# UVA CS 6316: Machine Learning

# Lecture 8: Supervised Classification

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

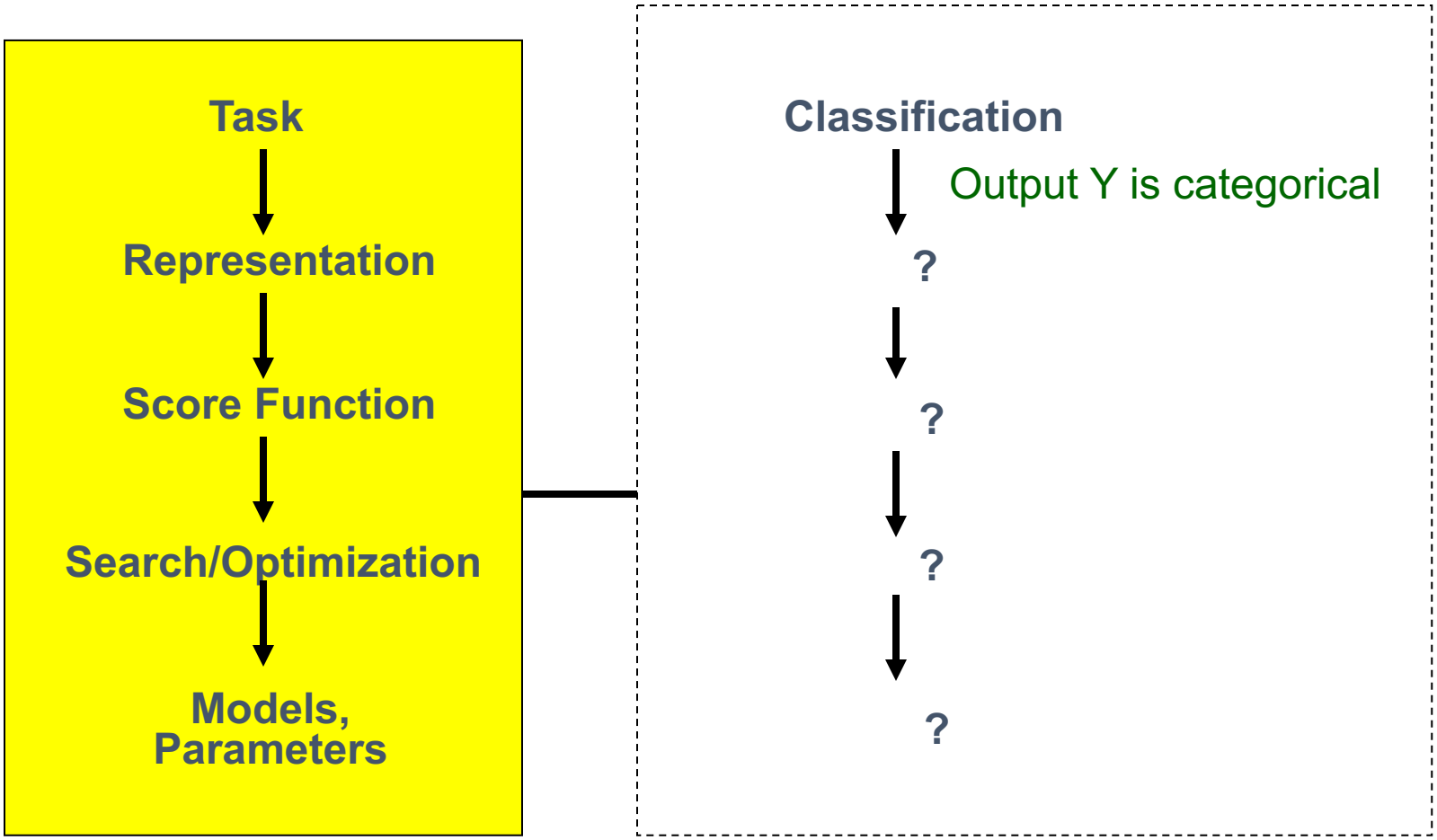
# Course Content Plan ➔
# Six major sections of this course

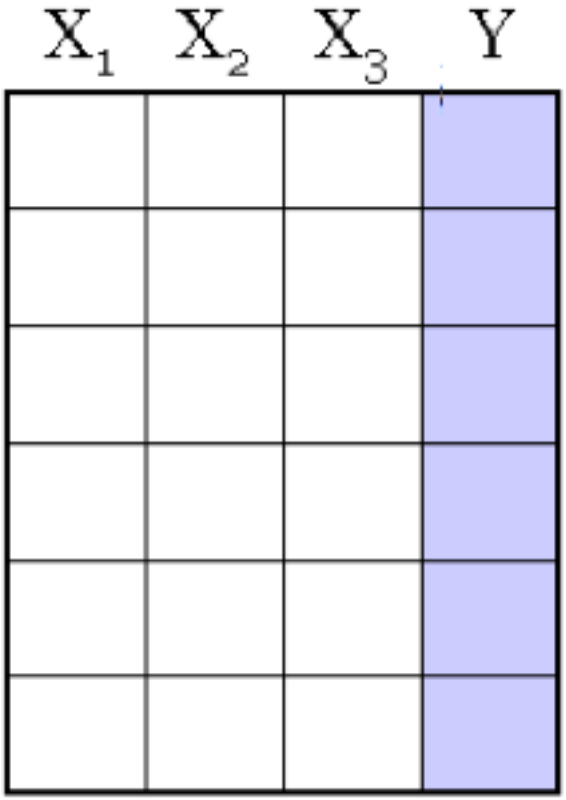
- ~~Regression (supervised)~~ — Y is a continuous
- Classification (supervised) — Y is a discrete
- Unsupervised models — NO Y
- Learning theory — About f()

- Graphical models — About interactions among X1,… Xp

- Reinforcement Learning — Learn program to Interact with its environment

# Supervised Classifiers





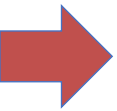

# A Dataset for classification

$$f : X \longrightarrow Y$$

Output Class: categorical variable

- **Data***/points/instances/examples/samples/records*: [ rows ]
- **Features***/attributes/dimensions/independent variables/covariates/predictors/regressors*: [ columns, except the last]
- **Target***/outcome/response/label/dependent variable*: special column to be predicted [ last column ]

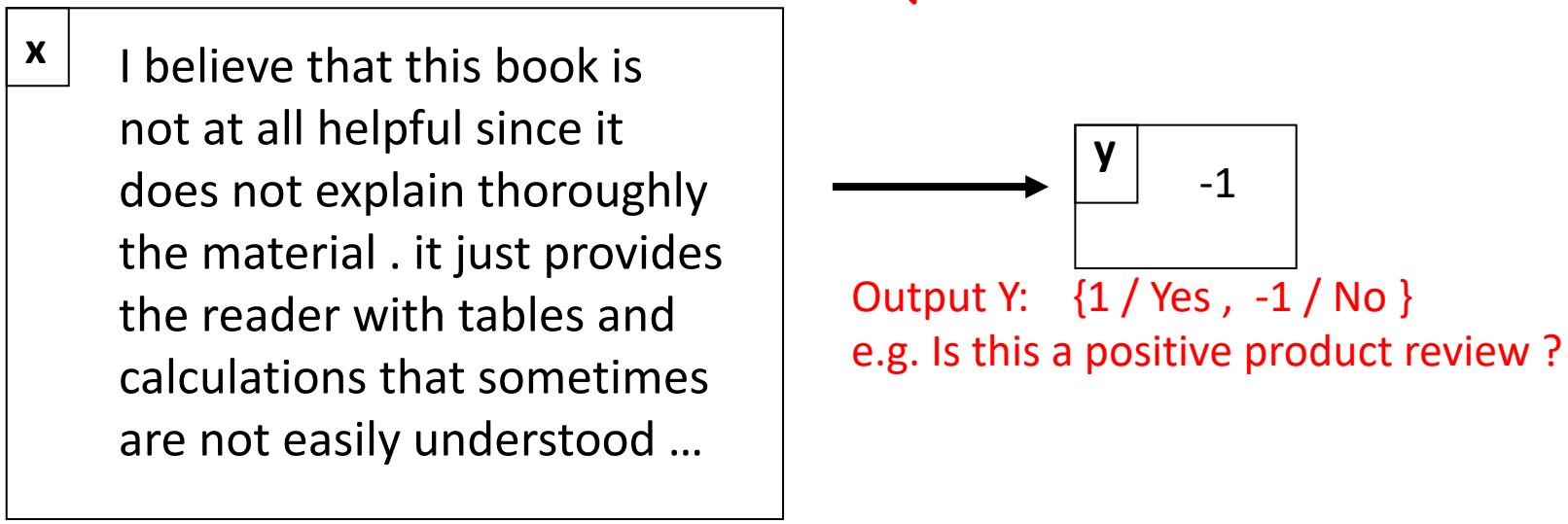
# Many Variants w.r.t. Y



- Binary Classification
- Multi-class Classification
- Hierarchical Classification
- Multi-label Classification
- Structured Predictions
- ……

# Binary: Text Review-based Sentiment Classification

sentiment / classification

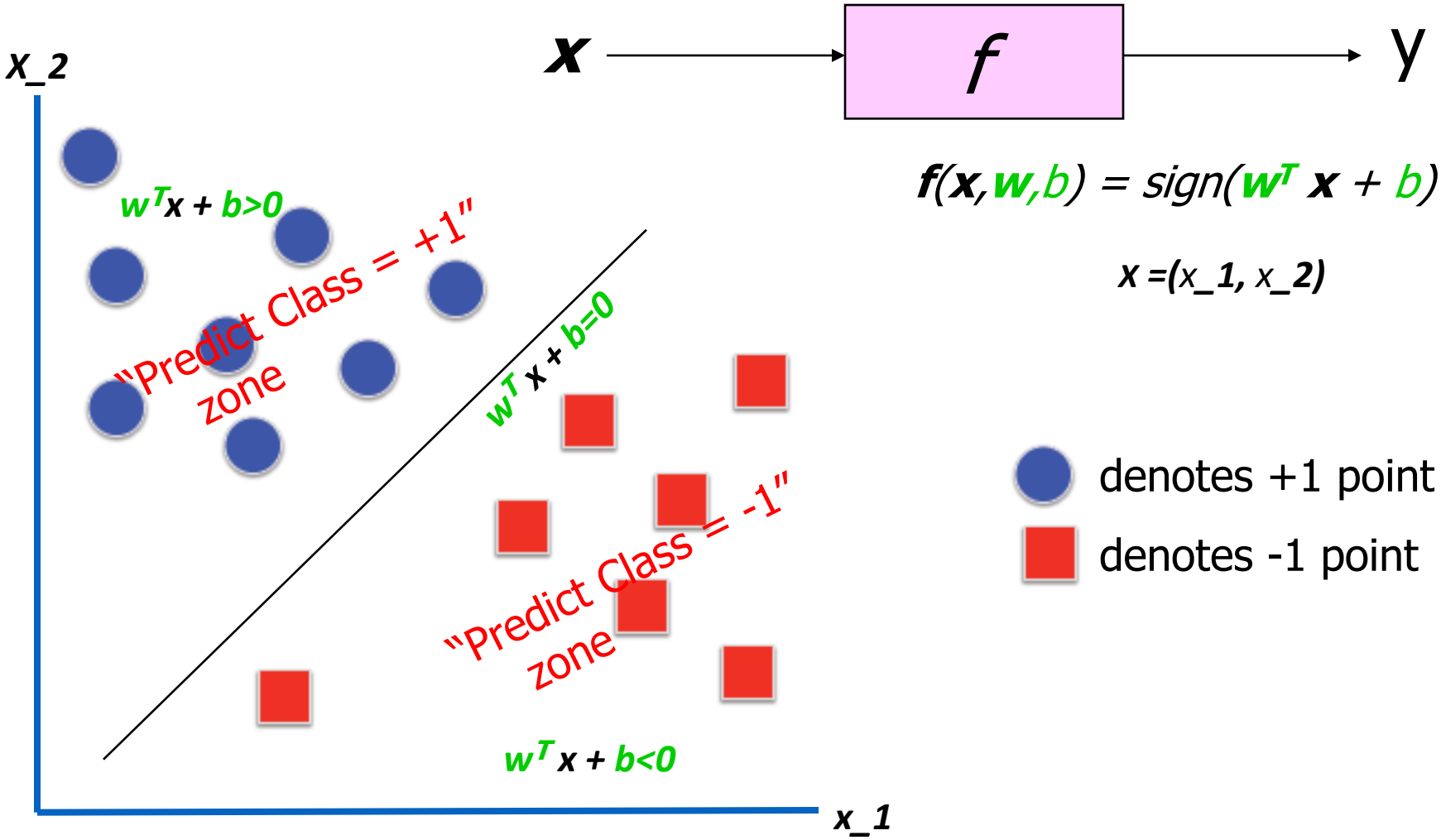



Input X : e.g. a piece of English text

Output Y: {1 / Yes , -1 / No }

e.g. Is this a positive product review ?

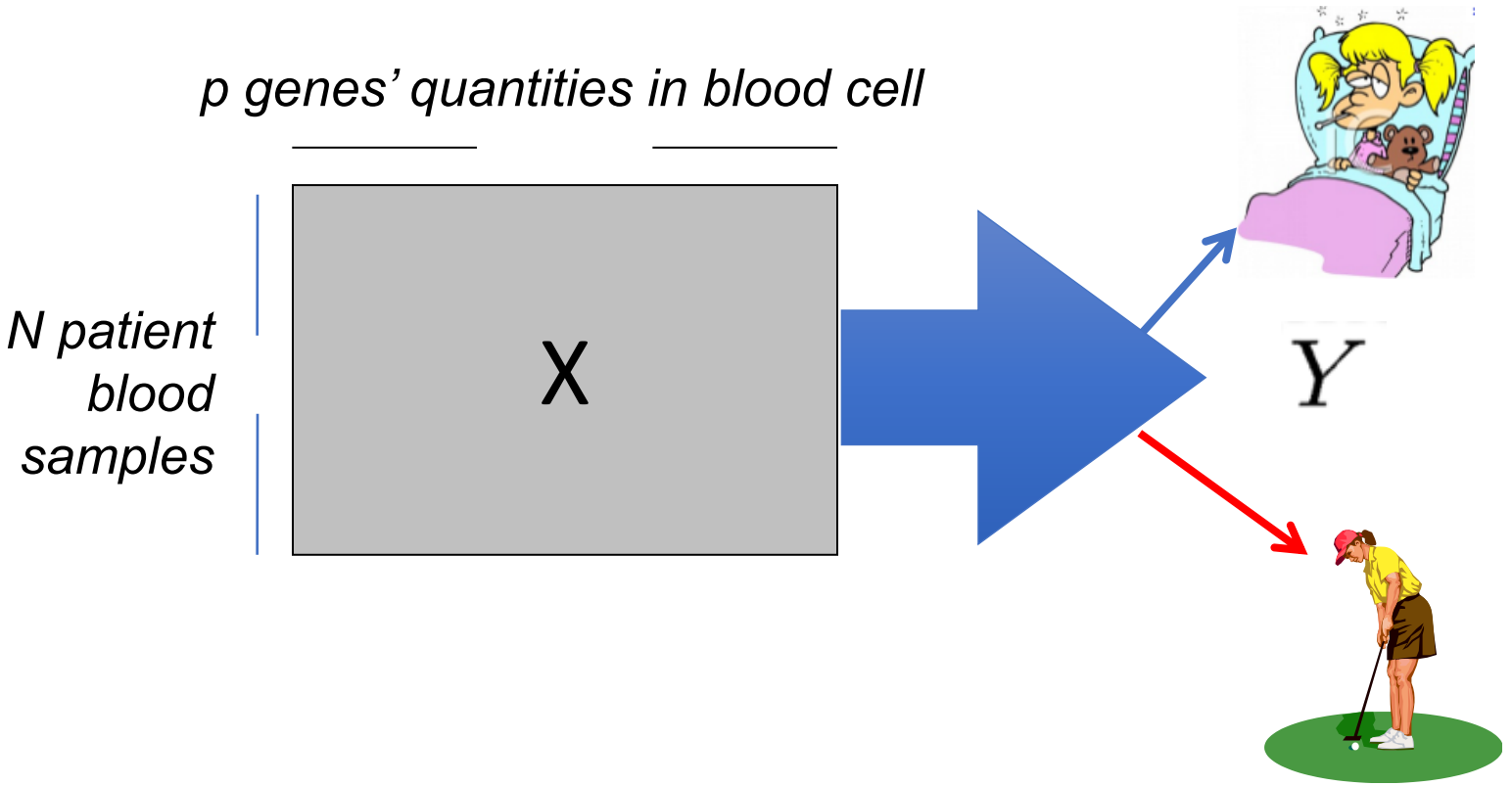
# Review: Linear Binary Classifier (2D)





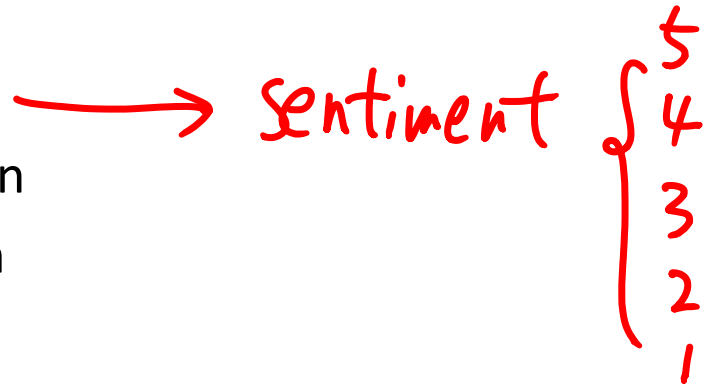

Courtesy slide from Prof. Andrew Moore's tutorial

# Binary: : Cancer Classification using gene expression



# Many Variants w.r.t. Y

- Binary Classification
- Multi-class Classification
- Hierarchical Classification
- Multi-label Classification
- Structured Predictions
- ……

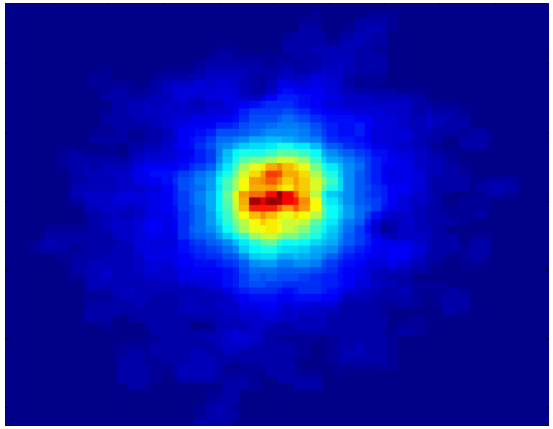


$x_i \rightarrow \{c_j\}$ $c_1, c_2, \ldots, c_m$

# Multi-Class: Classifying Galaxies

Courtesy: http://aps.umn.edu

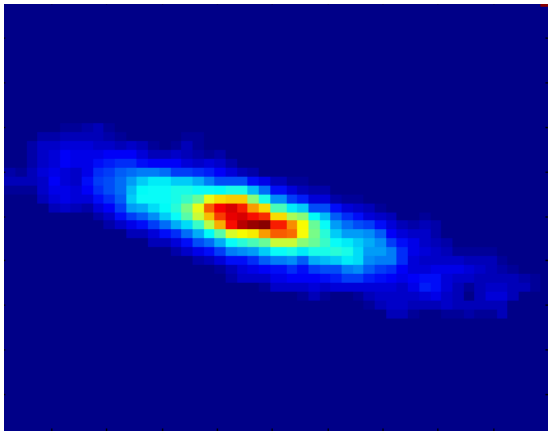
*Early*



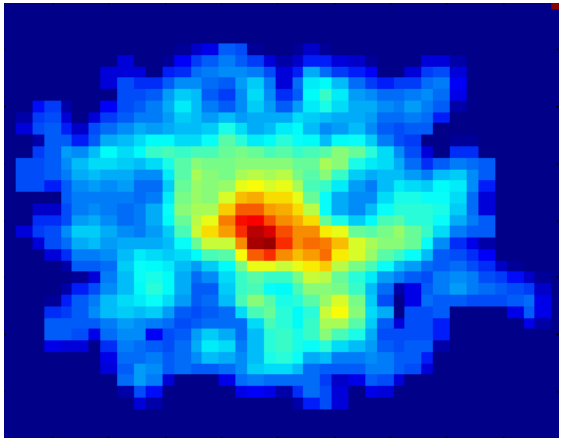
Class:
- Stages of Formation

Attributes:
- Image features,
- Characteristics of light waves received, etc.

*Intermediate*



*Late*

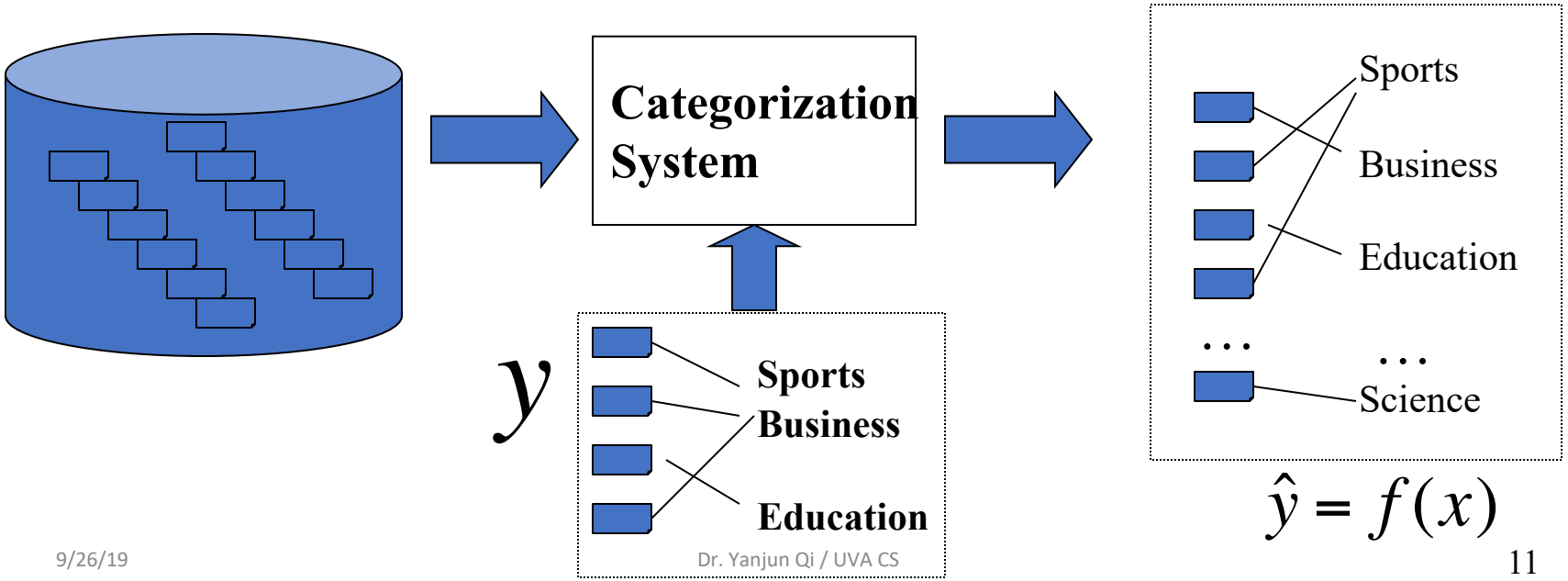


Data Size:
- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

From [Berry & Linoff] Data Mining Techniques, 1997

# Multi-Class: Text Categorization

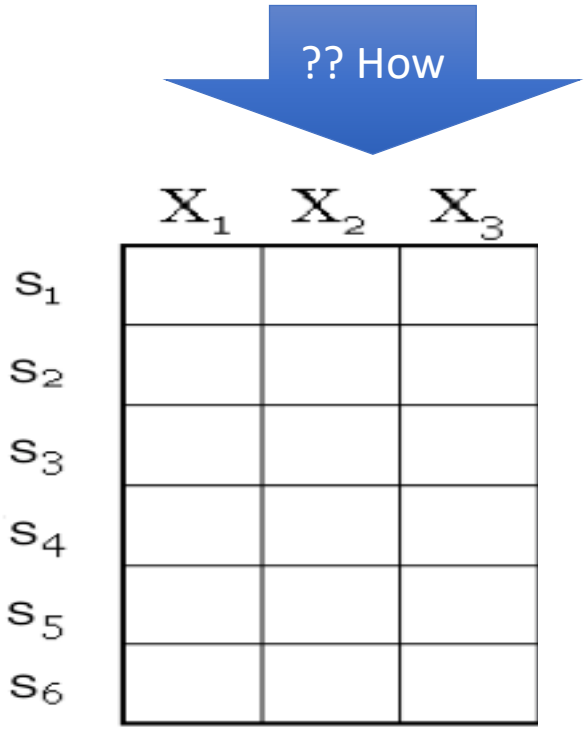
- Pre-given categories and labeled document examples (Categories may form hierarchy)
- Classify new documents
- A standard supervised learning problem



Text / Image / Audio

# Bag of Words Representation

- Text / String / Symbolic sequences
  - Variable length
  - Discrete
  - Combinatorial
  - Spatial ordering among units



# Text Document Representation (LATER)

- Each document becomes a `term' vector, [word]
  - each term is an (attribute) of the vector,
  - the value of each describes the number of times the corresponding term occurs in the document.

Fixed length

**Bag of 'words'**

| | team | coach | play | ball | score | game | win | lost | timeout | season |
|---|---|---|---|---|---|---|---|---|---|---|
| Document 1 | 3 | 0 | 5 | 0 | 2 | 6 | 0 | 2 | 0 | 2 |
| Document 2 | 0 | 7 | 0 | 2 | 1 | 0 | 0 | 3 | 0 | 0 |
| Document 3 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 | 3 | 0 |

histogram of occurrence

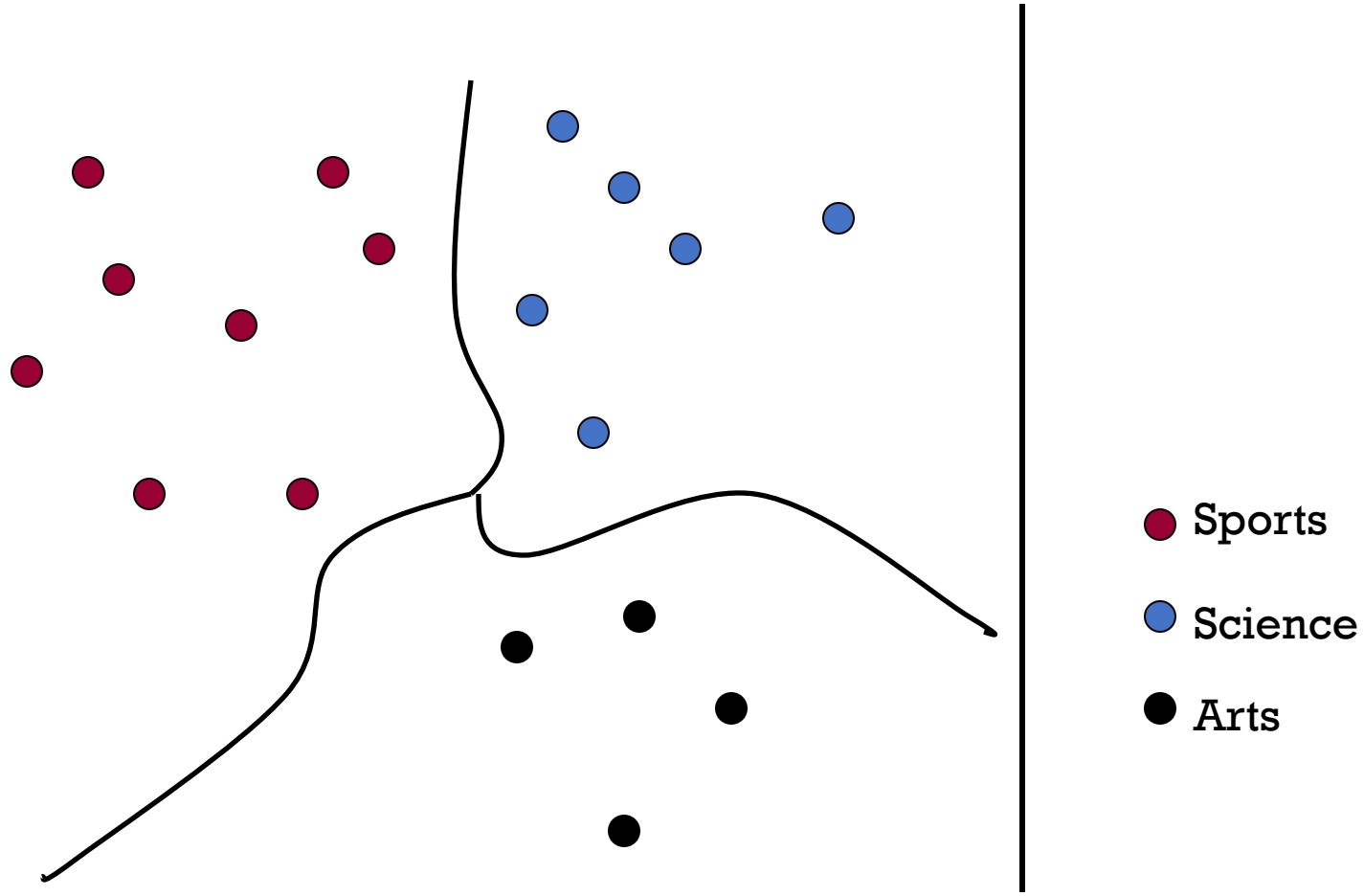
# Recall: Vector Space Representation

- Each document is a vector, one component for each term (= word).

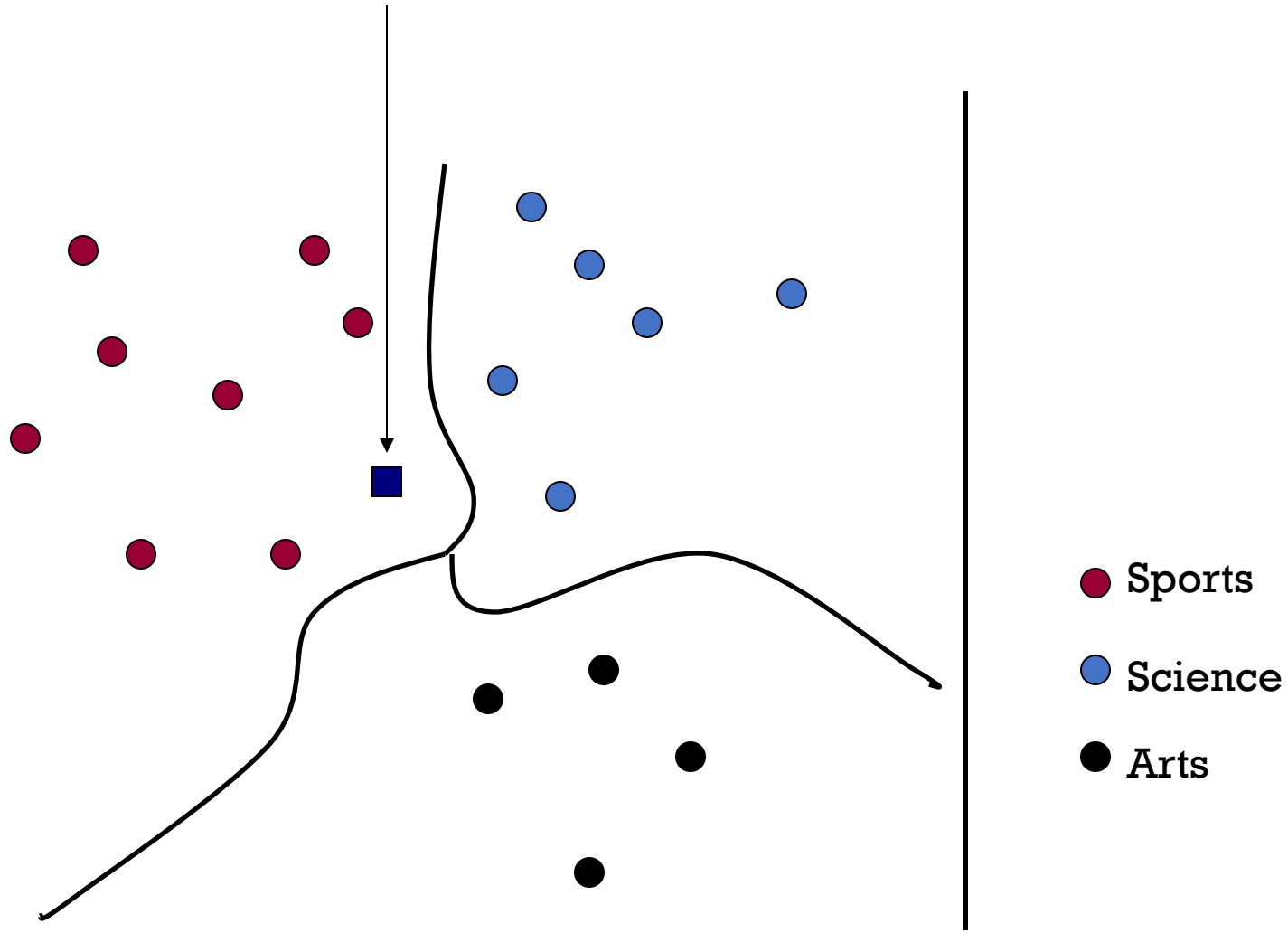
| | Doc 1 | Doc 2 | Doc 3 | ... |
|---|---|---|---|---|
| **Word 1** | 3 | 0 | 0 | ... |
| **Word 2** | 0 | 8 | 1 | ... |
| **Word 3** | 12 | 1 | 10 | ... |
| ... | 0 | 1 | 3 | ... |
| ... | 0 | 0 | 0 | ... |

- Normalize to unit length.
- High-dimensional vector space:
  - Terms are axes, 10,000+ dimensions, or even 100,000+
  - Docs are vectors in this space

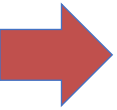
# Multiple Classes in a Vector Space



# Test Document = ?

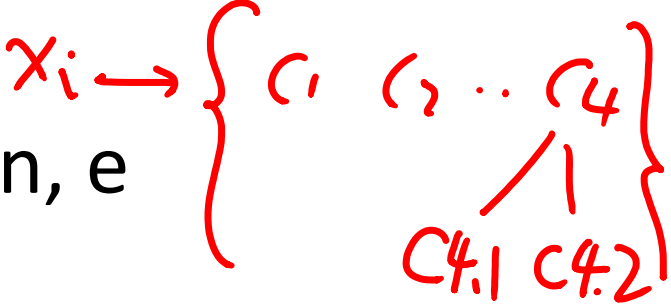


# Many Variants w.r.t. Y



- Binary Classification
- Multi-class Classification
- Hierarchical Classification
- Multi-label Classification
- Structured Predictions
- ……

# Hierarchical: Text Categorization, e.g. Google News




Google

Search News | Search the Web

Search and browse 4,500 news sources updated continuously.

7

- News
  - Top Stories
  - News near you
  - World
  - U.S.
  - Business
  - Technology (C4)
    - iPhone (C4.1)
    - Microsoft Windows (C4.2)
    - Minecraft
    - Safety
    - IBM
    - General Motors
    - Facebook
    - Microsoft Corporation
    - Tablet computers
    - Tor
  - Entertainment
  - Sports
  - Science
  - Health
  - Spotlight

Technology


PhoneDog

**Microsoft Keyboard Works With Windows, iOS, and Android**

PC Magazine - 53 minutes ago

With a handful of new peripherals, Microsoft is revamping older products and embracing the new mobile reality. 0shares. Microsoft Universal Mobile Keyboard.

Microsoft announces new line of accessories for Windows, Android, iOS, and ... BetaNews

Microsoft's new Universal Mobile Keyboard works with iOS, Android and ... ZDNet

Related
Microsoft Corporation »
Computer keyboards »
Microsoft Windows »

See realtime coverage

Trending on Google+: Microsoft's Universal Bluetooth Keyboard Will Work With Windows, Android, And ... Android Police

Opinion: Microsoft's New Universal Mobile Keyboard Has Android and iOS in Mind Gizmodo


BetaNews | PhoneDog | SlashGear | WinBeta | Hot Hardware


USA TODAY

**Microsoft/Minecraft Deal Gets a Skit On Conan O'Brien's Show**

GameSpot - 1 hour ago

During Monday's episode of Conan, the comedian aired a segment about how the inventor of Minecraft would be celebrating the massive pay day.


Moneycontr...

**Apple's iOS 8 available Wednesday**

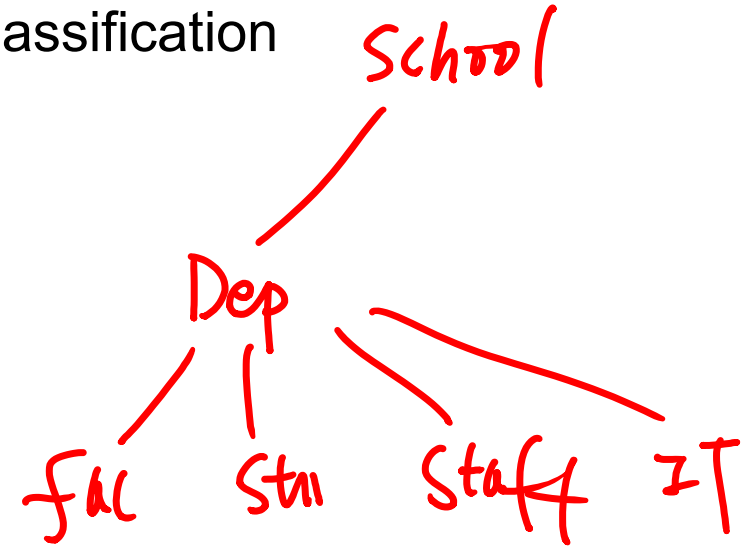
New York Daily News - 15 minutes ago

You don't need to order an iPhone 6 to feel like you've gotten a brand new phone. Apple's much-anticipated operating system update, iOS 8, will be available for download Wednesday.

**IBM Watson Data Analysis Service Revealed**

# Applications of Text Categorization

- News article classification
- Meta-data annotation
- Automatic Email sorting
- Web page classification
- ……




$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Hinge loss - Binary
Hie Class -
Cross-entropy - multi
Class

# Many Variants w.r.t. Y

- Binary Classification
- Multi-class Classification
- Hierarchical Classification
- Multi-label Classification
- Structured Predictions
- ……


Mountains
Castle
Road

# Multi Label Classification (MLC)

- MLC is the task of assigning a set of target labels for a given sample
- Given input ***x***, predict the set of labels $\{ y_1, y_2, \ldots, y_L \}$, $y_i \in \{0,1\}$

***x***

$y_1$ **Castle** ✓

$y_2$ **City** ✗

$y_3$ **Mountains** ✓

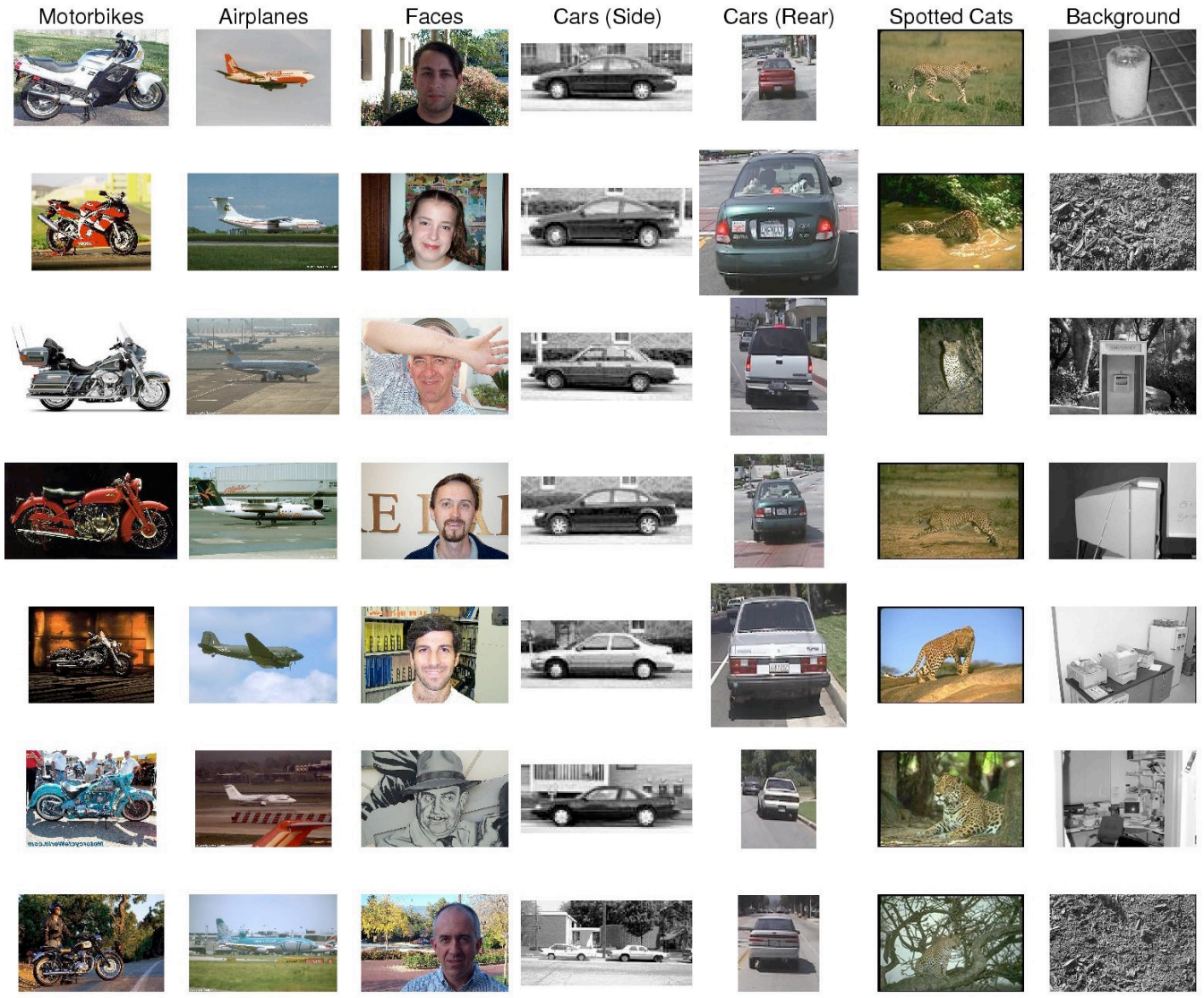
$y_4$ **Car** ✗

$y_5$ **Road** ✓



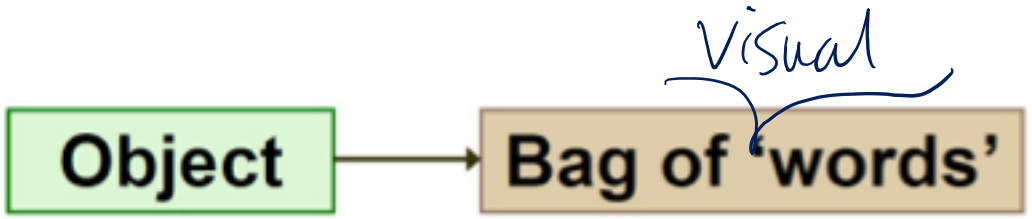
vs. multi class only C?

# Application: e.g., (Label Images into predefined word labels)

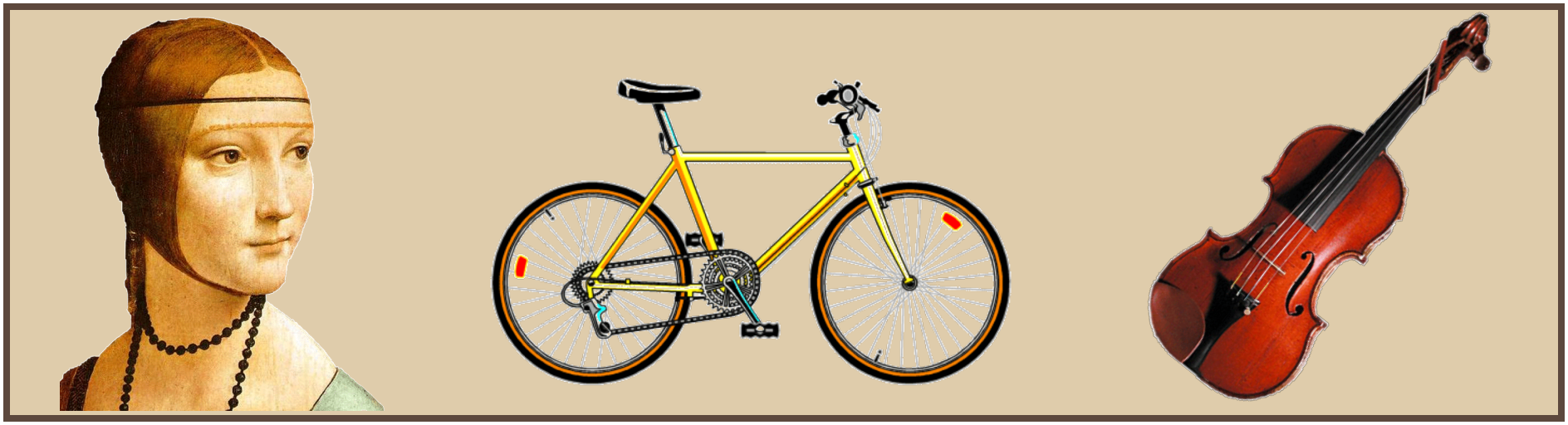


# When not using Deep Learning: Image Representation for – Objective recognition

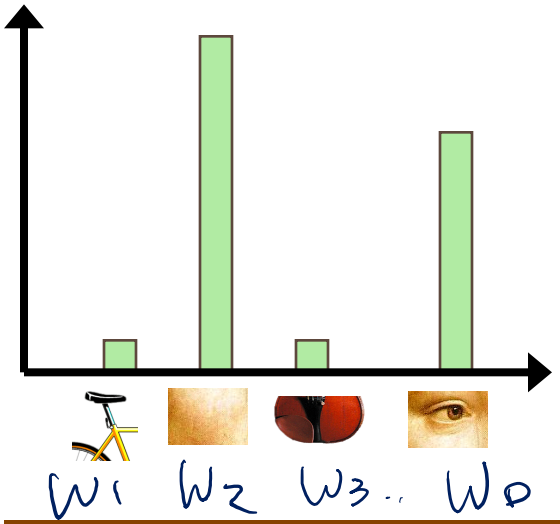
- Image representation ➔ bag of “visual words”

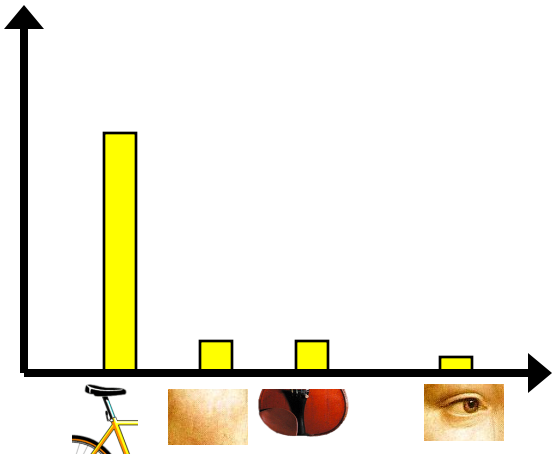


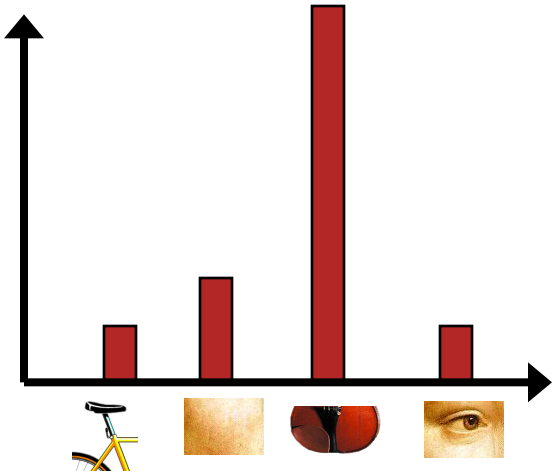

- An object image: histogram of visual vocabulary – a numerical vector of D dimensions.



Occlusion







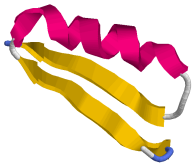
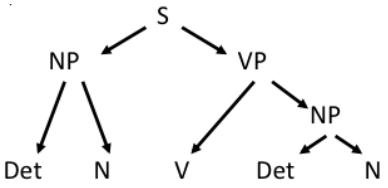
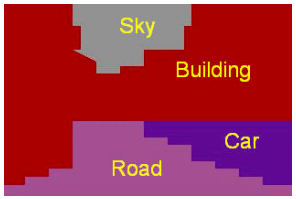
$W_1$ $W_2$ $W_3$ .. $W_D$

# Many Variants w.r.t. Y

- Binary Classification
- Multi-class Classification
- Hierarchical Classification
- Multi-label Classification
- Structured Predictions
- ……

# STRUCTURAL OUTPUT LEARNING : [ COMPLEXITY OF Y ]

- Many prediction tasks involve output labels having structured correlations or constraints among instances

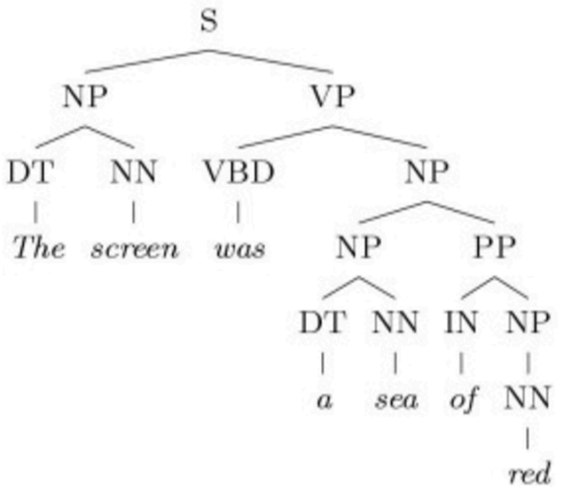
| Structured Dependency between Examples' Y | Sequence | Tree | Grid |
|---|---|---|---|
| Input $X$ | APAFSVSPASGACGPECA… | The dog chased the cat |  |
| Output $Y$ |  CCEEEEECCCCCHHHCCC… |   |   |

Many more possible structures between y_i , e.g. spatial , temporal, relational …

# Structured Output: -
# Natural Language Parsing

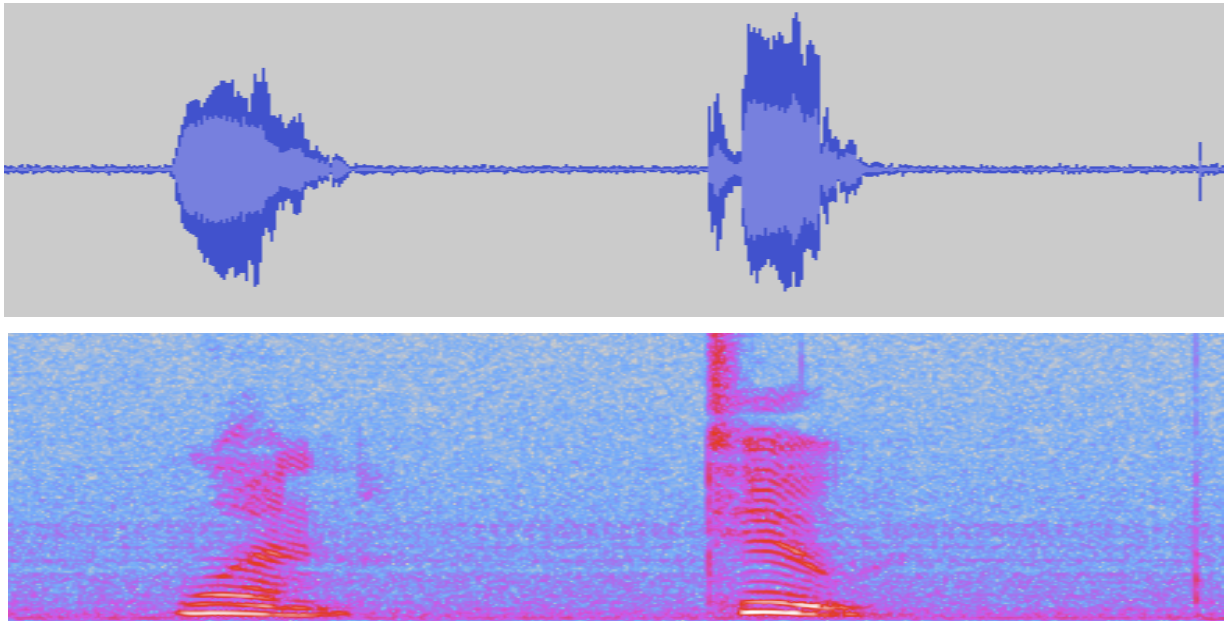
The screen was a sea of red

Language Parsing




From Vicente UVA

# Structured Output:
# – Audio Segmentation / Tagging



- Real-life applications:
  - Customer service phone routing
  - Voice recognition software

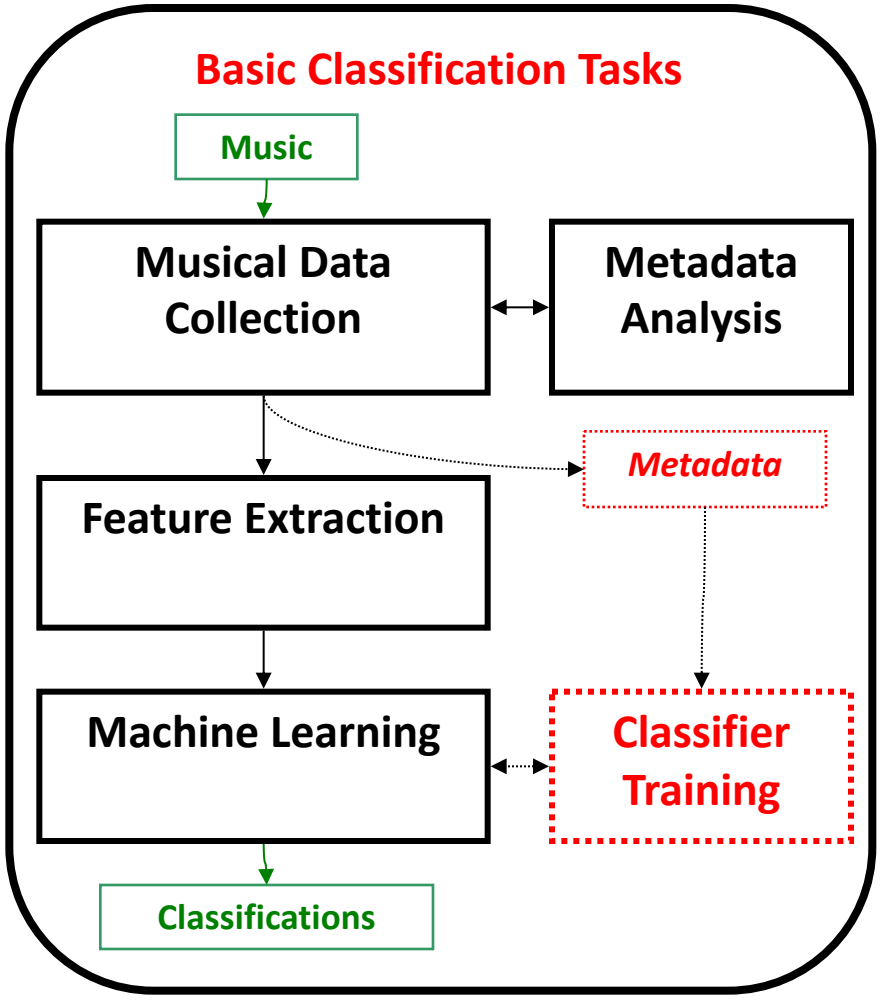
# Music Information Retrieval Systems
# e.g., Automatic Music Recommendation

- To classify music pieces in various ways
  - Genre or style classification
  - Mood classification
  - Performer or composer identification
  - Music recommendation
  - Playlist generation
  - Hit prediction
  - Audio to symbolic transcription
  - etc.
- Such areas often share similar central procedures

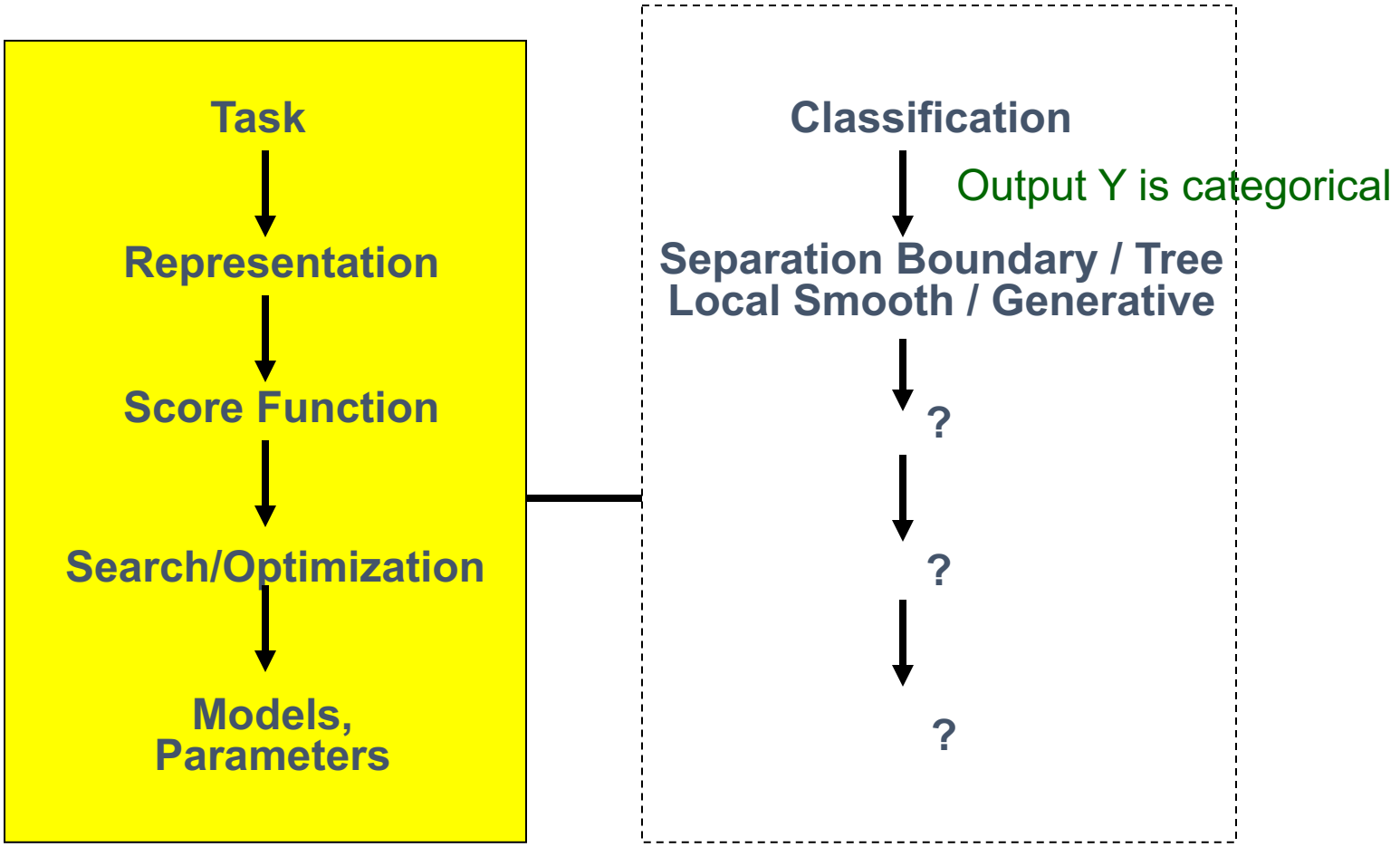
codebook/Audio word

# Music Information Retrieval Systems
## e.g., Automatic Music Recommendation (Extra)

- Musical data collection
  - The instances (basic entities) to classify
  - Audio recordings, scores, cultural data, etc.
- Feature extraction
  - Features represent characteristic information about instances
  - Must provide sufficient information to segment instances among classes (categories)
- Machine learning
  - Algorithms ("classifiers" or "learners") learn to associate feature patterns of instances with their classes



# Supervised Classifiers



# Three major sections for classification

- We can divide the large variety of classification approaches into roughly three major types

1. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., support vector machine, decision tree, logistic regression,
   - e.g. neural networks (NN), deep NN

HW3, preprocessing

HW4 Image

2. Generative:
   - build a generative statistical model
   - e.g., Bayesian networks, Naïve Bayes classifier

HW5 Text

3. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

HW3 scalability,

# Review: How can we build more intelligent computer / machine ?

- Able to
  - **perceive the world**
  - **understand the world**
  - **react to the world**

- This needs
  - Basic speech capabilities
  - Basic vision capabilities
  - Language/semantic understanding
  - User behavior / emotion understanding
  - Able to act
  - Able to think ?

# Detour: three planned programming assignments about AI tasks

- HW:  Semantic language understanding (sentiment classification on movie review text)
- HW: Visual object recognition (labeling images about handwritten digits)
- HW: Audio speech recognition (unsupervised learning based speech recognition task )

ICML '06 Proceedings of the 23rd international conference on Machine learning

# A study comparing Classifiers

## An Empirical Comparison of Supervised Learning Algorithms

**Rich Caruana** CARUANA@CS.CORNELL.EDU
**Alexandru Niculescu-Mizil** ALEXN@CS.CORNELL.EDU
Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

### Abstract

A number of supervised learning methods have been introduced in the last decade. Unfortunately, the last comprehensive empirical evaluation of supervised learning was the Statlog Project in the early 90's. We present a large-scale empirical comparison between ten supervised learning methods: SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps. We also examine the effect that calibrating the models via Platt Scaling and Isotonic Regression has on their performance. An important aspect of our study is the use of a variety of performance criteria to evaluate the learning methods.

This paper presents results of a large-scale empirical comparison of ten supervised learning algorithms using eight performance criteria. We evaluate the performance of SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps on eleven binary classification problems using a variety of performance metrics: accuracy, F-score, Lift, ROC Area, average precision, precision/recall break-even point, squared error, and cross-entropy. For each algorithm we examine common variations, and thoroughly explore the space of parameters. For example, we compare ten decision tree styles, neural nets of many sizes, SVMs with many kernels, etc.

Because some of the performance metrics we examine interpret model predictions as probabilities and models such as SVMs are not designed to predict probabil-

# A study comparing Classifiers

➔ 11 binary classification datasets

Small data

*Table 1.* Description of problems

| PROBLEM | #ATTR | TRAIN SIZE | TEST SIZE | %POZ |
|---|---|---|---|---|
| ADULT | 14/104 | 5000 | 35222 | 25% |
| BACT | 11/170 | 5000 | 34262 | 69% |
| COD | 15/60 | 5000 | 14000 | 50% |
| CALHOUS | 9 | 5000 | 14640 | 52% |
| COV_TYPE | 54 | 5000 | 25000 | 36% |
| HS | 200 | 5000 | 4366 | 24% |
| LETTER.P1 | 16 | 5000 | 14000 | 3% |
| LETTER.P2 | 16 | 5000 | 14000 | 53% |
| MEDIS | 63 | 5000 | 8199 | 11% |
| MG | 124 | 5000 | 12807 | 17% |
| SLAC | 59 | 5000 | 25000 | 50% |

# A study comparing Classifiers

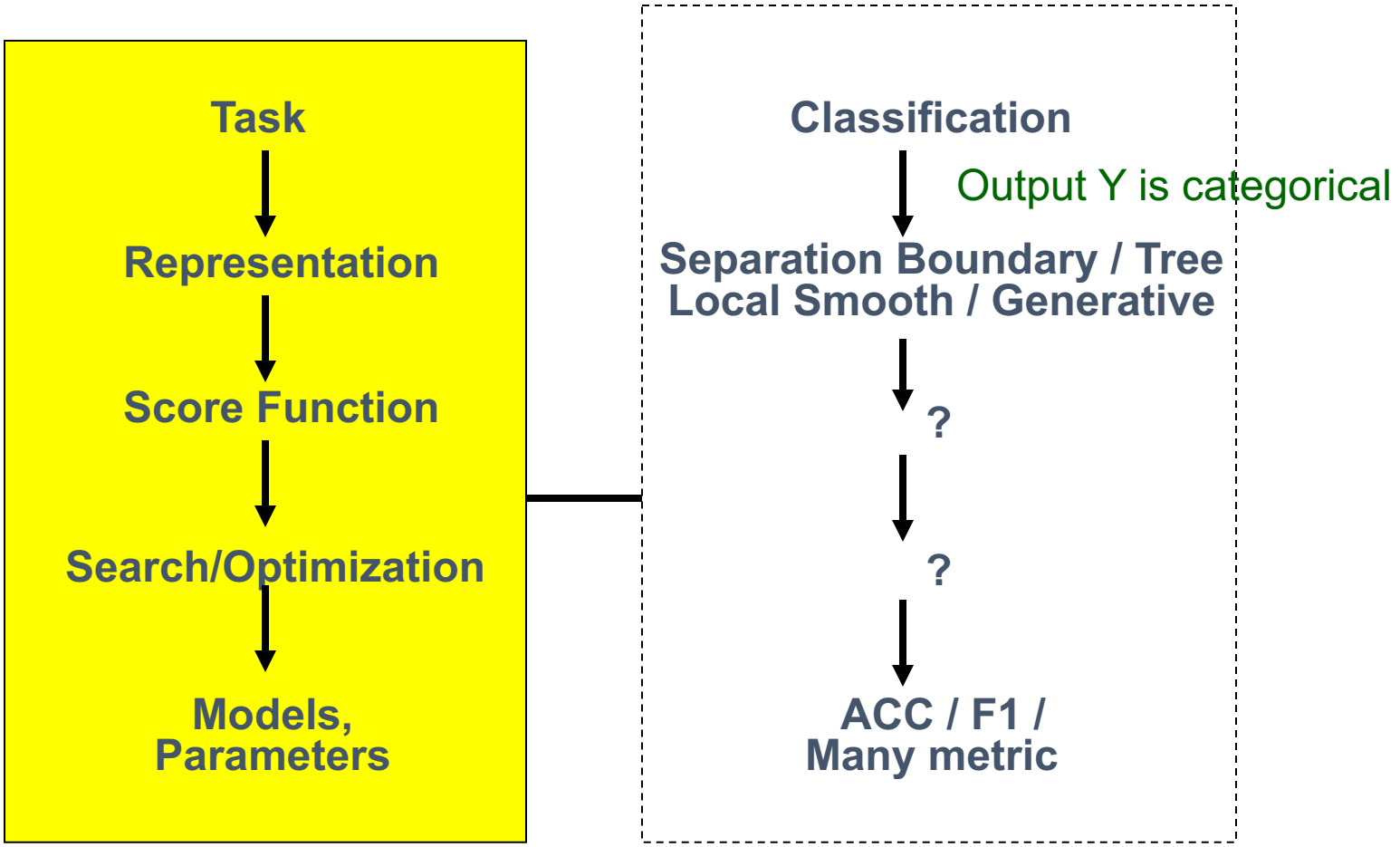
➔ 11 binary classification problems / 8 metrics

Tree, SVM, NN, DeepLearn

*Table 2.* Normalized scores for each learning algorithm by metric (average over eleven problems)

| MODEL | CAL | ACC | FSC | LFT | ROC | APR | BEP | RMS | MXE | MEAN | OPT-SEL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BST-DT | PLT | .843* | .779 | **.939** | **.963** | **.938** | .929* | **.880** | **.896** | **.896** | **.917** |
| RF | PLT | .872* | .805 | .934* | .957 | .931 | **.930** | .851 | .858 | .892 | .898 |
| BAG-DT | – | .846 | .781 | .938* | .962* | .937* | .918 | .845 | .872 | .887* | .899 |
| BST-DT | ISO | .826* | .860* | .929* | .952 | .921 | .925* | .854 | .815 | .885 | .917* |
| RF | – | **.872** | .790 | .934* | .957 | .931 | **.930** | .829 | .830 | .884 | .890 |
| BAG-DT | PLT | .841 | .774 | .938* | .962* | .937* | .918 | .836 | .852 | .882 | .895 |
| RF | ISO | .861* | **.861** | .923 | .946 | .910 | .925 | .836 | .776 | .880 | .895 |
| BAG-DT | ISO | .826 | .843* | .933* | .954 | .921 | .915 | .832 | .791 | .877 | .894 |
| SVM | PLT | .824 | .760 | .895 | .938 | .898 | .913 | .831 | .836 | .862 | .880 |
| ANN | – | .803 | .762 | .910 | .936 | .892 | .899 | .811 | .821 | .854 | .885 |
| SVM | ISO | .813 | .836* | .892 | .925 | .882 | .911 | .814 | .744 | .852 | .882 |
| ANN | PLT | .815 | .748 | .910 | .936 | .892 | .899 | .783 | .785 | .846 | .875 |
| ANN | ISO | .803 | .836 | .908 | .924 | .876 | .891 | .777 | .718 | .842 | .884 |
| BST-DT | – | .834* | .816 | **.939** | **.963** | **.938** | .929* | .598 | .605 | .828 | .851 |
| KNN | PLT | .757 | .707 | .889 | .918 | .872 | .872 | .742 | .764 | .815 | .837 |
| KNN | – | .756 | .728 | .889 | .918 | .872 | .872 | .729 | .718 | .810 | .830 |
| KNN | ISO | .755 | .758 | .882 | .907 | .854 | .869 | .738 | .706 | .809 | .844 |
| BST-STMP | PLT | .724 | .651 | .876 | .908 | .853 | .845 | .716 | .754 | .791 | .808 |
| SVM | – | .817 | .804 | .895 | .938 | .899 | .913 | .514 | .467 | .781 | .810 |
| BST-STMP | ISO | .709 | .744 | .873 | .899 | .835 | .840 | .695 | .646 | .780 | .810 |
| BST-STMP | – | .741 | .684 | .876 | .908 | .853 | .845 | .394 | .382 | .710 | .726 |
| DT | ISO | .648 | .654 | .818 | .838 | .756 | .778 | .590 | .589 | .709 | .774 |

# Supervised Classifiers



| | | actual | |
|---|---|---|---|
| | | + | − |
| PP | predicted+ | *TP* | *FP* |
| PN | predicted− | *FN* | *TN* |

Binary Class {T, F}

$\hat{y}$ vs $y$

- (number of) true positive (TP)
- (number of) true negative (TN)
- (number of) false positive (FP)
- (number of) false negative (FN)

| Metric | Formula | Interpretation |
|---|---|---|
| Accuracy | $\frac{TP + TN}{TP + TN + FP + FN}$ | Overall performance of model |
| Precision | $\frac{TP}{TP + FP}$ | How accurate the positive predictions are |
| Recall<br>Sensitivity | $\frac{TP}{TP + FN}$ | Coverage of actual positive sample |
| Specificity | $\frac{TN}{TN + FP}$ | Coverage of actual negative sample |
| F1 score | $\frac{2TP}{2TP + FP + FN}$ | Hybrid metric useful for unbalanced classes |



Credit: Stanford Machine Learning course

$$\text{Accuracy} = \frac{\#\text{Correct Predicted}}{\#\text{all test Examples}}$$

$$= \frac{TP + TN}{TP + FP + TN + FN}.$$

**(number of) false positive (FP)**

eqv. with false alarm, Type I error

**(number of) false negative (FN)**

eqv. with miss, Type II error

---

**sensitivity or true positive rate (TPR)**

eqv. with hit rate, recall

$$TPR = TP/P = TP/(TP + FN)$$

**specificity (SPC) or true negative rate**

$$SPC = TN/N = TN/(TN + FP)$$

**precision or positive predictive value (PPV)**

$$PPV = TP/(TP + FP)$$

**negative predictive value (NPV)**

$$NPV = TN/(TN + FN)$$

**fall-out or false positive rate (FPR)**

$$FPR = FP/N = FP/(FP + TN) = 1 - SPC$$

**false negative rate (FNR)**

$$FNR = FN/(TP + FN) = 1 - TPR$$

**false discovery rate (FDR)**

$$FDR = FP/(TP + FP) = 1 - PPV$$

---

**accuracy (ACC)**

$$ACC = (TP + TN)/(TP + FP + FN + TN)$$

**F1 score**

is the harmonic mean of precision and sensitivity

$$F1 = 2TP/(2TP + FP + FN)$$

Actual Positive

PP: Predicted Positive

Actual Negative

Recall

# When with Unbalanced Issue (binary case)

Acc Bad

#AP << #AN

- Class imbalance issue

- Balanced accuracy:

| | actual | |
|---|---|---|
| | + | − |
| predicted+ | *TP* | *FP* |
| predicted− | *FN* | *TN* |

# When with Unbalanced Issue (binary case)

- Class imbalance issue

num AP << num AN

actual positive　　actual neg.

- Balanced accuracy:

$$= \frac{1}{2}\left(\frac{TP}{PP} + \frac{TN}{PN}\right)$$

PP: TP+FP　　PN: TN+FN

| | actual | |
|---|---|---|
| | + | − |
| predicted+ | $TP$ | $FP$ |
| predicted− | $FN$ | $TN$ |

AP vs. AN = 1 : 99

① classifier[1]

| | | $y$ AP | $y$ AN |
|---|---|---|---|
| $\hat{y}$ | PP | 0 | 0 |
| $\hat{y}$ | PN | 1 | 99 |

$$ACC = \frac{99}{100} = 99\%$$

$$BACC = \frac{1}{2}\left(\frac{0}{0+\varepsilon} + \frac{99}{100}\right)$$

$$= 49.5\%$$

# Low Ratio of Positive Class (binary case)

If $\frac{\text{Actual P}}{AP + AN}$ very small (e.g. < 1%)

(1, 99)
pos neg

⇒ a classifier can predict every example as Neg

⇒ ①

| | AP ① | AN 99 |
|---|---|---|
| predict P | 0 | 0 |
| predict N | 1 | 99 |

⇒ Accuracy = $\frac{99}{100}$ = 0.99

⇒ Balanced Acc =

Bad - neg - classifier

① Balanced Acc $= \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$

$= \frac{1}{2}\left(\frac{0}{0+\varepsilon} + \frac{99}{100}\right) = 0.495$

[0, 1]

another classifier

② 

| | AP | AN |
|---|---|---|
| PP | 1 | 0 |
| PN | 0 | 99 |

Balanced Acc $= \frac{1}{2}\left(\frac{1}{1} + \frac{99}{99}\right) = 1$

$Acc = \frac{1+99}{1+0+99+0} = 1$

(posRatio 1/100)

③ Third classifier

| | AP | AN |
|---|---|---|
| PP+ | 0 | 1 |
| PN- | 1 | 99 |

$$ACC = \frac{99}{101} \approx 99\%$$

$$BACC = \frac{1}{2}\left(\frac{0}{1} + \frac{99}{100}\right) \approx 0.495$$

---

(pos Ratio 2/120)

④ Fourth case

| | AP | AN |
|---|---|---|
| PP+ | 1 | 19 |
| PN- | 1 | 99 |

$$ACC = \frac{100}{120} \approx 83\%$$

$$BACC = \frac{1}{2}\left(\frac{1}{20} + \frac{99}{100}\right) \approx 0.52$$

# When with Unbalanced Issue (binary case)

**4th case on previous page**

another case: 2 vs. 118 ≈ 1:60

⇒ Balanced Acc cares all classes

⇒ If care more about pos+ class

| | AP | AN |
|---|---|---|
| PP | 1 | 19 |
| PN | 1 | 99 |

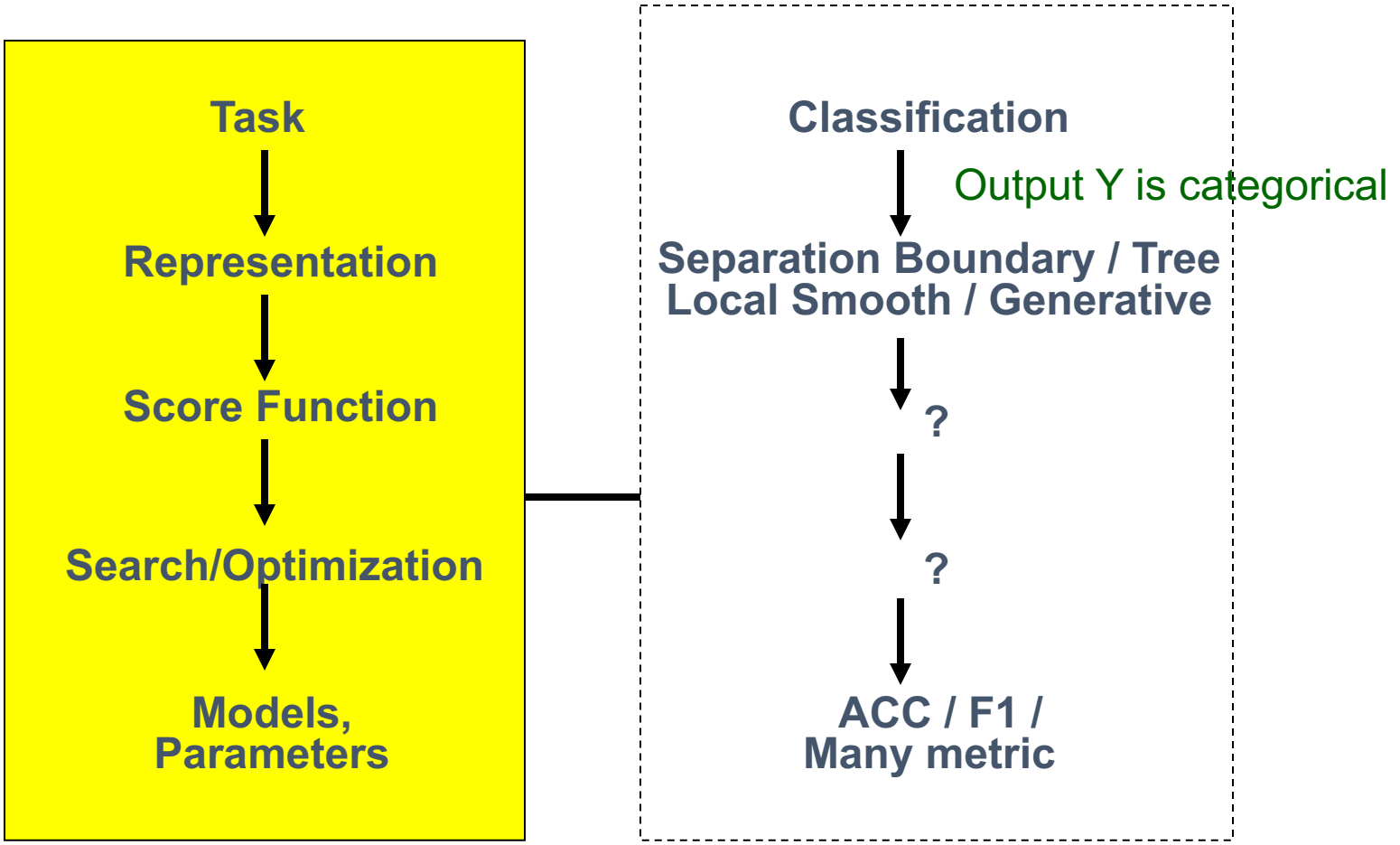
⟹

- Precision = $\frac{TP}{TP+FP}$ = $\frac{1}{20}$ = 5%

| | actual + | actual − |
|---|---|---|
| predicted+ | $TP$ | $FP$ |
| predicted− | $FN$ | $TN$ |

- Recall = $\frac{TP}{TP+FN}$ = $\frac{1}{2}$ = 50%

# Today Recap: Supervised Classifiers



# References

- Big thanks to Prof. Ziv Bar-Joseph and Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Elements of Statistical Learning, by Hastie, Tibshirani and Friedman
- Prof. Andrew Moore @ CMU's slides
- Tutorial slides from Dr. Tie-Yan Liu, MSR Asia