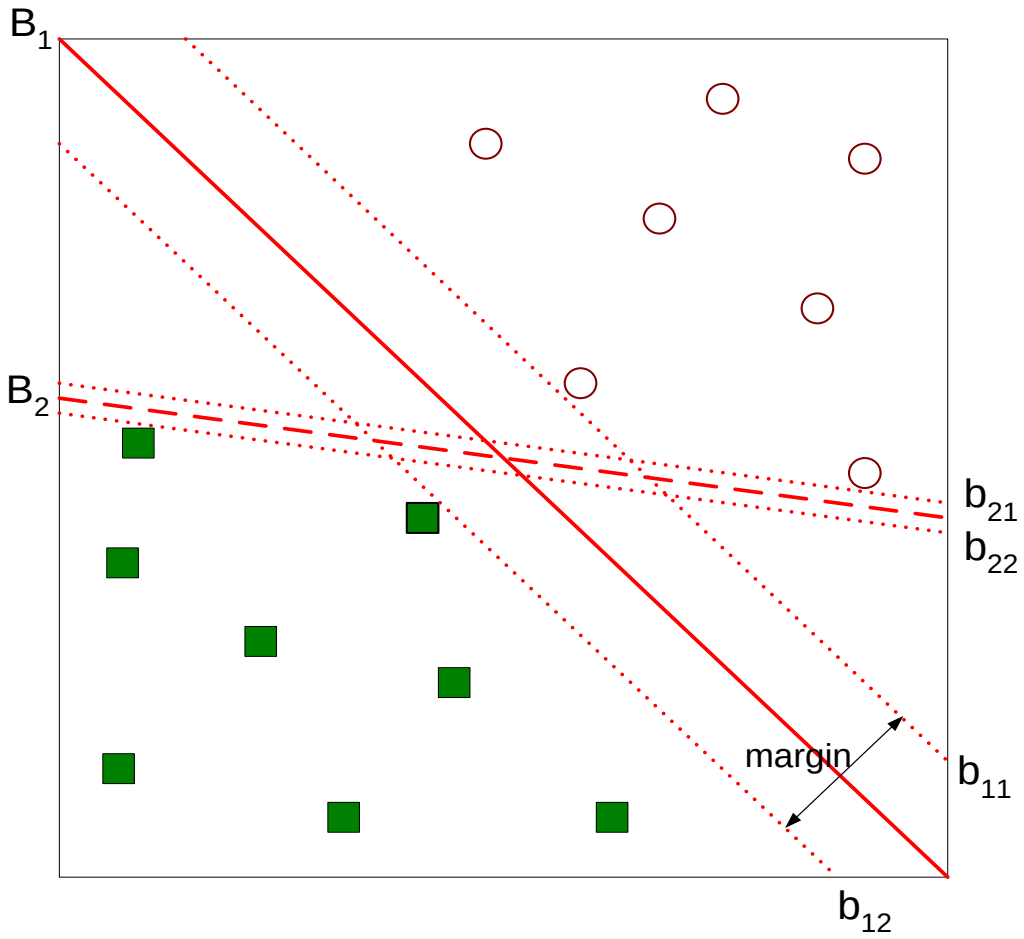


Linear Support Vector Machines

- Which line is better? B1 or B2?
- How do we define better?



Support Vector Machines



- Find hyperplane maximizes the margin

Support Vector Machines Illustration

- A separating hyperplane can be written as

  $\mathbf{W} \bullet \mathbf{X} + b = 0$

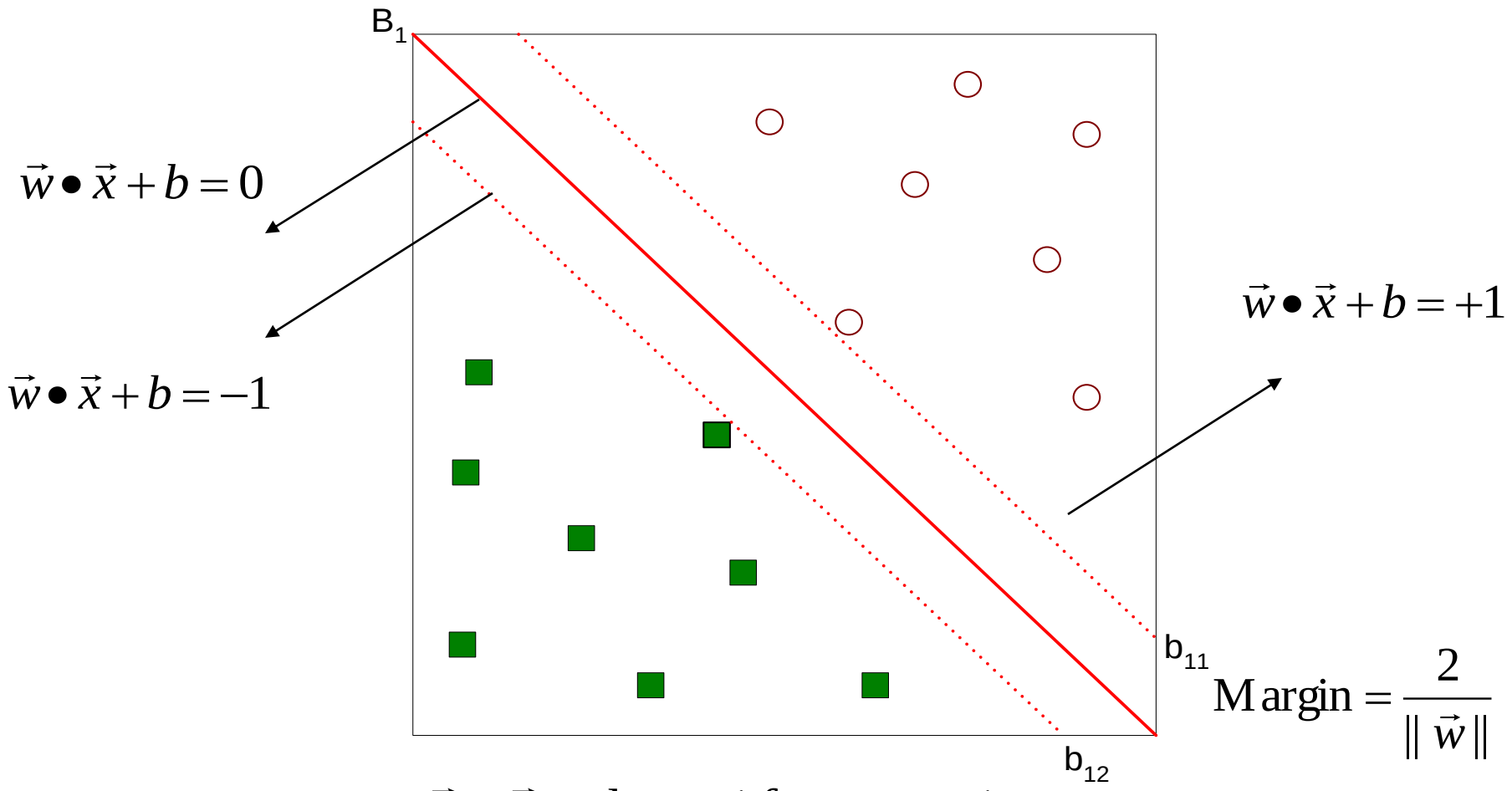
  where $\mathbf{W}=\{w_1, w_2, \ldots, w_n\}$ is a weight vector and b a scalar (bias)
- For 2-D it can be written as

  $w_0 + w_1 x_1 + w_2 x_2 = 0$
- The hyperplane defining the sides of the margin:

  $H_1: w_0 + w_1 x_1 + w_2 x_2 = 1$

  $H_2: w_0 + w_1 x_1 + w_2 x_2 = -1$
- Any training tuples that fall on hyperplanes H_1 or H_2 (i.e., the sides defining the margin) are **support vectors**

Support Vector Machines



For all training points:

$$\vec{w} \bullet \vec{x}_i + b \geq +1 \text{ for } y_i = +1$$
$$\vec{w} \bullet \vec{x}_i + b \leq -1 \text{ for } y_i = -1$$

$$\Longrightarrow \quad y_i (\vec{w} \bullet \vec{x}_i + b) - 1 \geq 0$$

Support Vector Machines

- We want to maximize: $\text{Margin} = \frac{2}{\| \vec{w} \|}$
  - Equivalent to minimizing: $\| \vec{w} \|^2$
  - But subjected to the constraints: $y_i(\vec{w} \bullet \vec{x}_i + b) - 1 \geq 0$
- Constrained optimization problem
  - Lagrange reformulation

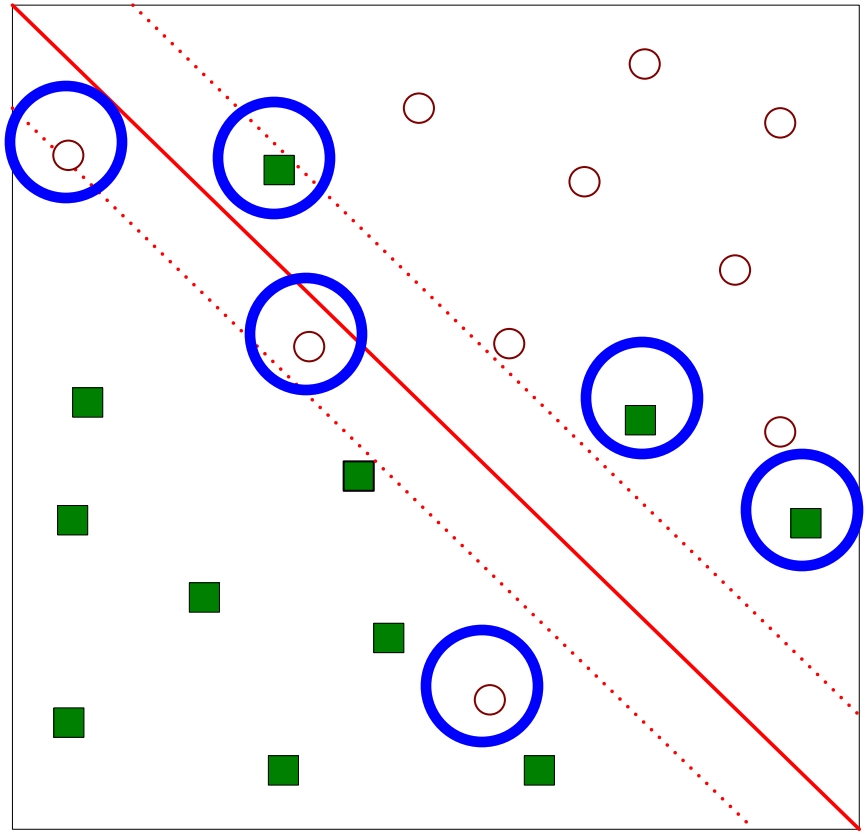
$$L_P \equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i$$

Support Vector Machines

- What if the problem is not linearly separable?
- Introduce slack variables to the constraints:

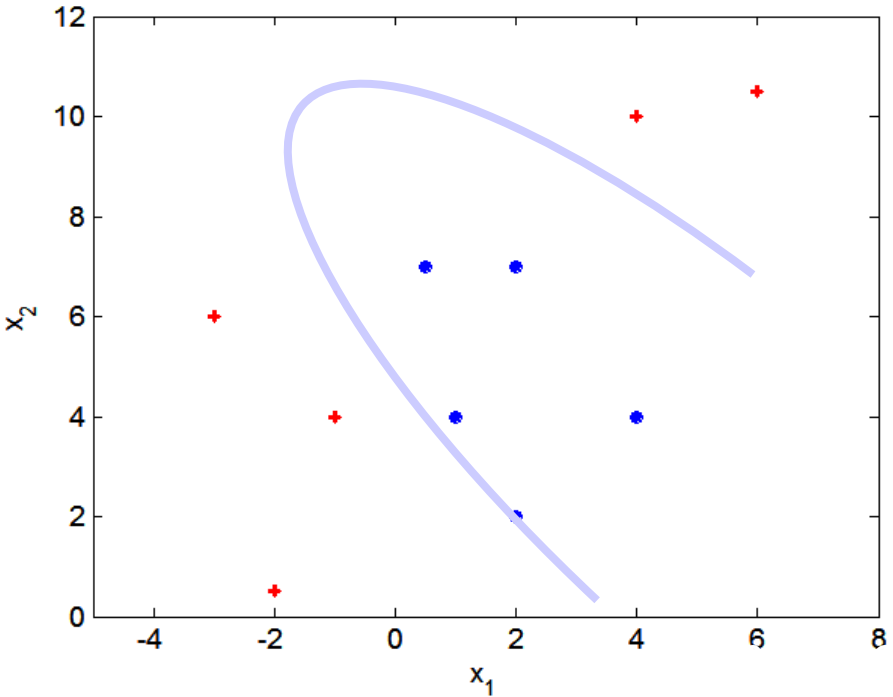
$$\vec{w} \bullet \vec{x}_i + b \geq +1 - \xi_i \text{ for } y_i = +1$$
$$\vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \text{ for } y_i = -1$$

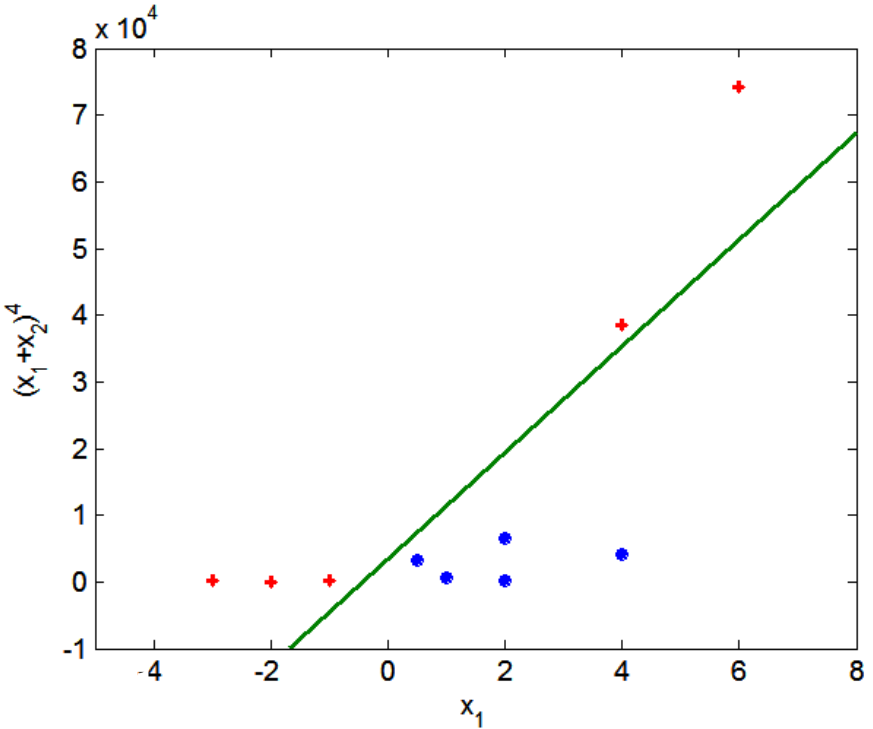
- Upper bound on the training errors: $\sum_i \xi_i$



Nonlinear Support Vector Machines

- What if decision boundary is not linear?
- Transform the data into higher dimensional space and search for a hyperplane in the new space
- Convert the hyperplane back to the original space





SVM—Kernel functions

- Instead of computing the dot product on the transformed data tuples, it is mathematically equivalent to instead applying a kernel function $K(\mathbf{X}_i, \mathbf{X}_j)$ to the original data, i.e., $K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i)\ \Phi(\mathbf{X}_j)$
- Typical Kernel Functions

$$\text{Polynomial kernel of degree } h: \quad K(X_i, X_j) = (X_i \cdot X_j + 1)^h$$

$$\text{Gaussian radial basis function kernel}: \quad K(X_i, X_j) = e^{-\|X_i - X_j\|^2/2\sigma^2}$$

$$\text{Sigmoid kernel}: \quad K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$$

- SVM can also be used for classifying multiple (> 2) classes and for regression analysis (with additional user parameters)

Support Vector Machines: Comments and Research Issues

- Robust and accurate with nice generalization properties
- Effective (insensitive) to high dimensions - Complexity characterized by # of support vectors rather than dimensionality
- Scalability in training - While the speed in test phase is largely solved, training for very large datasets is an unsolved problem.
- Extension to regression analysis
- Extension to multiclass SVM – still in research
- Kernel selection – still in research

SVM Related Links

- SVM web sites
  - www.kernel-machines.org
  - www.kernel-methods.net
  - www.support-vector.net
  - www.support-vector-machines.org
- Representative implementations
  - LIBSVM: an efficient implementation of SVM, multi-class classifications
  - SVM-light: simpler but performance is not better than LIBSVM, support only binary classification and only C language
  - SVM-torch: another recent implementation also written in C.

SVM—Introduction Literature

- "Statistical Learning Theory" by Vapnik: extremely hard to understand, containing many errors too
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
  - Better than the Vapnik's book, but still written too hard for introduction, and the examples are not-intuitive
- The book "An Introduction to Support Vector Machines" by N. Cristianini and J. Shawe-Taylor
  - Also written hard for introduction, but the explanation about the Mercer's theorem is better than above literatures
- The neural network book by Haykins
  - Contains one nice chapter of SVM introduction

Classification and Prediction

- Last lecture
  - Overview
  - Decision tree induction
  - Bayesian classification
- Today
  - Training (learning) Bayesian network
  - kNN classification and collaborative filtering
  - Support Vector Machines (SVM)
- Upcoming lectures
  - Rule based methods
  - Neural Networks
  - Regression
  - Ensemble methods
  - Model evaluation