

Chapter 8. Classification: Basic Concepts

- ❑ Classification: Basic Concepts
- ❑ Decision Tree Induction
- ❑ Bayes Classification Methods
- ❑ Linear Classifier
- ❑ Model Evaluation and Selection
- ❑ Techniques to Improve Classification Accuracy: Ensemble Methods
- ❑ Additional Concepts on Classification
- ❑ Summary



What Is Bayesian Classification?

- A statistical classifier
 - Perform *probabilistic prediction* (*i.e.*, predict class membership probabilities)
- Foundation—Based on Bayes' Theorem
- Performance
 - A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental
 - Each training example can incrementally increase/decrease the probability that a hypothesis is correct—prior knowledge can be combined with observed data
- Theoretical Standard
 - Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayes' Theorem: Basics

- Total probability Theorem:

$$p(B) = \sum_i p(B|A_i)p(A_i)$$

- Bayes' Theorem:

$$p(H|X) = \frac{p(X|H)p(H)}{p(X)}$$

posteriori probability

What we should choose

likelihood

What we just see

when you know

about feature X

Statistic

Prob of feature X
when you know
about n statistic

prior probability

What we knew previously

On prob yes/no 277117

- X: a data sample ("evidence")

Prediction can be done based on Bayes' Theorem:

- H: X belongs to class C

Classification is to derive the maximum posteriori

Bayes' Theorem: Basics

- Total probability Theorem:

$$p(B) = \sum_i p(B|A_i)p(A_i)$$

- Bayes' Theorem:

$$p(H|X) = \frac{p(X|H)p(H)}{p(X)} \propto p(X|H) p(H)$$

posteriori probability

What we should choose

likelihood

What we just see

prior probability

What we knew previously

- X : a data sample ("evidence")
- H : X belongs to class C

Prediction can be done based on Bayes' Theorem:

Classification is to derive the maximum posteriori

Naïve Bayes Classifier: Making a Naïve Assumption

ନେଟ୍‌ବେଳେ କଥିତ ହେଉଥାଏ ପରିମାଣିତ କରିବାକୁ ପରିଚାରିତ କରିବାକୁ ପରିଚାରିତ କରିବାକୁ

- Practical difficulty of Naïve Bayes inference: It requires initial knowledge of many probabilities, which may not be available or involving significant computational cost
- A Naïve Special Case
 - Make an additional assumption to simplify the model, but achieve comparable performance.

attributes are conditionally independent
(i.e., no dependence relation between attributes)

$$p(X|C_i) = \prod_k p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots p(x_n|C_i)$$

- Only need to count the class distribution w.r.t. features

Naïve Bayes Classifier: Categorical vs. Continuous Valued Features

- If feature x_k is categorical, $p(x_k = v_k | C_i)$ is the # of tuples in C_i with $x_k = v_k$, divided by $|C_{i,D}|$ (# of tuples of C_i in D)

$$p(X|C_i) = \prod_k p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots p(x_n|C_i)$$

- If feature x_k is continuous-valued, $p(x_k = v_k | C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$p(x_k = v_k | C_i) = N(x_k | \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma_{C_i}} e^{-\frac{(x-\mu_{C_i})^2}{2\sigma^2}}$$

AIH0121

Naïve Bayes Classifier: Training Dataset

INPUT Age 31-40, Income = high, Student = yes, Credit_rating = fair

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30, Income = medium,

Student = yes, Credit_rating = Fair)

H | X

$$\text{Prob (buy?|X)} = P(X|buy) P(buy)$$

$\xleftarrow{\quad} \vec{x} \xrightarrow{\quad} \xleftarrow{\quad} y \xrightarrow{\quad}$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

9/14

Naïve Bayes Classifier: Training Dataset

QUIZ 8

input → Data to be classified

$$x = (\text{Age} = 31 \dots 40, \text{income} = \text{high}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$$

age	income	student	credit rating	buys computer
<=30	high	no	fair	no /
<=30	high	no	excellent	no /
31...40	high	no	fair	yes / /
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes /
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes /
31...40	high	yes	fair	yes /
>40	medium	no	excellent	no

Class:

C1: buys_computer = 'yes'

C2: buys_computer = 'no'

$$P(C_1): P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$$

$$P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$$

$$\frac{0+1}{5+3} \xrightarrow{\downarrow \text{age}} \begin{cases} <=30 \\ 31\dots40 \\ >40 \end{cases} = \frac{1}{8}$$

Wet: 2/12
/ 3

Compute $P(X|C_i)$ for each class

$$P(\text{age} = \text{"31...40"} | \text{buys_computer} = \text{"yes"}) = (4+1)/9 = 5/9 = 0.555$$

$$P(\text{age} = \text{"31...40"} | \text{buys_computer} = \text{"no"}) = 0/5 = 0 \rightarrow 1/5 = 0.2 \nearrow \frac{1}{8}$$

$$P(\text{income} = \text{"high"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{income} = \text{"high"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

$$P(\text{student} = \text{"yes"} | \text{buys_comper} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} | \text{buys_comper} = \text{"no"}) = 1/5 = 0.2$$

$$P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

$X = (\text{Age} = 31 \dots 40, \text{income} = \text{high}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

$$P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) = 0.555 \times 0.222 \times 0.667 \times 0.667 = 0.055$$

$$P(X|\text{buys_computer} = \text{"no"}) = 0.2 \times 0.4 \times 0.2 \times 0.4 = 0.006$$

$$\begin{aligned} P(X|C_i) * P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) \\ = 0.055 \times 0.643 = 0.035 \quad \checkmark \end{aligned}$$

$$\begin{aligned} P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) \\ = 0.006 \times 0.357 = 0.002 \end{aligned}$$

Therefore, X belongs to class ("buys_computer = "yes")

643021266-5 ຖະໜາດຕິພາບ ອົງກອນ

Naïve Bayes Classifier: An Example

- $P(C_i)$: $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
- $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$

- Compute $P(X|C_i)$ for each class

$$\begin{aligned}
 P(\text{age} = \text{"}<=30\text{"} | \text{buys_computer} = \text{"yes"}) &= 2/9 = 0.222 \\
 P(\text{age} = \text{"}<= 30\text{"} | \text{buys_computer} = \text{"no"}) &= 3/5 = 0.6 \\
 P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) &= 4/9 = 0.444 \\
 P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) &= 2/5 = 0.4 \\
 P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) &= 6/9 = 0.667 \\
 P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) &= 1/5 = 0.2 \\
 P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) &= 6/9 = 0.667 \\
 P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) &= 2/5 = 0.4
 \end{aligned}$$

- $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

$$\begin{aligned}
 P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) &= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044 \\
 P(X|\text{buys_computer} = \text{"no"}) &= 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019
 \end{aligned}$$

$$\begin{aligned}
 P(X|C_i)*P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) &= 0.028 \\
 P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) &= 0.007
 \end{aligned}$$

Therefore, X belongs to class ("buys_computer = yes")

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional probability be **non-zero**
 - Otherwise, the predicted probability will be zero

$$p(X|C_i) = \prod_k p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots p(x_n|C_i)$$

- Example. Suppose a dataset with 1000 tuples:
 - income = low (0), income = medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
 - *Adding 1 to each case*

$$\text{Prob}(\text{income} = \text{low}) = 1/(1000 + 3)$$

$$\text{Prob}(\text{income} = \text{medium}) = (990 + 1)/(1000 + 3)$$

$$\text{Prob}(\text{income} = \text{high}) = (10 + 1)/(1000 + 3)$$

- The “corrected” probability estimates are close to their “uncorrected” counterparts

Naïve Bayes Classifier: Strength vs. Weakness

- ❑ Strength
 - ❑ Easy to implement
 - ❑ Good results obtained in most of the cases
- ❑ Weakness
 - ❑ Assumption: attributes conditional independence, therefore loss of accuracy
 - ❑ Practically, dependencies exist among variables
 - ❑ E.g., Patients: Profile: age, family history, etc.
 - ❑ Symptoms: fever, cough etc.
 - ❑ Disease: lung cancer, diabetes, etc.
 - ❑ Dependencies among these cannot be modeled by Naïve Bayes Classifier
- ❑ How to deal with these dependencies?
 - ❑ Use Bayesian Belief Networks (to be covered in the next chapter)

Chapter 8. Classification: Basic Concepts

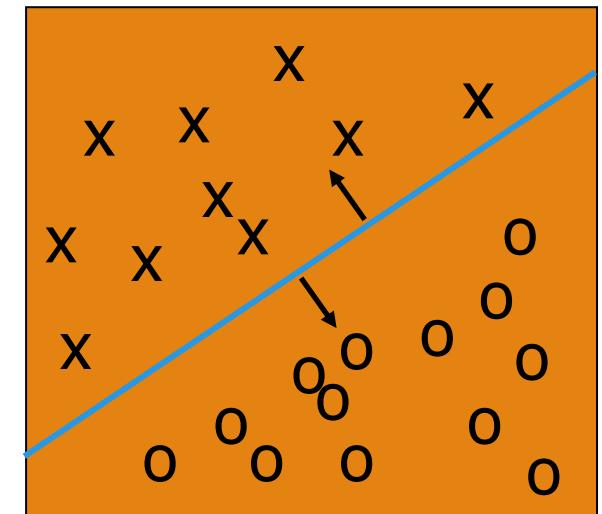
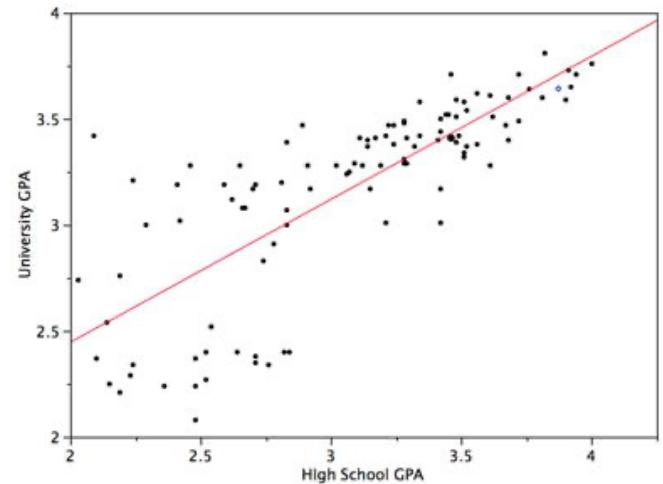
- ❑ Classification: Basic Concepts
- ❑ Decision Tree Induction
- ❑ Bayes Classification Methods
- ❑ Linear Classifier
- ❑ Model Evaluation and Selection
- ❑ Techniques to Improve Classification Accuracy: Ensemble Methods
- ❑ Additional Concepts on Classification
- ❑ Summary



Linear Regression vs. Linear Classifier

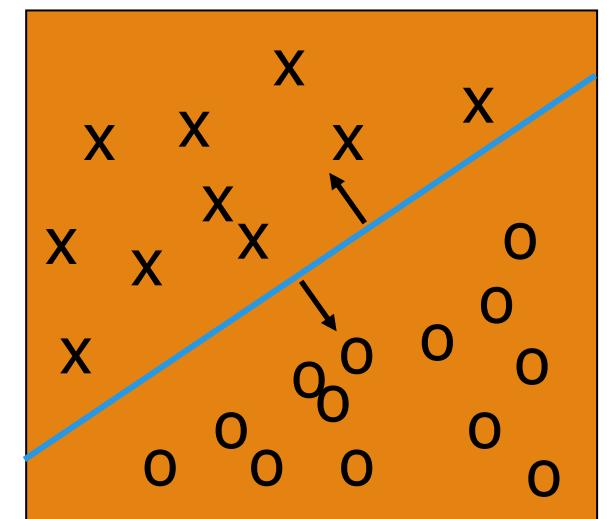
- ❑ Linear regression
 - ❑ Data modeled to fit a straight line
 - ❑ *Linear equation:* $Y = w X + b$
 - ❑ Often uses the least-square method to fit the line
 - ❑ Used to predict continuous values

- ❑ Linear Classifier
 - ❑ Built a classification model using a straight line
 - ❑ Used for (categorical data) binary classification



Linear Classifier: General Ideas

- Binary Classification
- $f(x)$ is a linear function based on the example's attribute values
 - The prediction is based on the value of $f(x)$
 - Data above the blue line belongs to class 'x' (i.e., $f(x) > 0$)
 - Data below blue line belongs to class 'o' (i.e., $f(x) < 0$)
- Classical Linear Classifiers
 - Linear Discriminant Analysis (LDA) (not covered)
 - Logistic Regression
 - Perceptron (later)
 - SVM (later)

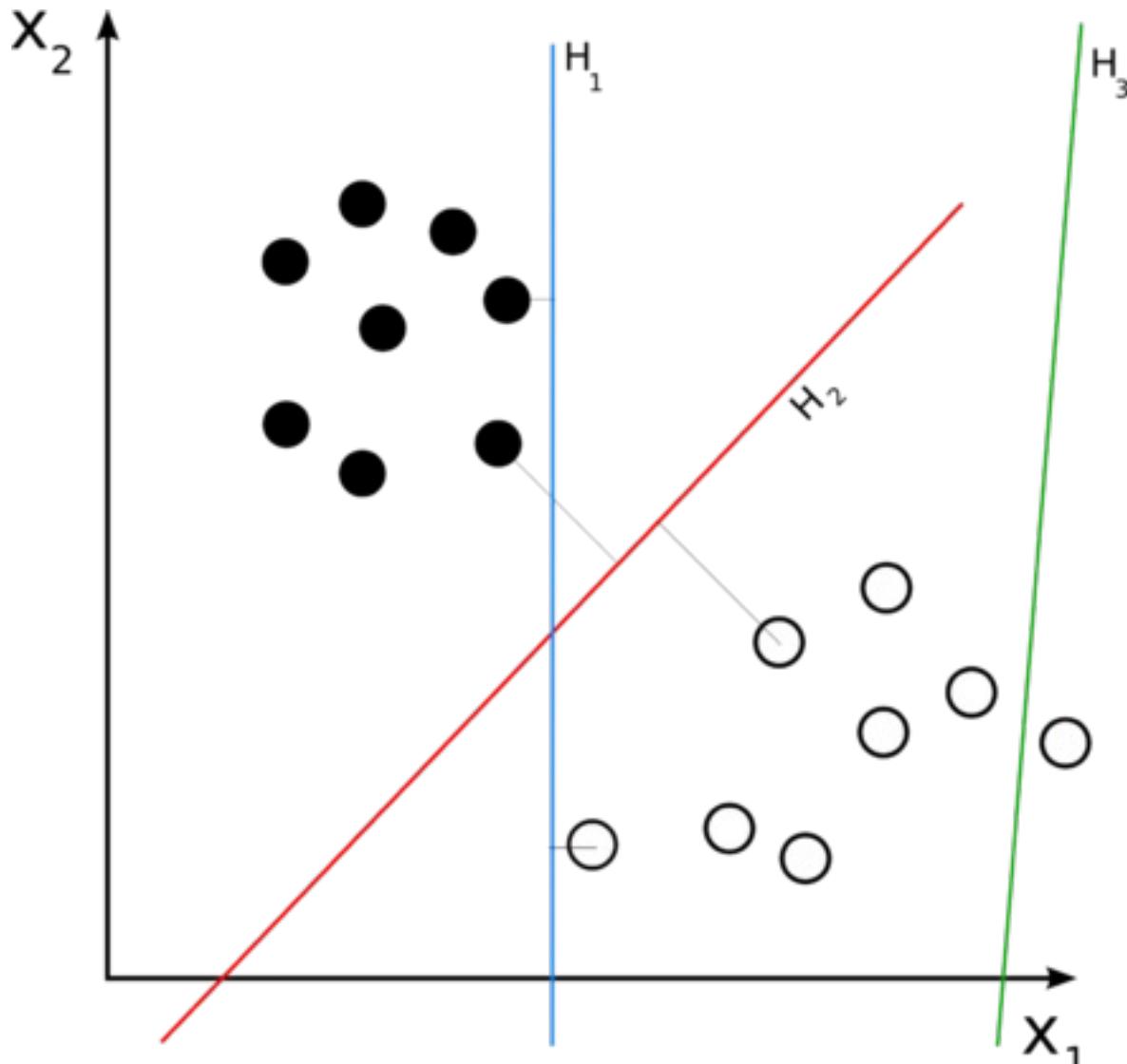


Linear Classifier: An Example

- A toy rule to determine whether a faculty member has tenure
 - Year ≥ 6 or Title = “Professor” \Leftrightarrow Tenure
- How to express the rule as a linear classifier?
- Features
 - $x_1 (x_1 \geq 0)$ is an integer denoting the year
 - x_2 is a Boolean denoting whether the title is “Professor”
- A feasible linear classifier: $f(x) = (x_1 - 5) + 6 \cdot x_2$
 - When x_2 is True, because $x_1 \geq 0$, $f(x)$ is always greater than 0
 - When x_2 is False, because $f(x) > 0 \Leftrightarrow x_1 \geq 6$
- There are many more feasible classifiers
 - $f(x) = (x_1 - 5.5) + 6 \cdot x_2$
 - $f(x) = 2 \cdot (x_1 - 5) + 11 \cdot x_2$
 -

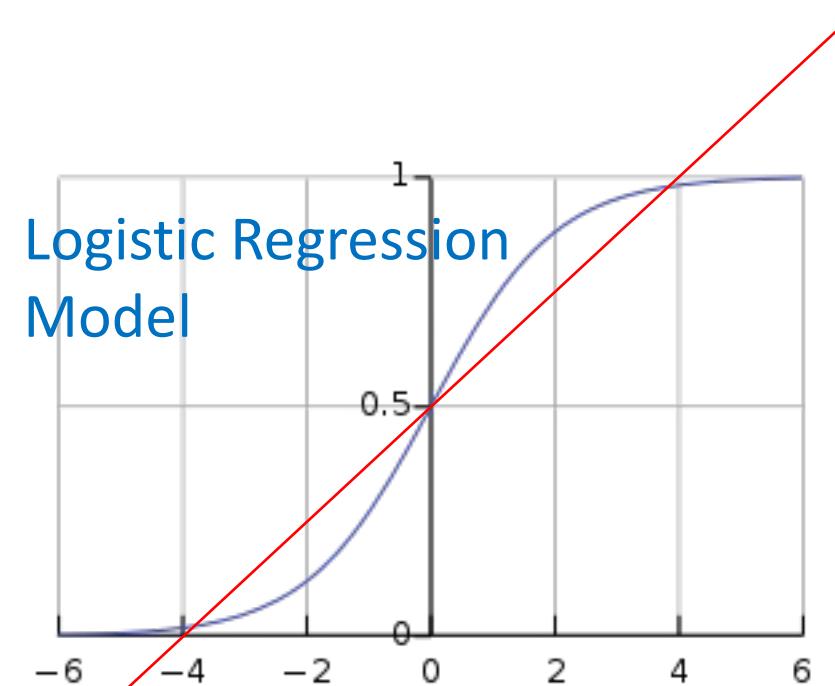
Key Question: Which Line Is Better?

- ❑ There might be many feasible linear functions
 - ❑ Both H_1 and H_2 will work
- ❑ Which one is better?
 - ❑ H_2 looks “better” in the sense that it is also furthest from both groups
- ❑ We will introduce more in the SVM section



Logistic Regression: General Ideas

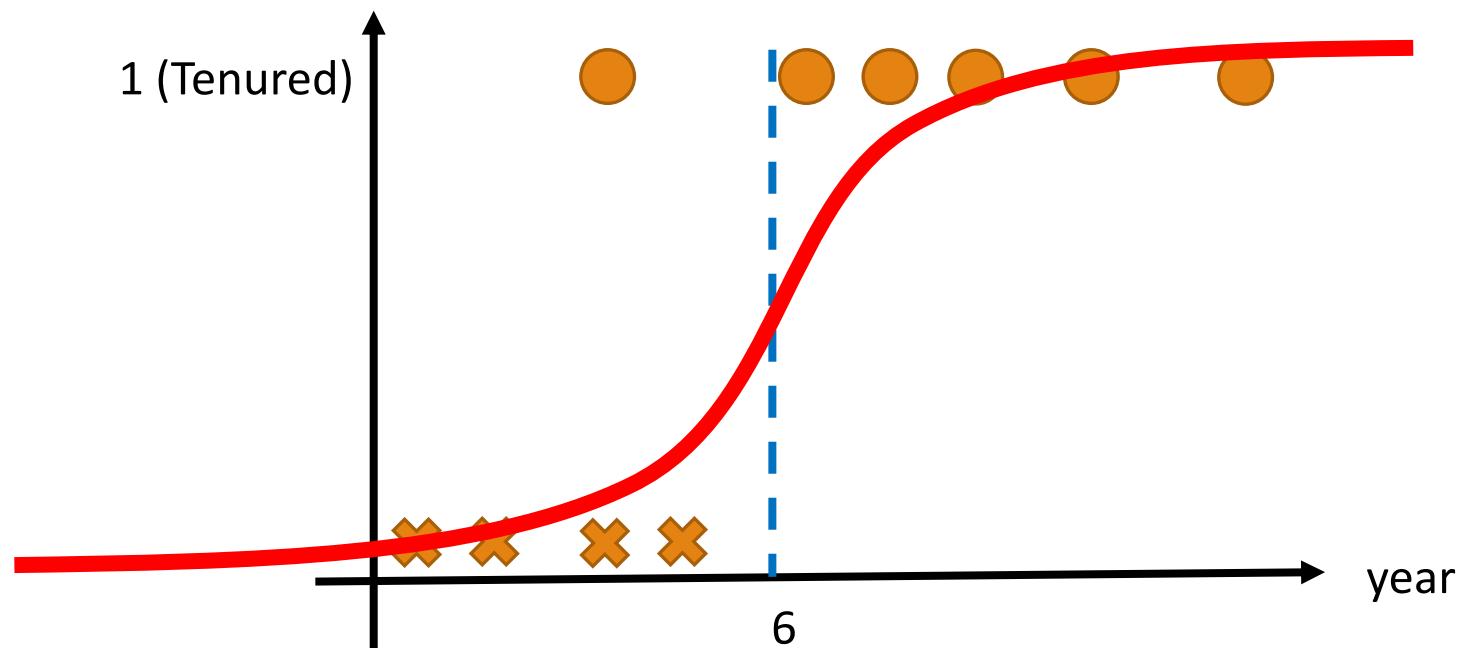
- Key Idea: Turns linear predictions into probabilities
- Sigmoid function:
 - $S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$
 - Projects $(-\infty, +\infty)$ to $[0, 1]$
- Compare to linear probability model
- More smooth



Linear Probability
Model

Logistic Regression: An Example

- Suppose we only consider the year as feature



Logistic Regression: Maximum Likelihood

- ❑ The prediction function to learn

- ❑ $p(Y = 1 | X = x; \mathbf{w}) = S(w_0 + \sum_{i=1}^n w_i \cdot x_i)$
 - ❑ $\mathbf{w} = (w_0, w_1, w_2, \dots, w_n)$ are the parameters

- ❑ Maximum Likelihood

- ❑ Log likelihood:

$$l(\mathbf{w}) = \sum_{i=1}^N y_i \log p(Y = 1 | X = x_i; \mathbf{w}) + (1 - y_i) \log(1 - p(Y = 1 | X = x_i; \mathbf{w}))$$

- ❑ There's no close form solution

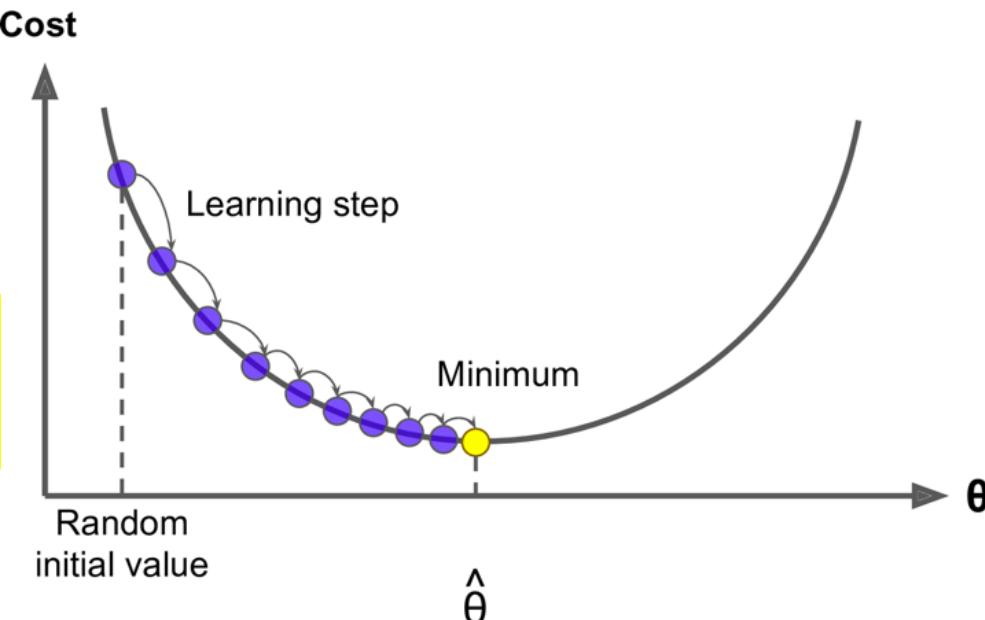
- ❑ Gradient Descent
 - ❑ Update \mathbf{w} based on training data
 - ❑ Chain-rule for the gradient

Gradient Descent

- Gradient Descent is an iterative optimization algorithm for finding the minimum of a function (e.g., the negative log likelihood)
- For a function $F(x)$ at a point a , $F(x)$ decreases fastest if we go in the direction of the negative gradient of a

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

When the gradient is zero, we arrive at the local minimum



Generative vs. Discriminative Classifiers

- X: observed variables (features)
- Y: target variables (class labels)
- A generative classifier models $p(Y, X)$
 - It models how the data was "generated"? "what is the likelihood this or that class generated this instance?" and pick the one with higher probability
- Naïve Bayes
- Bayesian Networks
- A discriminative classifier models $p(Y|X)$
 - It uses the data to create a decision boundary
- Logistic Regression
- Support Vector Machines

Further Comments on Discriminative Classifiers

- Strength
 - Prediction accuracy is generally high
 - As compared to generative models
 - Robust, works when training examples contain errors
 - Fast evaluation of the learned target function
 - Comparing to [\(covered in future\)](#) Bayesian networks (which are normally slow)
- Criticism
 - Long training time
 - Difficult to understand the learned function (weights)
 - Bayesian networks can be used easily for pattern discovery
 - Not easy to incorporate domain knowledge
 - Easy in the form of priors on the data or distributions

Chapter 8. Classification: Basic Concepts

- ❑ Classification: Basic Concepts
- ❑ Decision Tree Induction
- ❑ Bayes Classification Methods
- ❑ Linear Classifier
- ❑ Model Evaluation and Selection
- ❑ Techniques to Improve Classification Accuracy: Ensemble Methods
- ❑ Additional Concepts on Classification
- ❑ Summary



Model Evaluation and Selection

- ❑ Evaluation metrics
 - ❑ How can we measure accuracy?
 - ❑ Other metrics to consider?
 - ❑ Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
 - ❑ Methods for estimating a classifier's accuracy
 - ❑ Holdout method
 - ❑ Cross-validation
 - ❑ Bootstrap
 - ❑ Comparing classifiers:
 - ❑ ROC Curves
- ✓ import
✓ define
✓ train
✓ test
- 4 Decision tree
- leave 1 out

ມີຫຼາຍກວມພົນບໍ່

Classifier Evaluation Metrics: Confusion Matrix

- Confusion Matrix:

Actual class \ Predicted class	C_1 ພົນ C_1	$\neg C_1$ ພົນ $\neg C_1$
C_1 ລົງ + $\neg C_1$	True Positives (TP) \rightarrow ຢູ່ນັ້ນ ນັ້ນ / ກຸ່ມ \rightarrow Row / column	False Negatives (FN)
$\neg C_1$ ລົງ - C_1	False Positives (FP)	True Negatives (TN)

- In a confusion matrix w. m classes, $CM_{i,j}$ indicates # of tuples in class i that were labeled by the classifier as class j
- May have extra rows/columns to provide totals
- Example of Confusion Matrix:

Actual class \ Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Confusion Matrix

- ທ່ານ
□ Confusion Matrix: → ແກແນວບີ / ຕົ້ນ → Row / column
ຈະນຸ່ມ ເປັນ ແຕ່ ມັນ - ສິນໄປ F

Actual class \ Predicted class	\oplus C_1 ເປັນ c_1	\ominus C_1 ມີ້ນຢູ່ບັນດາ c_1
ຈະນຸ່ມ \oplus C_1 Class c_1	True Positives (TP)	False Negatives (FN)
\ominus C_1 ມີ້ນຢູ່ບັນດາ c_1	False Positives (FP)	True Negatives (TN)

- In a confusion matrix w. m classes, $CM_{i,j}$ indicates # of tuples in class i that were labeled by the classifier as class j
 □ May have extra rows/columns to provide totals
 □ Example of Confusion Matrix:
- ນັບໃຈທີ່ເຫັນ predict
 ຈຳນານຫັນວັນປັບປຸງ
 ດຳຕົວບັນດາ
 ເປັນ c_1
 ແລ້ວ ດຳນວນເປັນ c_1
- ຈະນຸ່ມ -
 ແຕ່ເປັນ -
 ຂ້າວັບ True

Actual class \ Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

Ground Truth

Actual	Predicted
N	N
P	P
P	N
P	P
N	P
N	N
N	P
P	N
P	P
P	P

TN ✗
TP ✗
FN ✗
TP ✗
FP ✗
TN ✗
FP ✗
FN ✗
TP ✗
TP ✗

Actual class\ Predicted class	yes	no	Total
yes	TP = 4	FN = 2	P = 6
no	FP = 2	TN = 2	N = 4
Total	P' = 6	N' = 4	10

$$\text{Accuracy} = \frac{(TP+TN)}{\text{All}} = \frac{(4+2)}{10} = \underline{\underline{\frac{6}{10}}}$$

$$\text{Precision} = P = \frac{TP}{TP+FP} = \frac{4}{4+2} = \frac{4}{6} = 0.67$$

$$\text{Recall} = R = \frac{TP}{TP+FN} = \frac{4}{4+2} = \frac{4}{6} = 0.67$$

Այս առավելի մասնաւուն

$$\text{Sensitivity} = \frac{TP}{P} = \frac{4}{6} = \frac{2}{3} = 0.67$$

Առողջապահ

$$\text{Specificity} = \frac{TN}{N} = \frac{2}{4} = \frac{1}{2} = 0.5$$

F1-measure

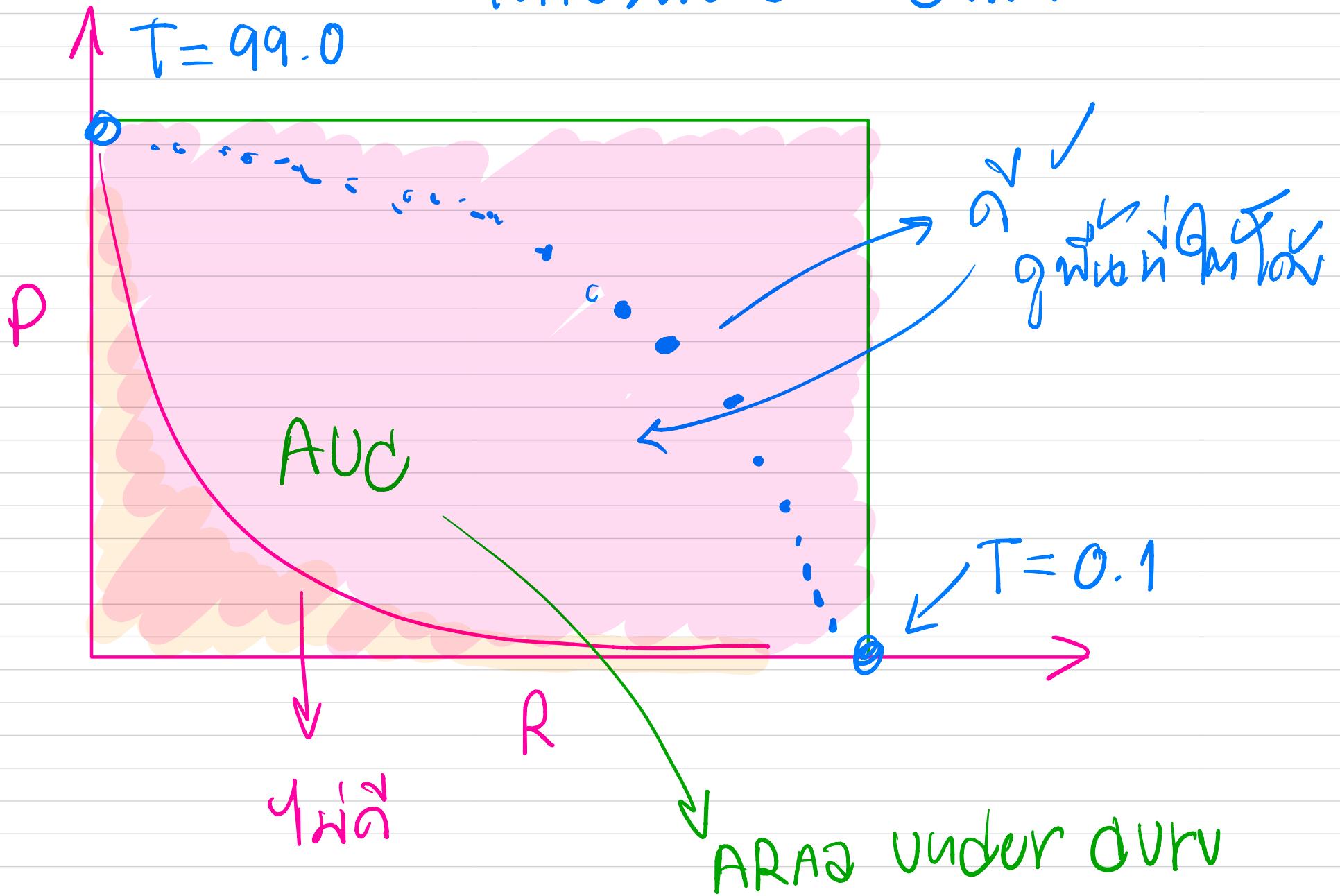
$$F_1 = \frac{2PR}{P+R} = \frac{2(0.67)(0.67)}{0.67 + 0.67} = 0.67$$

Ground Truth
(ମୂଳ ମୁଖ୍ୟ)
Threshold
ମୂଲ୍ୟରେଣ୍ଟ୍ସା

Actual	Prob.	Predicted
N	0.4	N
P	0.6	P
P	0.3	N
P	0.9	P
N	0.55	P
N	0.1	N
N	0.7	P
P	0.4	N
P	0.9	P
P	0.9	P

TN ✅
TP ✅
FN ✅
TP ✅
FP ✅
TN ✅
FP ✅
FN ✅
TP ✅
TP ✅

Threshold $\rightarrow 0 \dots 100$



Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	-C	
C	TP	FN	P
-C	FP	TN	N
	P'	N'	All

- Classifier accuracy, or recognition rate
 - Percentage of test set tuples that are correctly classified
- Accuracy = $(TP + TN)/All$**
- Error rate: $1 - accuracy$, or
- Error rate = $(FP + FN)/All$**

- Class imbalance problem
 - One class may be *rare*
 - E.g., fraud, or HIV-positive
 - Significant *majority of the negative class* and minority of the positive class
 - Measures handle the class imbalance problem
- **Sensitivity** (recall): True positive recognition rate
 - **Sensitivity = TP/P**
- **Specificity**: True negative recognition rate
 - **Specificity = TN/N**

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- ❑ **Precision:** Exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$P = \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- ❑ **Recall:** Completeness: what % of positive tuples did the classifier label as positive?

$$R = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- ❑ Range: $[0, 1]$
- ❑ The “inverse” relationship between precision & recall
- ❑ **F measure (or F-score):** harmonic mean of precision and recall
- ❑ In general, it is the weighted measure of precision & recall

$$F_{\beta} = \frac{1}{\alpha \cdot \frac{1}{P} + (1 - \alpha) \cdot \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

Assigning β times as much weight to recall as to precision)

- ❑ **F1-measure (balanced F-measure)**

- ❑ That is, when $\beta = 1$,

$$F_1 = \frac{2PR}{P + R}$$

Classifier Evaluation Metrics: Example

- Use the same confusion matrix, calculate the measure just introduced

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	90	210	300	30.00 (<i>sensitivity</i>)
cancer = no	140	9560	9700	98.56 (<i>specificity</i>)
Total	230	9770	10000	96.50 (<i>accuracy</i>)

- Sensitivity = $TP/P = 90/300 = 30\%$
- Specificity = $TN/N = 9560/9700 = 98.56\%$
- Accuracy = $(TP + TN)/All = (90+9560)/10000 = 96.50\%$
- Error rate = $(FP + FN)/All = (140 + 210)/10000 = 3.50\%$
- Precision = $TP/(TP + FP) = 90/(90 + 140) = 90/230 = 39.13\%$
- Recall = $TP/ (TP + FN) = 90/(90 + 210) = 90/300 = 30.00\%$
- $F1 = 2 P \times R / (P + R) = 2 \times 39.13\% \times 30.00\% / (39.13\% + 30\%) = 33.96\%$

Classifier Evaluation: Holdout & Cross-Validation

- **Holdout method**
 - Given data is randomly partitioned into two independent sets
 - Training set (e.g., 2/3) for model construction
 - Test set (e.g., 1/3) for accuracy estimation
 - Repeated random sub-sampling validation: a variation of holdout
 - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- **Cross-validation (k -fold, where $k = 10$ is most popular)**
 - Randomly partition the data into k *mutually exclusive* subsets, each approximately equal size
 - At i -th iteration, use D_i as test set and others as training set
 - Leave-one-out: k folds where $k = \#$ of tuples, for small sized data
 - ***Stratified cross-validation***: folds are stratified so that class distribution, in each fold is approximately the same as that in the initial data

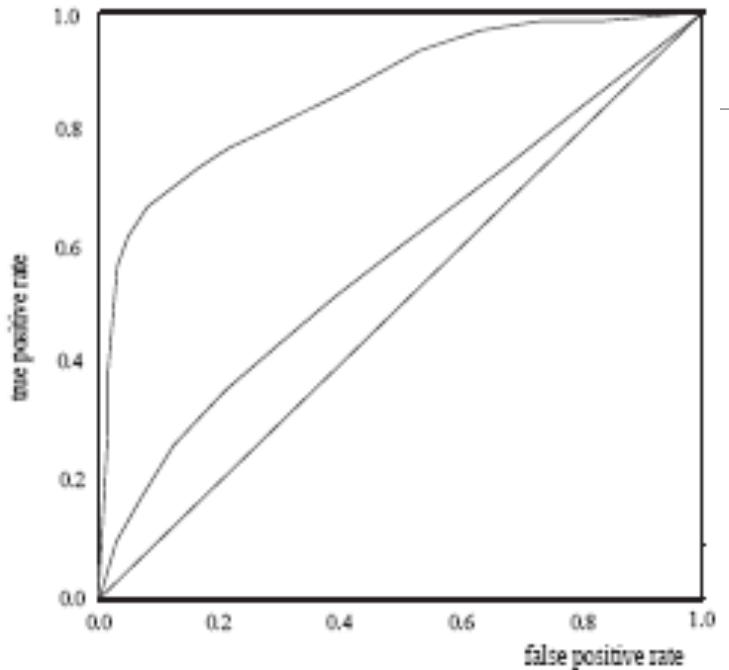
Classifier Evaluation: Bootstrap

- **Bootstrap**
 - Works well with small data sets
 - Samples the given training tuples uniformly *with replacement*
 - Each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- Several bootstrap methods, and a common one is **.632 bootstrap**
 - A data set with d tuples is sampled d times, with replacement, resulting in a training set of d samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2% of the original data end up in the bootstrap, and the remaining 36.8% form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
 - Repeat the sampling procedure k times, overall accuracy of the model:

$$Acc(M) = \frac{1}{k} \sum_{i=1}^k (0.632 \times Acc(M_i)_{test_set} + 0.368 \times Acc(M_i)_{train_set})$$

Model Selection: ROC Curves

- ❑ ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models
- ❑ Originated from signal detection theory
- ❑ Shows the trade-off between the true positive rate and the false positive rate
- ❑ The area under the ROC curve (**AUC**: Area Under Curve) is a measure of the accuracy of the model
- ❑ Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- ❑ The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



- ❑ Vertical axis represents the true positive rate
- ❑ Horizontal axis rep. the false positive rate
- ❑ The plot also shows a diagonal line
- ❑ A model with perfect accuracy will have an area of 1.0

Issues Affecting Model Selection

- **Accuracy**
 - classifier accuracy: predicting class label
- **Speed**
 - time to construct the model (training time)
 - time to use the model (classification/prediction time)
- **Robustness:** handling noise and missing values
- **Scalability:** efficiency in disk-resident databases
- **Interpretability**
 - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

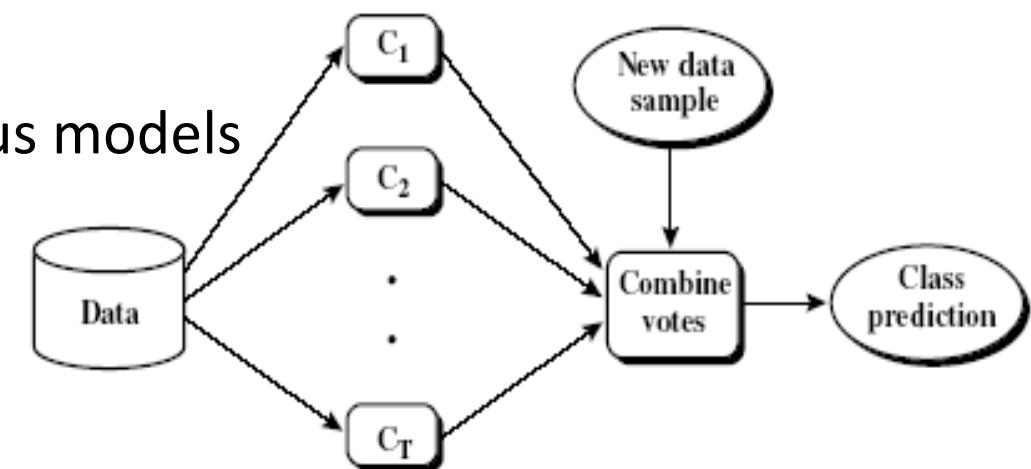
Chapter 8. Classification: Basic Concepts

- ❑ Classification: Basic Concepts
- ❑ Decision Tree Induction
- ❑ Bayes Classification Methods
- ❑ Linear Classifier
- ❑ Model Evaluation and Selection
- ❑ Techniques to Improve Classification Accuracy: Ensemble Methods
- ❑ Additional Concepts on Classification
- ❑ Summary



Ensemble Methods: Increasing the Accuracy

- ❑ Ensemble methods
 - ❑ Use a combination of models to increase accuracy
 - ❑ Combine a series of k learned models, M_1, M_2, \dots, M_k , with the aim of creating an improved model M^*
- ❑ Popular ensemble methods
 - ❑ Bagging: Trains each model using a subset of the training set, and models learned in parallel
 - ❑ Boosting: Trains each new model instance to emphasize the training instances that previous models mis-classified, and models learned in order



Bagging: Bootstrap Aggregation

- Analogy: Diagnosis based on multiple doctors' majority vote
- Training
 - Given a set D of d tuples, at each iteration i , a training set D_i of d tuples is sampled with replacement from D (i.e., bootstrap)
 - A classifier model M_i is learned for each training set D_i
- Classification: classify an unknown sample X
 - Each classifier M_i returns its class prediction
 - The bagged classifier M^* counts the votes and assigns the class with the most votes to X
- Prediction: It can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple
- Accuracy: Improved accuracy in prediction
 - Often significantly better than a single classifier derived from D
 - For noise data: Not considerably worse, more robust

Random Forest: Basic Concepts

- Random Forest (first proposed by L. Breiman in 2001)
 - A variation of bagging for *decision trees*
 - *Data bagging*
 - Use a subset of training data by sampling with replacement for each tree
 - *Feature bagging*
 - At each node use a random selection of attributes as candidates and split by the best attribute among them
 - Compared to original bagging, increases the diversity among generated trees
 - During classification, each tree votes and the most popular class is returned

Random Forest

- Two Methods to construct Random Forest:
 - Forest-RI (*random input selection*): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
 - Forest-RC (*random linear combinations*): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- Comparable in accuracy to Adaboost, but more robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split, and faster than typical bagging or boosting

Boosting

- ❑ Analogy: Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy
- ❑ How boosting works?
 - ❑ **Weights** are assigned to each training tuple
 - ❑ A series of k classifiers is iteratively learned
 - ❑ After a classifier M_i is learned, the weights are updated to allow the subsequent classifier, M_{i+1} , to **pay more attention to the training tuples that were misclassified** by M_i
 - ❑ The final **M^* combines the votes** of each individual classifier, where the weight of each classifier's vote is a function of its accuracy
- ❑ Boosting algorithm can be extended for numeric prediction
- ❑ Comparing with bagging: Boosting tends to have greater accuracy, but it also risks overfitting the model to misclassified data

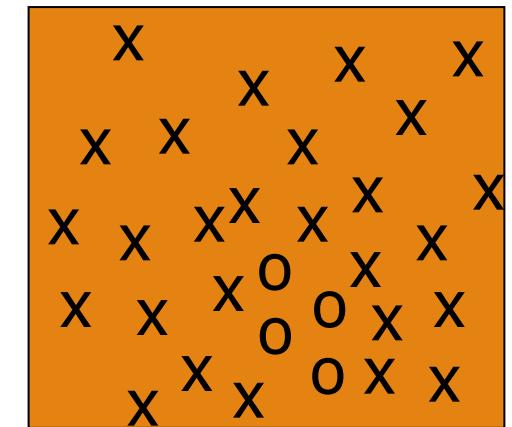
Adaboost (Freund and Schapire, 1997)

- Given a set of d class-labeled tuples, $(\mathbf{X}_1, y_1), \dots, (\mathbf{X}_d, y_d)$
- Initially, all the weights of tuples are set the same ($1/d$)
- Generate k classifiers in k rounds. At round i ,
 - Tuples from D are sampled (with replacement) to form a training set D_i of the same size
 - Each tuple's chance of being selected is based on its weight
 - A classification model M_i is derived from D_i
 - Its error rate is calculated using D_i as a test set
 - If a tuple is misclassified, its weight is increased; otherwise, it is decreased
- Error rate: $\text{err}(\mathbf{X}_j)$ is the misclassification error of tuple \mathbf{X}_j . Classifier M_i error rate is the sum of the weights of the misclassified tuples:
- The weight of classifier M_i 's vote is
$$\log \frac{1 - \text{error}(M_i)}{\text{error}(M_i)}$$

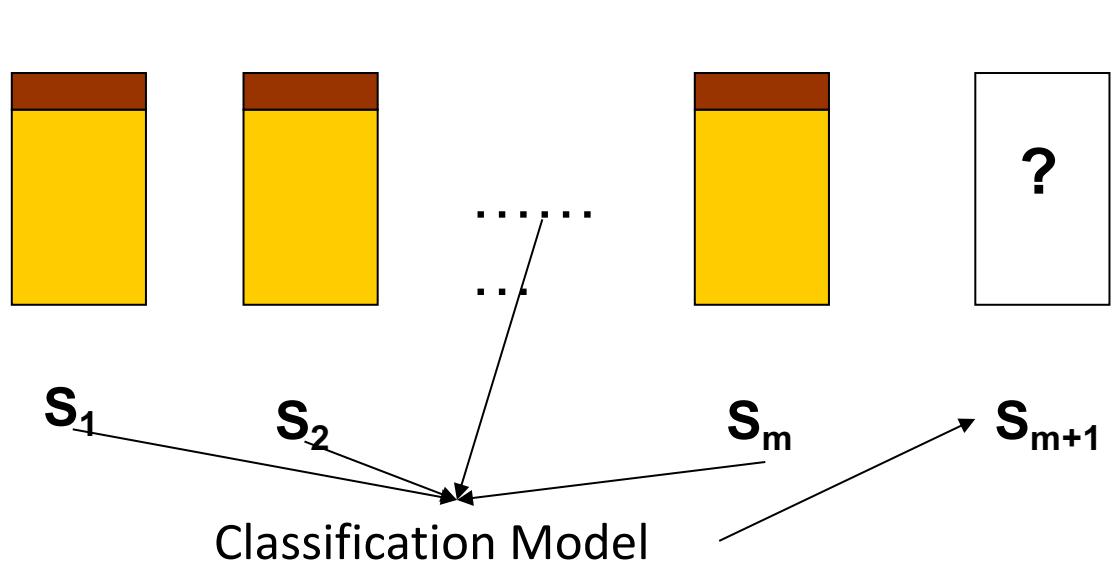
$$\text{error}(M_i) = \sum_j^d w_j \times \text{err}(\mathbf{X}_j)$$

Classification of Class-Imbalanced Data Sets

- Class-imbalance problem: Rare positive examples but numerous negative ones
 - E.g., medical diagnosis, fraud transaction, accident (oil-spill), and product fault
- Traditional methods assume a balanced distribution of classes and equal error costs: not suitable for class-imbalanced data
- Typical methods on imbalanced data in two-class classification
 - **Oversampling:** Re-sampling of data from positive class
 - **Under-sampling:** Randomly eliminate tuples from negative class
 - **Threshold-moving:** Move the decision threshold, t , so that the rare class tuples are easier to classify, and hence, less chance of costly false negative errors
 - **Ensemble techniques:** Ensemble multiple classifiers introduced above
- Still difficult for class imbalance problem on multiclass tasks



Classifying Data Streams with Skewed Distribution



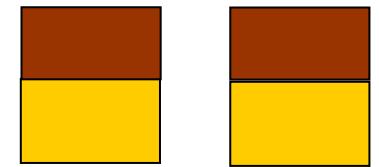
S_m as training data? Two few positive examples!

J. Gao, et al., "A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions", SDM'07

Biased Sampling



Ensemble



$$f^E(x) = \frac{1}{k} \sum_{i=1}^k f^i(x)$$

- Classify data stream with skewed distribution (i.e., rare events)
- **Biased sampling:** Save only the positive examples in the streams
- **Ensemble:** Partition negative examples of S_m into k portions to build k classifiers
- Effectively reduce classification errors on the minority class

Chapter 8. Classification: Basic Concepts

- ❑ Classification: Basic Concepts
- ❑ Decision Tree Induction
- ❑ Bayes Classification Methods
- ❑ Linear Classifier
- ❑ Model Evaluation and Selection
- ❑ Techniques to Improve Classification Accuracy: Ensemble Methods
- ❑ Additional Concepts on Classification
- ❑ Summary

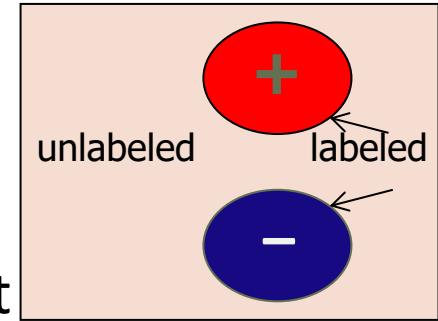


Multiclass Classification

- Classification involving more than two classes (i.e., > 2 Classes)
- Methodology: Reducing the multi-class problem into multiple binary problems
- Method 1. **One-vs.-rest** (or **one-vs.-all**)
 - Given m classes, train m classifiers: one for each class
 - Classifier j: treat tuples in class j as *positive* & **all the rest** as *negative*
 - To classify a tuple X , the set of classifiers vote as an ensemble
- Method 2. **one-vs.-one** (or **all-vs.-all**): Learn a classifier for each pair of classes
 - Given m classes, construct $m(m - 1)/2$ binary classifiers
 - A classifier is trained using tuples of the two classes
 - To classify a tuple X , each classifier votes
 - X is assigned to the class with maximal vote
- Comparison: One-vs.-one tends to perform better than one-vs.-rest
- Many new algorithms have been developed to go beyond binary classifier method

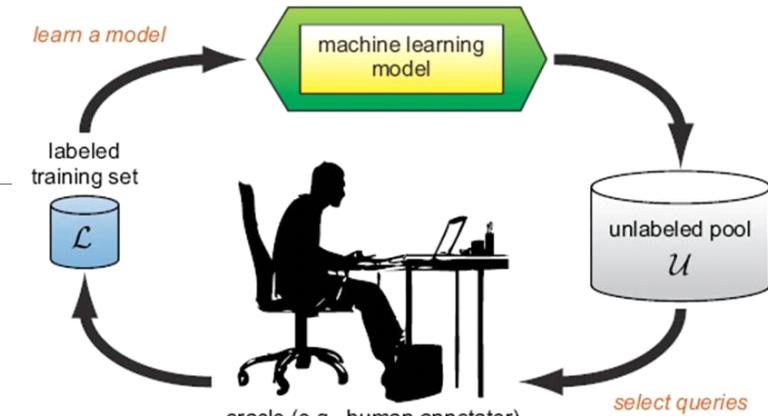
Semi-Supervised Classification

- ❑ Semi-supervised: Uses labeled and unlabeled data to build a classifier
- ❑ Self-training
 - ❑ Build a classifier using the labeled data
 - ❑ Use it to label the unlabeled data, and those with the most confident label prediction are added to the set of labeled data
 - ❑ Repeat the above process
 - ❑ Adv.: easy to understand; Disadv.: may reinforce errors
- ❑ Co-training: Use two or more classifiers to teach each other
 - ❑ Each learner uses a mutually independent set of features of each tuple to train a good classifier, say f_1 and f_2
 - ❑ Then f_1 and f_2 are used to predict the class label for unlabeled data X
 - ❑ Teach each other: The tuple having the most confident prediction from f_1 is added to the set of labeled data for f_2 & vice versa
- ❑ Other methods include joint probability distribution of features and labels



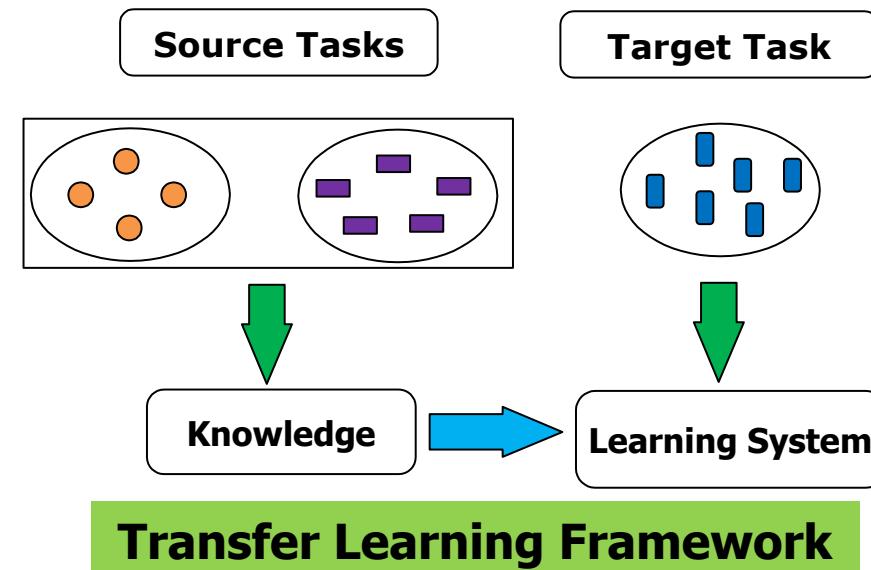
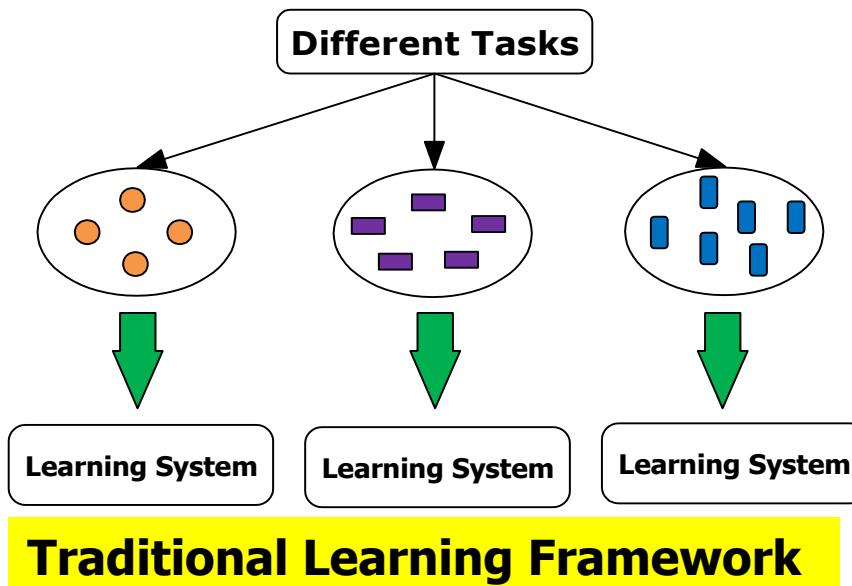
Active Learning

- A special case of semi-supervised learning
 - Unlabeled data: Abundant
 - Class labels are expensive to obtain
- Active learner: Interactively query teachers (oracle) for labels
- Pool-based approach: Uses a pool of unlabeled data
 - L: a small subset of D is labeled, U: a pool of unlabeled data in D
 - Use a query function to carefully select one or more tuples from U and request labels from an oracle (a human annotator)
 - The newly labeled samples are added to L, and learn a model
 - Goal: **Achieve high accuracy using as few labeled data as possible**
- Evaluated using *learning curves*: Accuracy as a function of the number of instances queried (# of tuples to be queried should be small)
- A lot of algorithms have been developed for active learning



Transfer Learning: Conceptual Framework

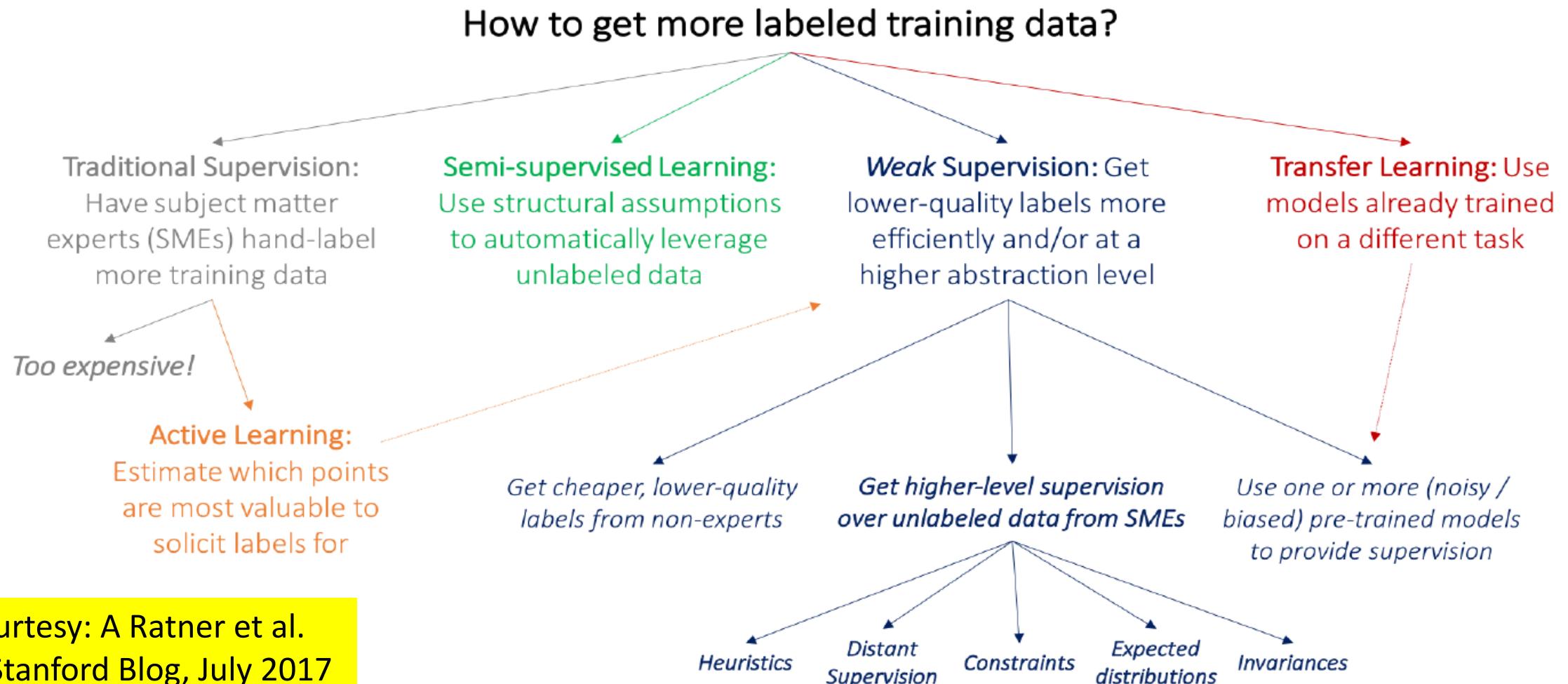
- Transfer learning: Extract knowledge from one or more source tasks (e.g., recognizing cars) and apply the knowledge to a target task (e.g., recognizing trucks)
- Traditional learning: Build a new classifier for each new task
- Transfer learning: Build new classifier by applying existing knowledge learned from source tasks
- Many algorithms are developed, applied to text classification, spam filtering, etc.



Weak Supervision: A New Programming Paradigm for Machine Learning

- ❑ Overcome the training data bottleneck
 - ❑ Leverage higher-level and/or noisier input from experts
- ❑ Exploring weak label distributions provided more cheaply and efficiently by
 - ❑ Higher-level, less precise supervision (e.g., heuristic rules, expected label distributions)
 - ❑ Cheaper, lower-quality supervision (e.g. crowdsourcing)
 - ❑ Existing resources (e.g. knowledge bases, pre-trained models)
- ❑ These weak label distributions could take many forms
 - ❑ Weak Labels from crowd workers, output of heuristic rules, or the result of distant supervision (from KBs), or the output of other classifiers, etc.
 - ❑ Constraints and invariances (e.g., from physics, logic, or other experts)
 - ❑ Probability distributions (e.g., from weak or biased classifiers or user-provided label or feature expectations or measurements)

Relationships Among Different Kinds of Supervisions



Courtesy: A Ratner et al.
@Stanford Blog, July 2017

Many areas of machine learning are motivated by the bottleneck of labeled training data, but are divided at a high-level by what information they leverage instead.

Chapter 8. Classification: Basic Concepts

- ❑ Classification: Basic Concepts
- ❑ Decision Tree Induction
- ❑ Bayes Classification Methods
- ❑ Linear Classifier
- ❑ Model Evaluation and Selection
- ❑ Techniques to Improve Classification Accuracy: Ensemble Methods
- ❑ Additional Concepts on Classification
- ❑ Summary



Summary

- Classification: Model construction from a set of training data
- Effective and scalable methods
 - Decision tree induction, Bayes classification methods, linear classifier, ...
 - No single method has been found to be superior over all others for all data sets
- Evaluation metrics: Accuracy, sensitivity, specificity, precision, recall, F measure
- Model evaluation: Holdout, cross-validation, bootstrapping, ROC curves (AUC)
- Improve Classification Accuracy: Bagging, boosting
- Additional concepts on classification: Multiclass classification, semi-supervised classification, active learning, transfer learning, weak supervision

References (1)

- C. Apte and S. Weiss. **Data mining with decision trees and decision rules**. Future Generation Computer Systems, 13, 1997
- P. K. Chan and S. J. Stolfo. **Learning arbiter and combiner trees from partitioned data for scaling machine learning**. KDD'95
- A. J. Dobson. **An Introduction to Generalized Linear Models**. Chapman & Hall, 1990.
- R. O. Duda, P. E. Hart, and D. G. Stork. **Pattern Classification**, 2ed. John Wiley, 2001
- U. M. Fayyad. **Branching on attribute values in decision tree generation**. AAAI'94.
- Y. Freund and R. E. Schapire. **A decision-theoretic generalization of on-line learning and an application to boosting**. J. Computer and System Sciences, 1997.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. **Rainforest: A framework for fast decision tree construction of large datasets**. VLDB'98.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, **BOAT -- Optimistic Decision Tree Construction**. SIGMOD'99.
- T. Hastie, R. Tibshirani, and J. Friedman. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer-Verlag, 2001.

References (2)

- ❑ T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. **A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms.** Machine Learning, 2000
- ❑ J. Magidson. **The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection.** In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, Blackwell Business, 1994
- ❑ M. Mehta, R. Agrawal, and J. Rissanen. **SLIQ : A fast scalable classifier for data mining.** EDBT'96
- ❑ T. M. Mitchell. **Machine Learning.** McGraw Hill, 1997
- ❑ S. K. Murthy, **Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey,** Data Mining and Knowledge Discovery 2(4): 345-389, 1998
- ❑ J. R. Quinlan. **Induction of decision trees.** *Machine Learning*, 1:81-106, 1986.
- ❑ J. R. Quinlan. **C4.5: Programs for Machine Learning.** Morgan Kaufmann, 1993.
- ❑ J. R. Quinlan. **Bagging, boosting, and c4.5.** AAAI'96.

References (3)

- R. Rastogi and K. Shim. **Public: A decision tree classifier that integrates building and pruning.** VLDB'98
- J. Shafer, R. Agrawal, and M. Mehta. **SPRINT : A scalable parallel classifier for data mining.** VLDB'96
- J. W. Shavlik and T. G. Dietterich. **Readings in Machine Learning.** Morgan Kaufmann, 1990
- P. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining.** Addison Wesley, 2005
- S. M. Weiss and C. A. Kulikowski. **Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.** Morgan Kaufman, 1991
- S. M. Weiss and N. Indurkhy. **Predictive Data Mining.** Morgan Kaufmann, 1997
- I. H. Witten and E. Frank. **Data Mining: Practical Machine Learning Tools and Techniques,** 2ed. Morgan Kaufmann, 2005



Bayes' Theorem: Basics

- Total probability Theorem:

$$P(B) = \sum_{i=1}^M P(B|A_i)P(A_i)$$

- Bayes' Theorem:

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Let \mathbf{X} be a data sample (“evidence”): class label is unknown
- Let H be a *hypothesis* that \mathbf{X} belongs to class C
- Classification is to determine $P(H|\mathbf{X})$, (i.e., *posteriori probability*): the probability that the hypothesis holds given the observed data sample \mathbf{X}
- $P(H)$ (*prior probability*): the initial probability
 - E.g., \mathbf{X} will buy computer, regardless of age, income, ...
- $P(\mathbf{X})$: probability that sample data is observed
- $P(\mathbf{X}|H)$ (likelihood): the probability of observing the sample \mathbf{X} , given that the hypothesis holds
 - E.g., Given that \mathbf{X} will buy computer, the prob. that \mathbf{X} is 31..40, medium income

Classification Is to Derive the Maximum Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n -D attribute vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are m classes C_1, C_2, \dots, C_m .
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i | \mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i | \mathbf{X}) = \frac{P(\mathbf{X} | C_i)P(C_i)}{P(\mathbf{X})}$$

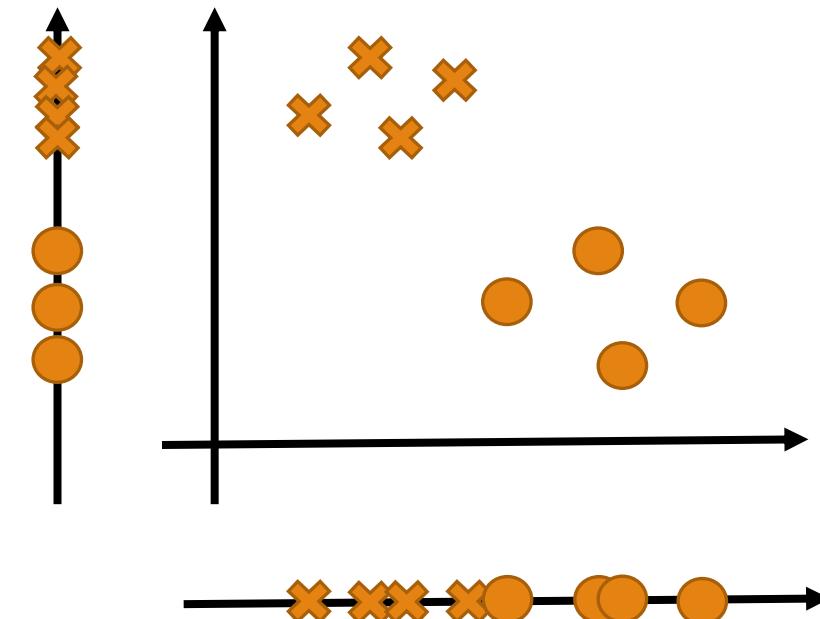
- Since $P(X)$ is constant for all classes, only

$$P(C_i | \mathbf{X}) \propto P(\mathbf{X} | C_i)P(C_i)$$

needs to be maximized

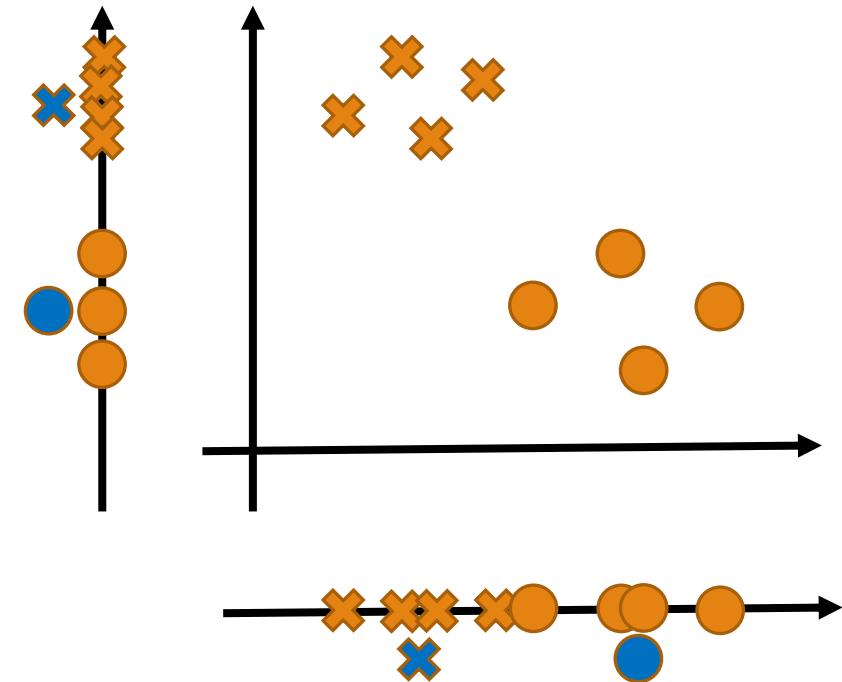
Linear Discriminant Analysis (LDA)

- ❑ Linear Discriminant Analysis (LDA) works when the attributes are all continuous
 - ❑ For the categorical attributes, discriminant correspondence analysis is the equivalent technique
- ❑ Basic Ideas: Project all samples on a line such that different classes are well separated
- ❑ Example: Suppose we have 2 classes and 2-dimensional samples x_1, \dots, x_n
 - ❑ n_1 samples come from class 1
 - ❑ n_2 samples come from class 2
- ❑ Let the line direction be given by unit vector \boldsymbol{v}
- ❑ There are two candidates of projections
 - ❑ Vertical: $\boldsymbol{v} = (0,1)$
 - ❑ Horizontal: $\boldsymbol{v} = (1,0)$
- ❑ Which one looks better?
- ❑ How to mathematically measure it?



Fisher's LDA (Linear Discriminant Analysis)

- ❑ $v^T x_i$ is the distance of projection of x_i from the origin
- ❑ Let μ_1 and μ_2 be the means of class 1 and class 2 in the original space
 - ❑ $\mu_1 = \frac{1}{n_1} \sum_{i \in \text{class 1}} x_i$
 - ❑ $\mu_2 = \frac{1}{n_2} \sum_{i \in \text{class 2}} x_i$
- ❑ The distance between the means of the projected points
 - ❑ $|v^T \mu_1 - v^T \mu_2|$
 - ❑ Good? No. Horizontal one may have larger distance



Fisher's LDA (con't)

- ❑ Normalization needed
- ❑ Scatter: Sample variance multiplied by n

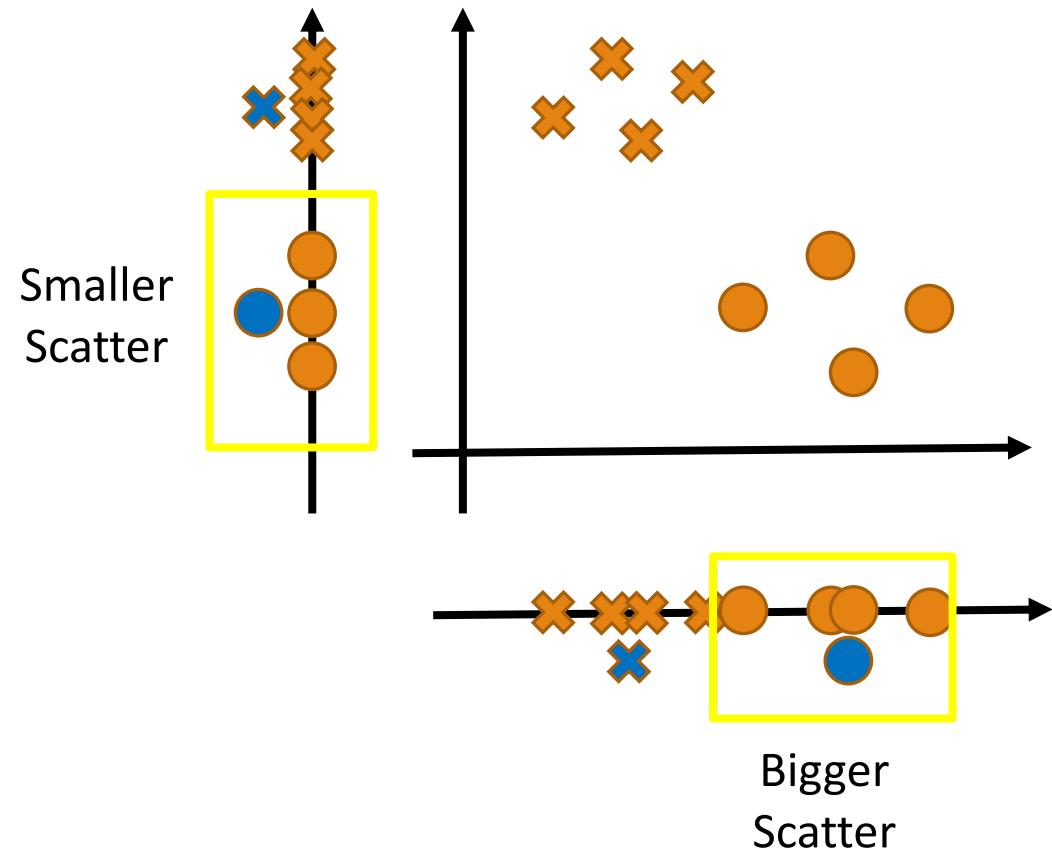
- ❑ $s_1 = \sum_{i \in \text{class 1}} (\mathbf{v}^T \mathbf{x}_i - \mathbf{v}^T \boldsymbol{\mu}_1)^2$

- ❑ $s_2 = \sum_{i \in \text{class 2}} (\mathbf{v}^T \mathbf{x}_i - \mathbf{v}^T \boldsymbol{\mu}_2)^2$

- ❑ Fisher's LDA

- ❑ Maximize $J(\mathbf{v}) = \frac{(\mathbf{v}^T \boldsymbol{\mu}_1 - \mathbf{v}^T \boldsymbol{\mu}_2)^2}{s_1 + s_2}$

- ❑ Closed-form optimal solution



Fisher's LDA: Summary

- Advantages
 - Useful for dimension reduction
 - Easy to extend to multi-classes

- Fisher's LDA will fail
 - When $\mu_1 = \mu_2, J(\nu)$ is always 0.
 - When classes have large overlap when projected to any line