

# UVA CS 4501 - 001 / 6501 – 007

## Introduction to Machine Learning and Data Mining

### Lecture 26: History / Review

Yanjun Qi / Jane, , PhD

University of Virginia  
Department of  
Computer Science

Yanjun Qi / UVA CS 4501-01-6501-07

## Announcements: Rough Plan

- HW5 Grades + Solution / Will be posted in Collab this weekend
- HW6 Grades / Will be posted in Collab this weekend
- **Please check your grades of HW1-6 and the midterm**

# Announcements: Final

- Open Note / Open Book
- No laptop / No Cell phone / No internet access / No electronic devices
- Covering contents after midterm
  - Practice with sample questions in HW6
  - Review course slides carefully

# Today

- History of AI & Machine Learning
- Review of ML methods covered in the course

# What are the goals of AI research?

Artifacts that THINK  
like HUMANS

Artifacts that THINK  
RATIONALLY

Artifacts that ACT  
like HUMANS

Artifacts that ACT  
RATIONALLY

12/4/14

5  
From: M.A. Papalaskar

## A Bit of History

Yanjun Qi / UVA CS 4501-01-6501-07

### 1940s

Advances in **mathematical logic**, **information theory**, **concept of neural computation**

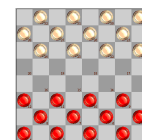
- 🕒 1943: McCulloch & Pitts Neuron
- 🕒 1948: Shannon: Information Theory
- 🕒 1949: Hebbian Learning
  - 🕒 cells that fire together, wire together



### 1950s

**Early computers**. Dartmouth conference **coins the phrase “artificial intelligence”** and Lisp is proposed as the AI programming language

- 🕒 1950: Turing Test
- 🕒 1956: Dartmouth Conference
- 🕒 1958: Friedberg: Learn Assembly Code
- 🕒 1959: Samuel: Learning Checkers



12/4/14

6  
From: M.A. Papalaskar

1960s

A.I. funding increased (mainly military). Famous quote: “Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved.”

Early symbolic reasoning approaches.

- ☉ Logic Theorist, GPS, Perceptrons
- ☉ 1969: Minsky & Papert “Perceptrons”

1970s

A.I. “winter” – Funding dries up as people realize this is a hard problem!

Limited computing power and dead-end frameworks lead to failures.

- ☉ eg: Machine Translation Failure

12/4/14

From: M.A. Papalaskar

1980s

Rule based “expert systems” used in medical / legal professions.

Bio-inspired algorithms (Neural networks, Genetic Algorithms).

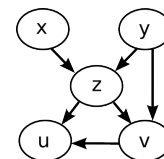
**Again: A.I. promises the world – lots of commercial investment**

Expert Systems (Mycin, Dendral, EMYCIN)

Knowledge Representation and reasoning:

Frames, Eurisko, Cyc, NMR, fuzzy logic

Speech Recognition (HEARSAY, HARPY, HWIM)

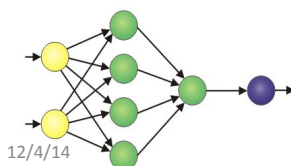


$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Machine Learning:

- ☉ 1982: Hopfield Nets, Decision Trees, GA & GP.

- ☉ 1986: Backpropagation, Explanation-Based Learning



12/4/14

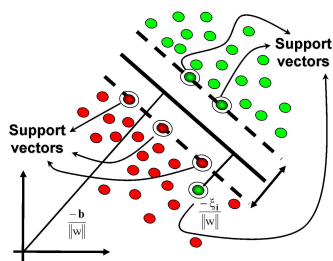
From: M.A. Papalaskar

1990s

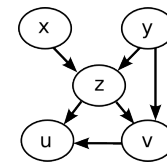
Some concrete successes begin to emerge. AI diverges into separate fields: Computer Vision, Automated Reasoning, Planning systems, Natural Language processing, **Machine Learning**...

...Machine Learning begins to overlap with statistics / probability theory.

- 1992: Koza & Genetic Programming
- 1995: Vapnik: Support Vector Machines



12/4/14



$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

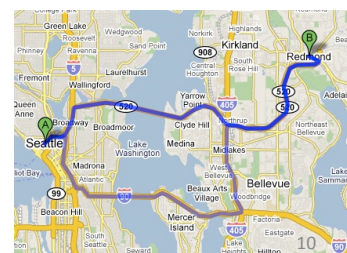
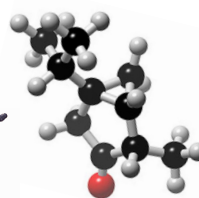
9

From: M.A. Papalaskar

2000s

First commercial-strength applications: Google, Amazon, computer games, route-finding, credit card fraud detection, spam filters, etc...

Tools adopted as standard by other fields e.g. biology



12/4/14

From: M.A. Papalaskar

2010s.... ??????

<b>Deep Learning</b> With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.	<b>Temporary Social Media</b> Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.	<b>Prenatal DNA Sequencing</b> Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?	<b>Additive Manufacturing</b> Skeptical about 3-D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts.	<b>Baxter: The Blue-Collar Robot</b> Rodney Brooks's newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.
<b>Memory Implants</b> A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss. 12/4/14	<b>Smart Watches</b> The designers of the Pebble watch realized that a mobile phone is more useful if you don't have to take it out of your pocket.	<b>Ultra-Efficient Solar Power</b> Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.	<b>Big Data from Cheap Phones</b> Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases.	<b>Supergrids</b> A new high-power circuit breaker could finally make highly efficient DC power grids practical. 11

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

- Able to
  - perceive the world
  - understand the world
- This needs
  - Basic speech capabilities
  - Basic vision capabilities
  - Language understanding
  - User behavior / emotion understanding
  - Able to think ??

# Plenty of Data

- **Text**: trillions of words of English + other languages
- **Visual**: billions of images and videos
- **Audio**: thousands of hours of speech per day
- **User activity**: queries, user page clicks, map requests, etc,
- **Knowledge graph**: billions of labeled relational triplets
- .....

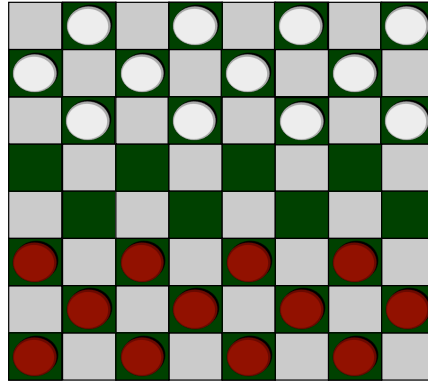
Data-driven machine learning methods have made machines / computers much more intelligent

## Detour: our programming assignments

- HW3: Semantic **language understanding** (sentiment classification on movie review text)
- HW5: **Visual object recognition** (labeling images about handwritten digits)
- Planned but omitted: **Audio speech recognition** (HMM based speech recognition task )

## Samuel's definition of ML (1959)

- Arthur Samuel (1959). Machine Learning: Field of study that **gives computers the ability to learn without being explicitly programmed.**



12/4/14

15

## Tom Mitchell (1998): Well-posed Learning Problem

*A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**.*

12/4/14

16



# Defining the Learning Task

Improve on task, T, with respect to performance metric, P, based on experience, E.

T: Playing checkers

P: Percentage of games won against an arbitrary opponent

E: Playing practice games against itself

T: Recognizing hand-written words

P: Percentage of words correctly classified

E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors

P: Average distance traveled before a human-judged error

E: A sequence of images and steering commands recorded while observing a human driver.

T: Determine which students like oranges or apples

P: Percentage of students' preferences guessed correctly

E: Student attribute data

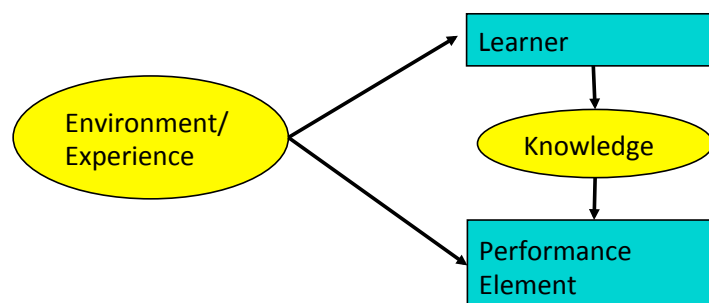
12/4/14

17

From: M.A. Papalaskar

# Designing a Learning System

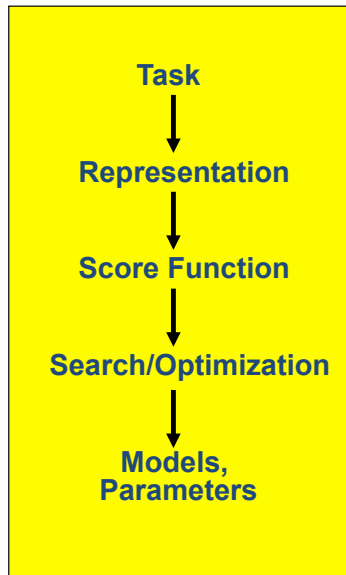
- Choose the **training experience**
- Choose exactly what is to be learned, i.e. the **target function**.
- Choose a **learning algorithm** to infer the target function from the experience.
- A learning algorithm will also determine a **performance measure**



18

12/4/14

## Machine Learning in a Nutshell



ML grew out of work in AI

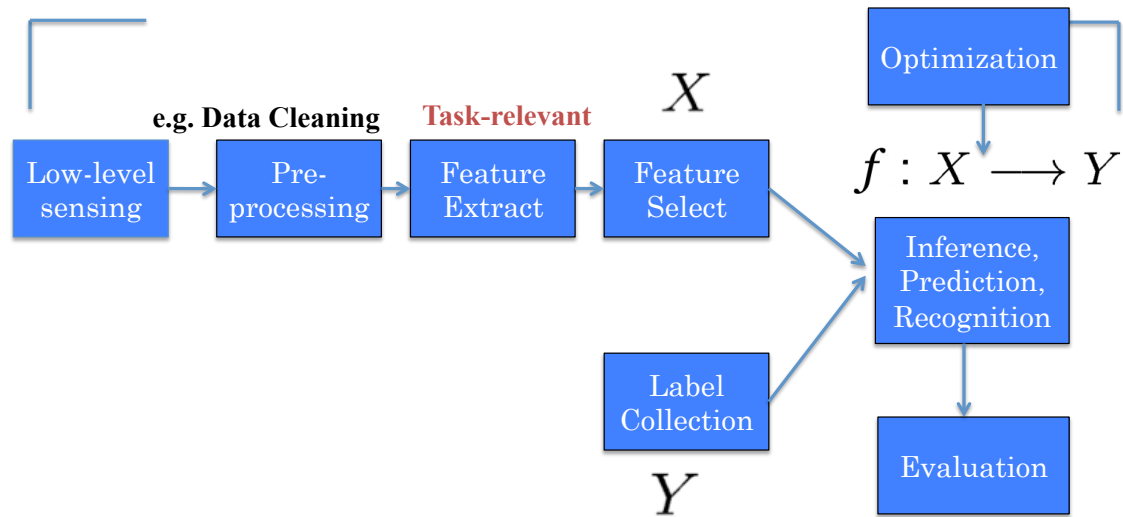
*Optimize a performance criterion using example data or past experience,*

*Aiming to generalize to unseen data*

## What we have covered for each component

<b>Task</b>	Regression, classification, clustering, dimen-reduction
<b>Representation</b>	Linear func, nonlinear function (e.g. polynomial expansion), local linear, logistic function (e.g. $p(c x)$ ), tree, multi-layer, prob-density family (e.g. Bernoulli, multinomial, Gaussian, mixture of Gaussians), local func smoothness,
<b>Score Function</b>	MSE, Hinge, log-likelihood, EPE (e.g. L2 loss for KNN, 0-1 loss for Bayes classifier), cross-entropy, cluster points distance to centers, variance,
<b>Search/ Optimization</b>	Normal equation, gradient descent, stochastic GD, Newton, Linear programming, Quadratic programming (quadratic objective with linear constraints), greedy, EM, asyn-SGD, eigenDecomp
<b>Models, Parameters</b>	Regularization (e.g. L1, L2)

## A Typical Machine Learning Pipeline



12/4/14

21

## Today

- History of Machine Learning & AI
- Review of ML methods covered in the course

12/4/14

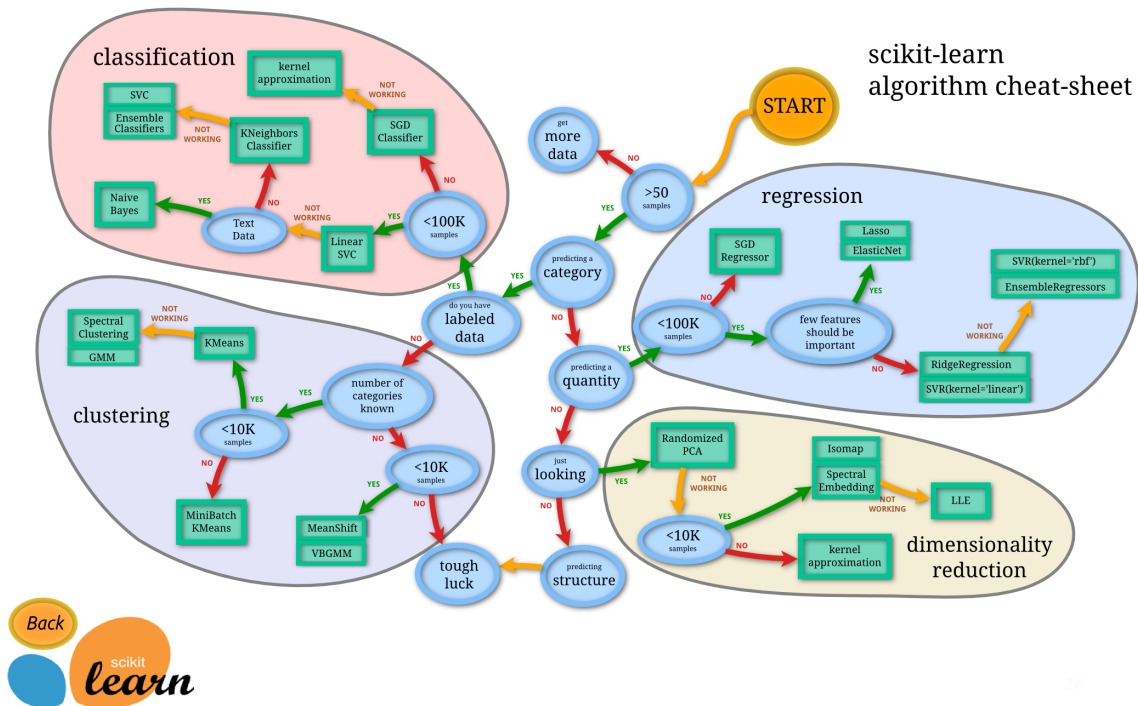
22

# Where are we ? → major sections of this course

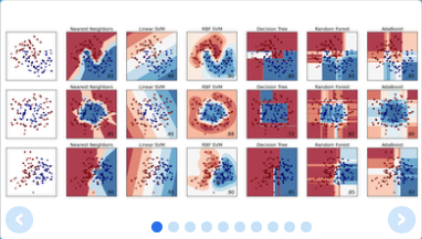
- ❑ Regression (supervised)
- ❑ Classification (supervised)
- ❑ Unsupervised models
- ❑ Learning theory
- ❑ ~~Graphical models~~

[http://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/](http://scikit-learn.org/stable/tutorial/machine_learning_map/)

# Scikit-learn algorithm cheat-sheet



<http://scikit-learn.org/stable/>



# scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## Classification

Identifying to which set of categories a new observation belong to.

**Applications:** Spam detection, Image recognition.

**Algorithms:** *SVM, nearest neighbors, random forest, ...* — Examples

## Regression

Predicting a continuous value for a new example.

**Applications:** Drug response, Stock prices.

**Algorithms:** *SVR, ridge regression, Lasso, ...* — Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** *k-Means, spectral clustering, mean-shift, ...* — Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** *PCA, feature selection, non-negative matrix factorization.* — Examples

## Model selection

Comparing, validating and choosing parameters and models.

**Goal:** Improved accuracy via parameter tuning

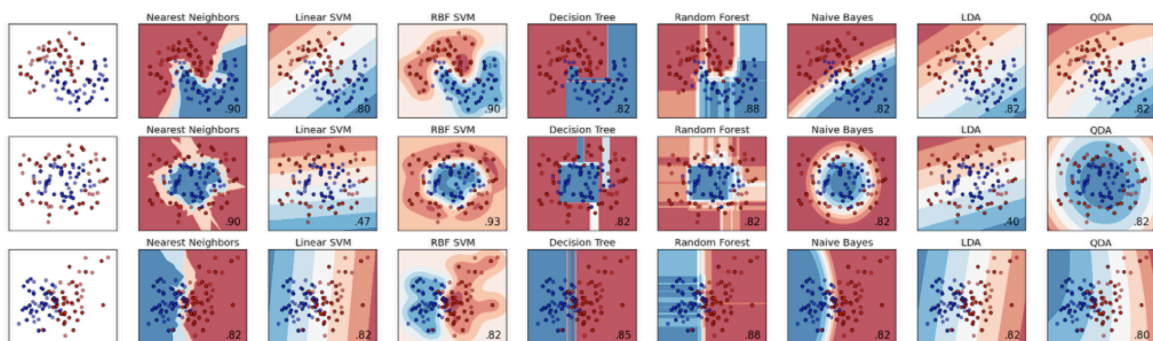
**Modules:** *grid search, cross validation, metrics.* — Examples

## Preprocessing

Feature extraction and normalization.

**Application:** Transforming input data such as text for use with machine learning algorithms.

**Modules:** *preprocessing, feature extraction.* — Examples



- ✓ different assumptions on data
- ✓ different scalability profiles at training time
- ✓ different latencies at prediction time
- ✓ different model sizes (embedability in mobile devices)

# What we have covered (I)

## □ Supervised Regression models

- Linear regression (LR)
- LR with non-linear basis functions
- Locally weighted LR
- LR with Regularizations

12/4/14

27

	$X_1$	$X_2$	$X_3$	$Y$
$S_1$				
$S_2$				
$S_3$				
$S_4$				
$S_5$				
$S_6$				

## A Dataset

$$f : X \rightarrow Y$$

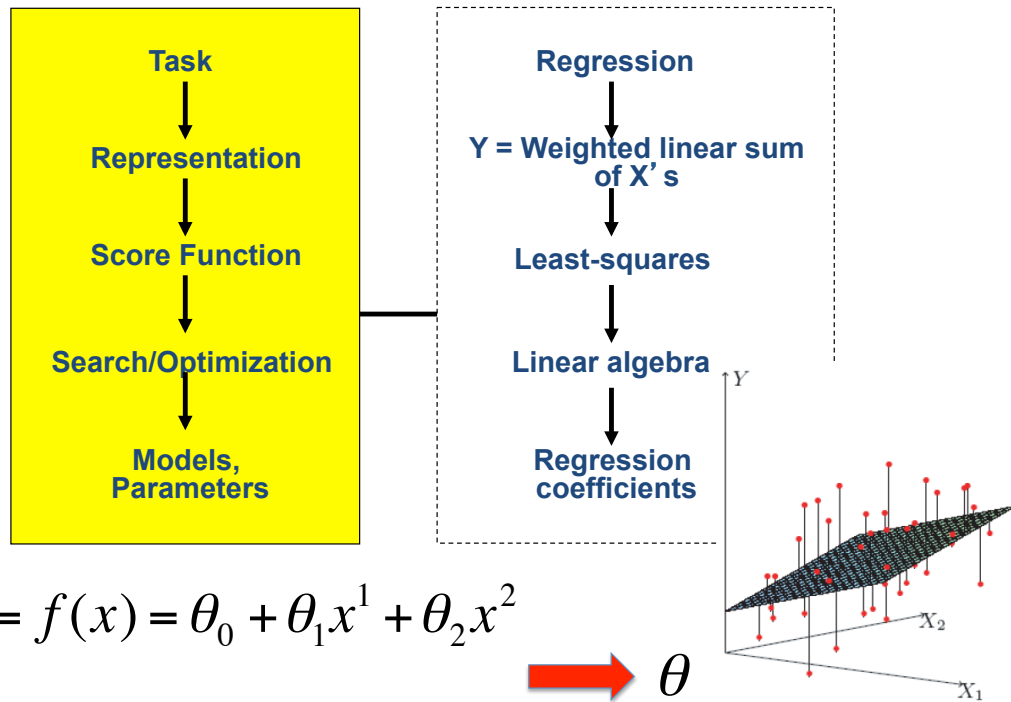
Output Y as  
continuous values

- **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 ]

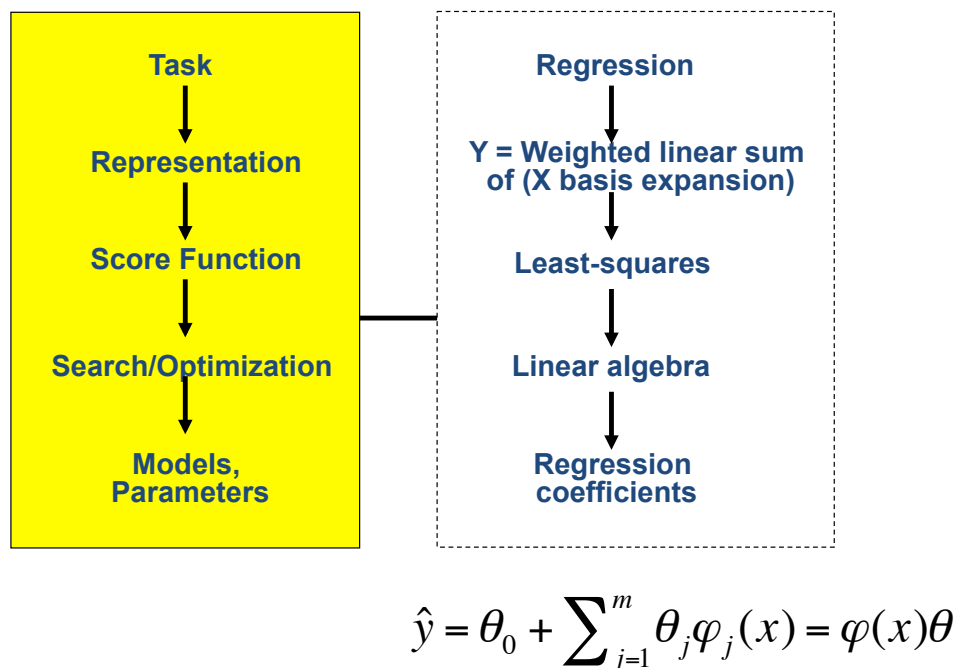
12/4/14

28

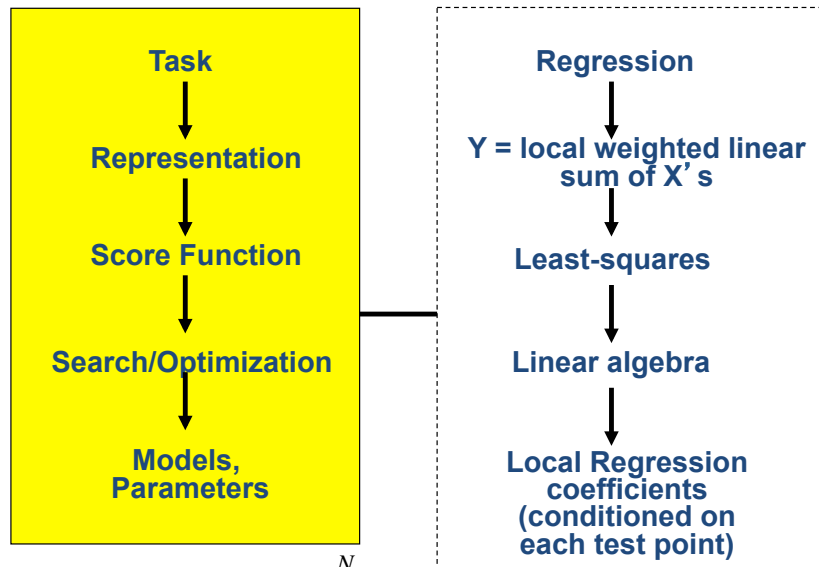
## (1) Multivariate Linear Regression



## (2) Multivariate Linear Regression with basis Expansion



### (3) Locally Weighted / Kernel Regression



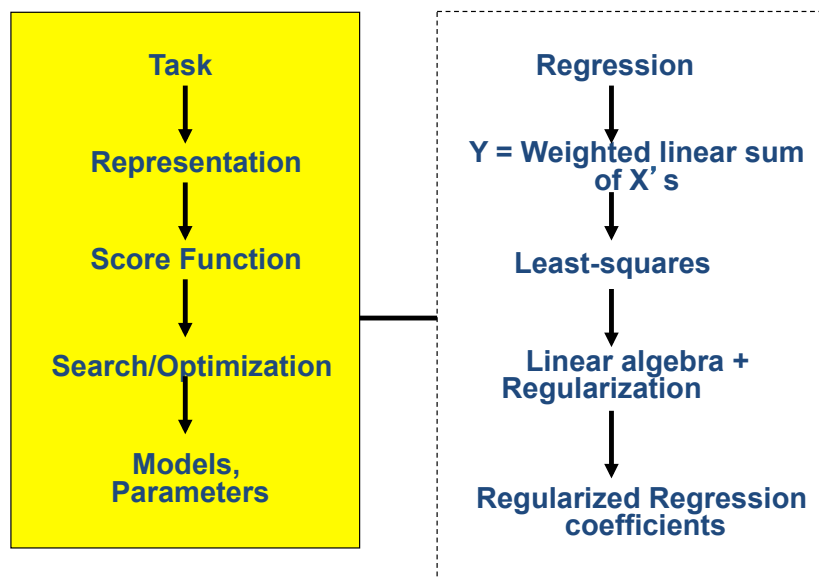
$$\min_{\alpha(x_0), \beta(x_0)} \sum_{i=1}^N K_{\lambda}(x_i, x_0) [y_i - \alpha(x_0) - \beta(x_0)x_i]^2$$

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$$

12/4/14

31

### (4) Regularized multivariate linear regression



$$\min J(\beta) = \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

12/4/14

32



## What we have covered (II)

### □ Supervised Classification models

- Support Vector Machine
- Bayes Classifier
- Logistic Regression
- K-nearest Neighbor
- Random forest / Decision Tree
- Neural Network (e.g. MLP)
- \*Feature selection

12/4/14

33

$X_1$	$X_2$	$X_3$	$C$

### A Dataset for classification

$$f : X \rightarrow C$$

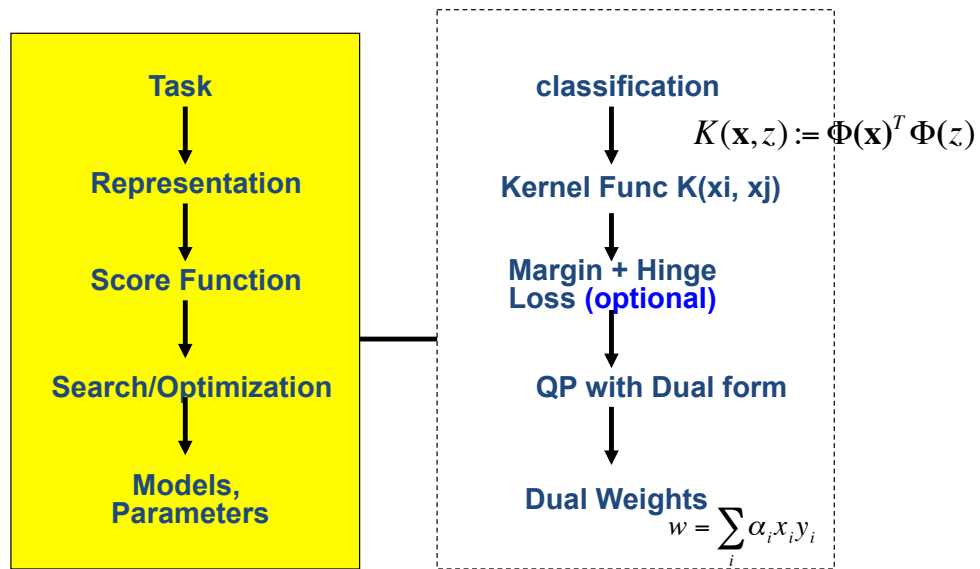
Output as Discrete  
Class Label  
 $C_1, C_2, \dots, C_L$

- **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 ]

12/4/14

34

## (1) Support Vector Machine



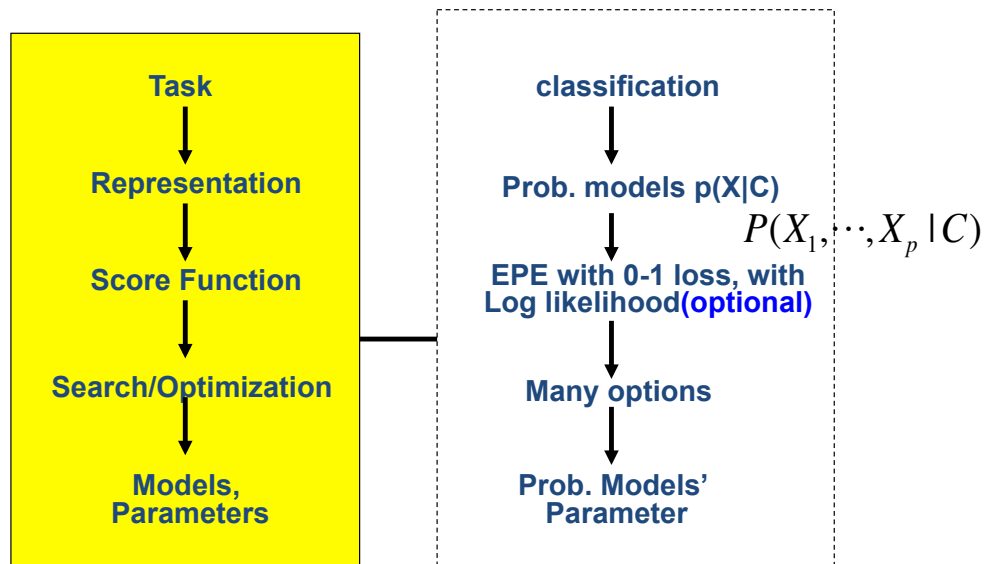
$$\operatorname{argmin}_{w,b} \sum_{i=1}^p w_i^2 + C \sum_{i=1}^n \varepsilon_i$$

$$\text{subject to } \forall x_i \in D_{\text{train}} : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \varepsilon_i$$

12/4/14

$$\operatorname{argmax}_k P(C = k | X) = \operatorname{argmax}_k P(X, C) = \operatorname{argmax}_k P(X | C) P(C)$$

## (2) Bayes Classifier



Bernoulli Naive  $p(W_i = \text{true} | c_k) = p_{i,k}$

12/4/14

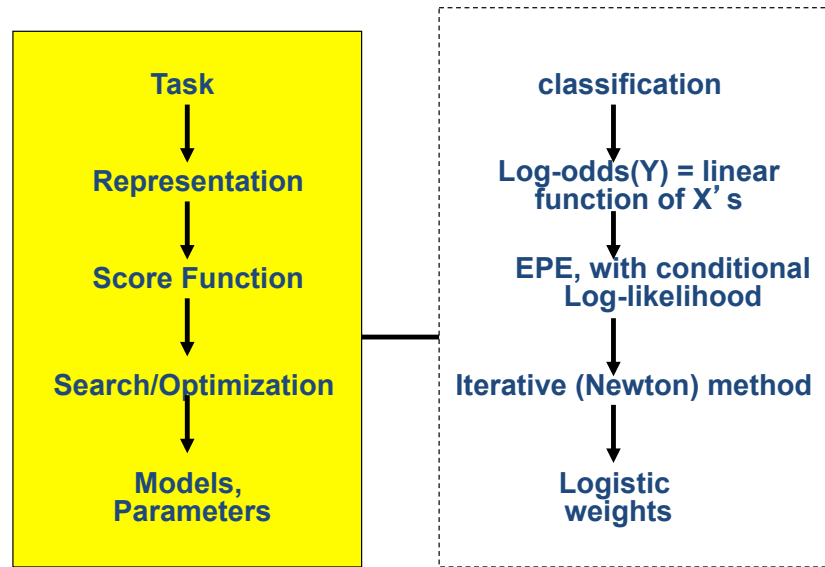
Gaussian Naive

$$\hat{P}(X_j | C = c_k) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left(-\frac{(X_j - \mu_{jk})^2}{2\sigma_{jk}^2}\right)$$

Multinomial

$$P(W_1 = n_1, \dots, W_v = n_v | c_k) = \frac{N!}{n_{1k}! n_{2k}! \dots n_{vk}!} \theta_{1k}^{n_{1k}} \theta_{2k}^{n_{2k}} \dots \theta_{vk}^{n_{vk}}$$

### (3) Logistic Regression

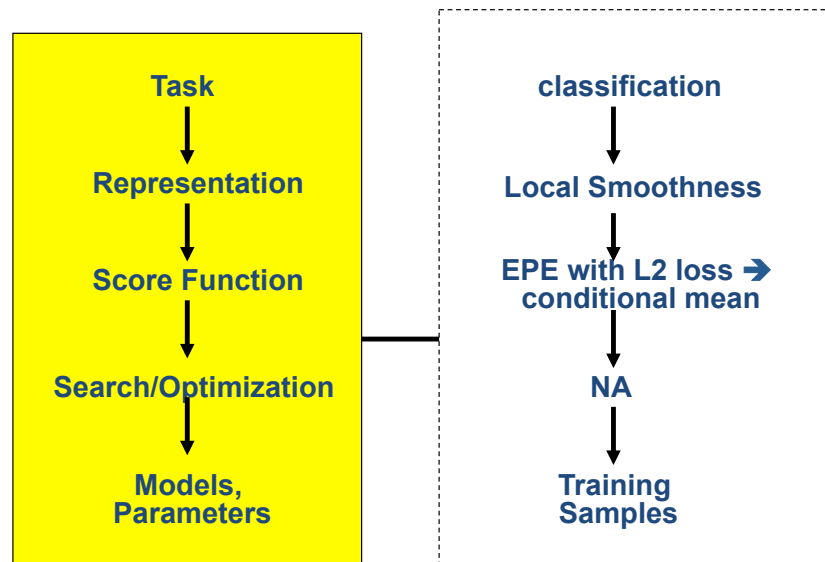


$$P(c = 1|x) = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}}$$

12/4/14

37

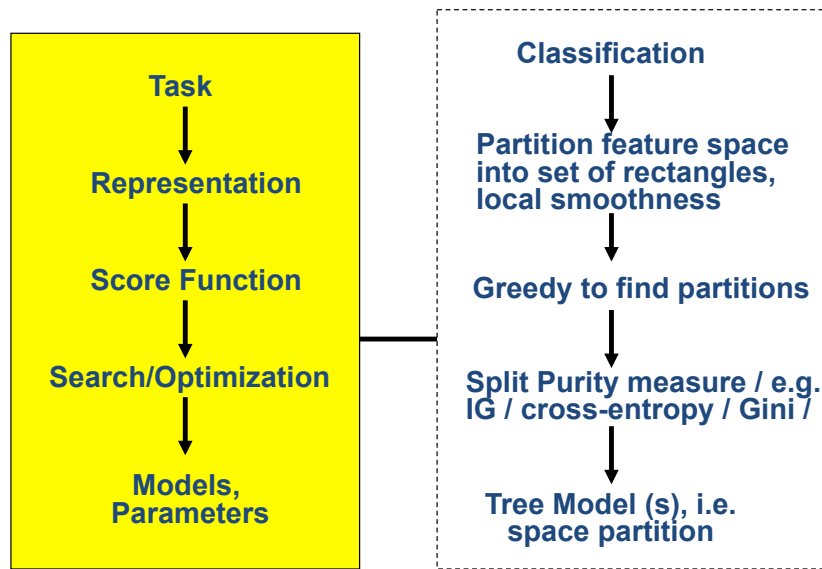
### (4) K-Nearest Neighbor



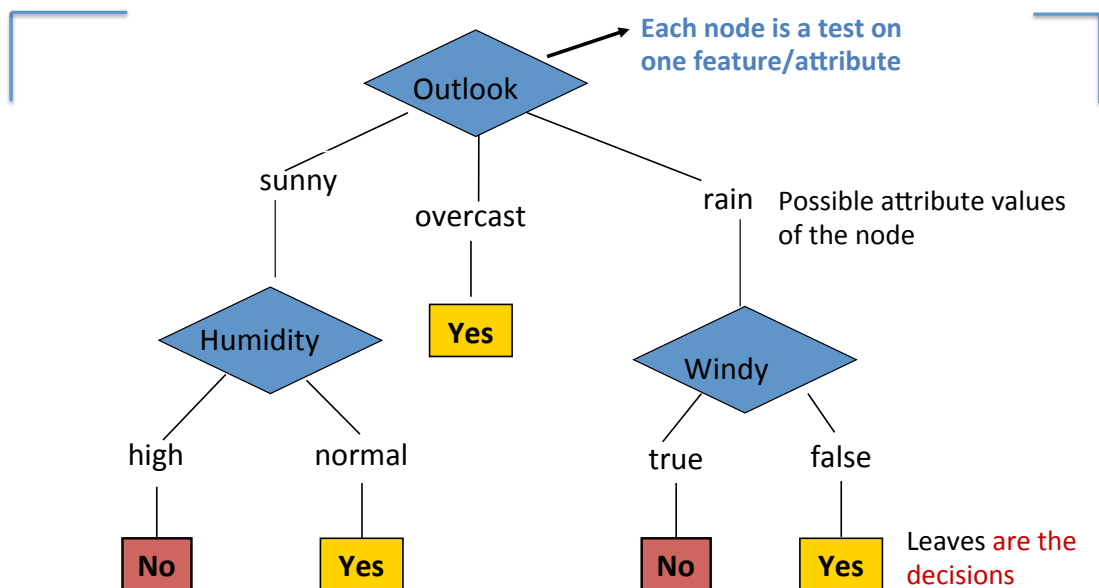
12/4/14

38

## (5) Decision Tree / Random Forest



## Anatomy of a decision tree



# Decision trees

- Decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances.

```

• (Outlook ==overcast)
• OR
• ((Outlook==rain) and (Windy==false))
• OR
• ((Outlook==sunny) and (Humidity=normal))
• => yes play tennis
  
```

# Information gain

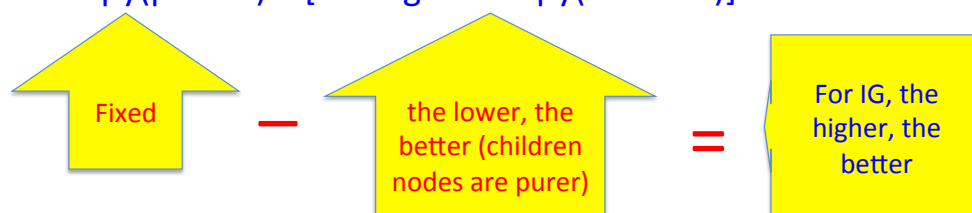
- $IG(X_i, Y) = H(Y) - H(Y|X_i)$

Reduction in uncertainty by knowing a feature  $X_i$

Information gain:

= (information before split) – (information after split)

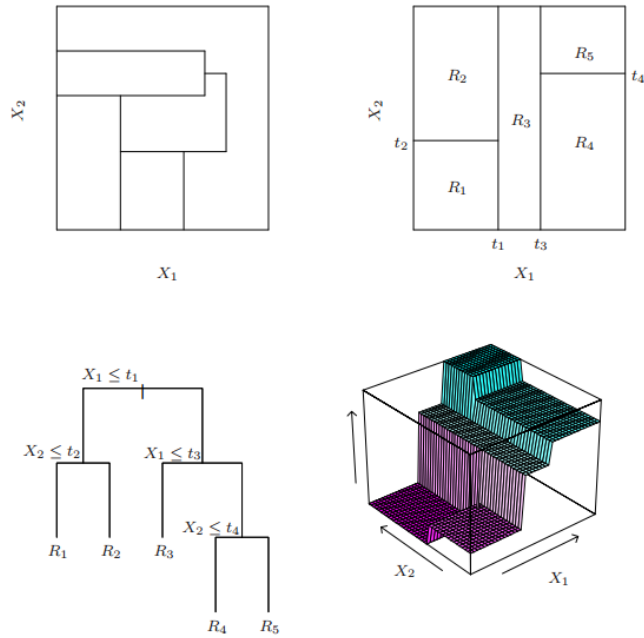
= entropy(parent) – [average entropy(children)]



From ESL book Ch9 :

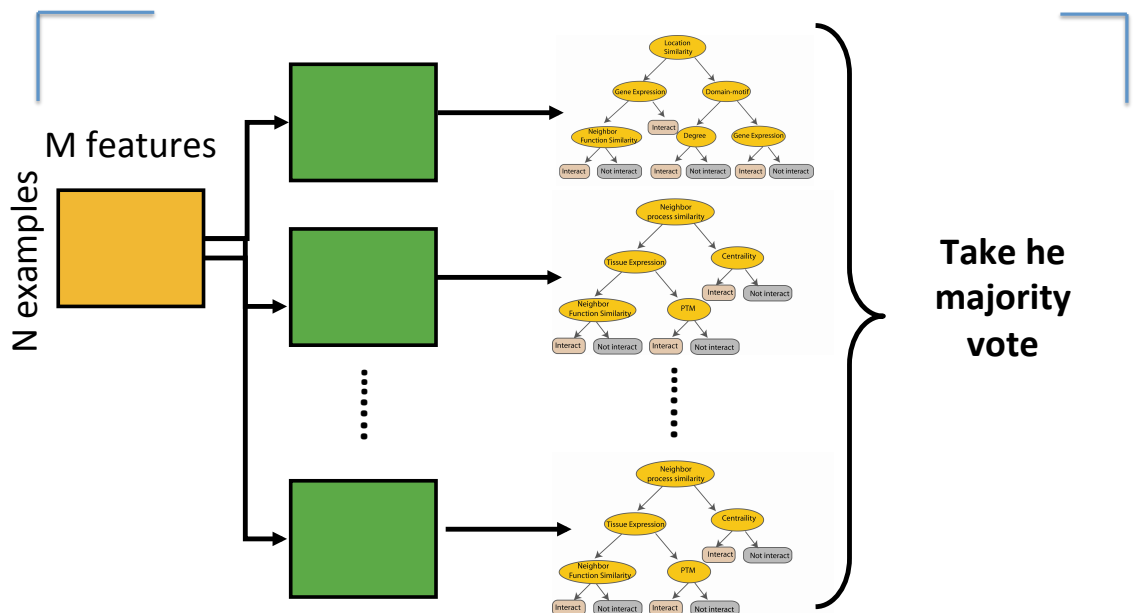
**Classification and Regression Trees (CART)**

- **Partition feature space into set of rectangles**
- Fit simple model in each partition

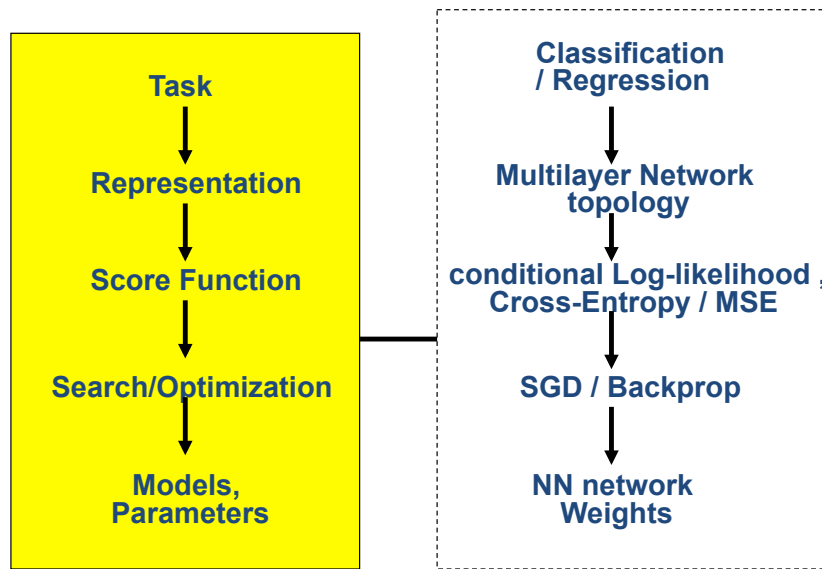


**FIGURE 9.2.** Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.

# Random Forest Classifier



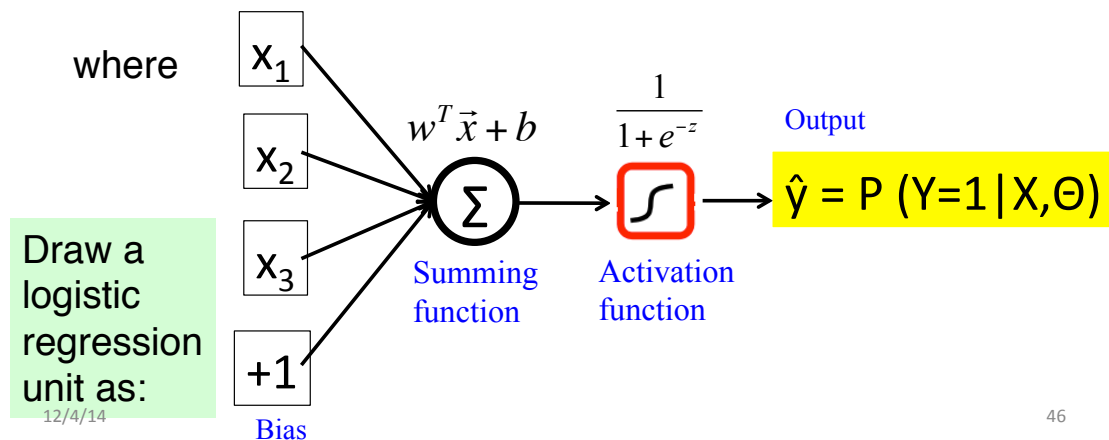
## (6) Neural Network



# Logistic regression

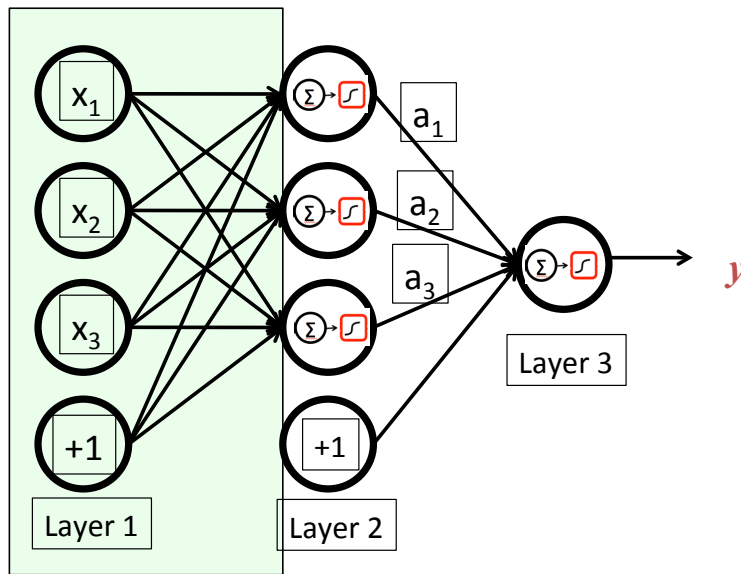
Logistic regression could be illustrated as a module

On input  $x$ , it outputs  $\hat{y}$ :



# Multi-Layer Perceptron (MLP)

String a lot of logistic units together. Example: 3 layer network:

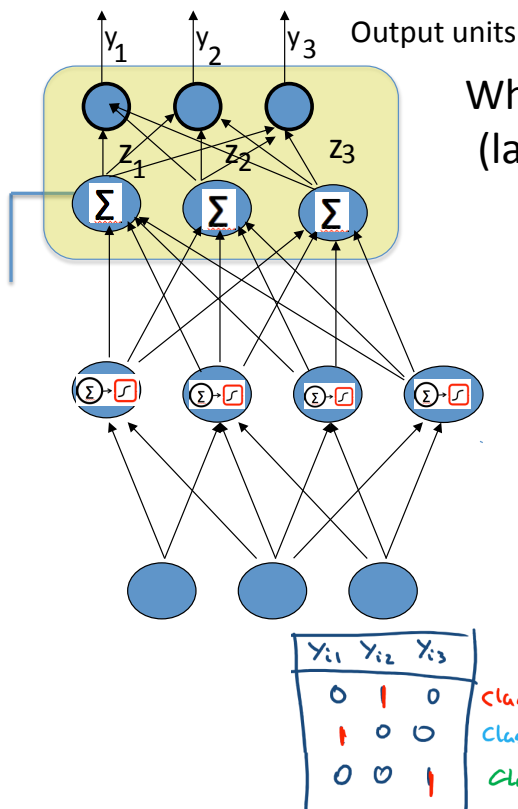


12/4/14 input

hidden

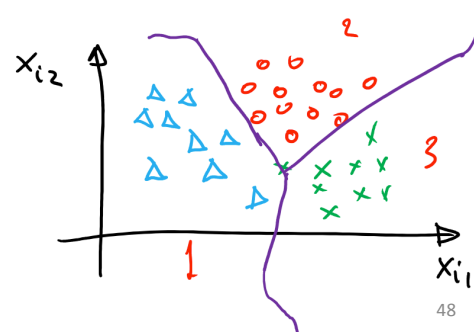
output

47



When for multi-class classification (last output layer: softmax layer)

When multi-class output, last layer is softmax output layer → a multinomial logistic regression unit



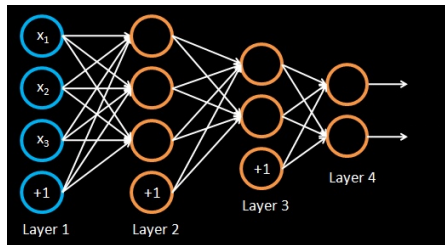
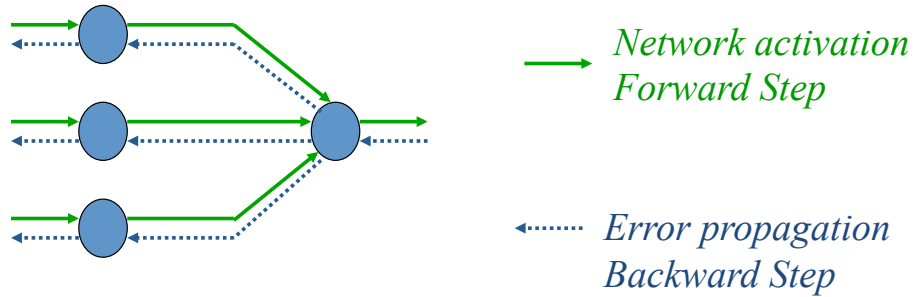
12/4/14

48



# Backpropagation

- Back-propagation training algorithm



12/4/14

49

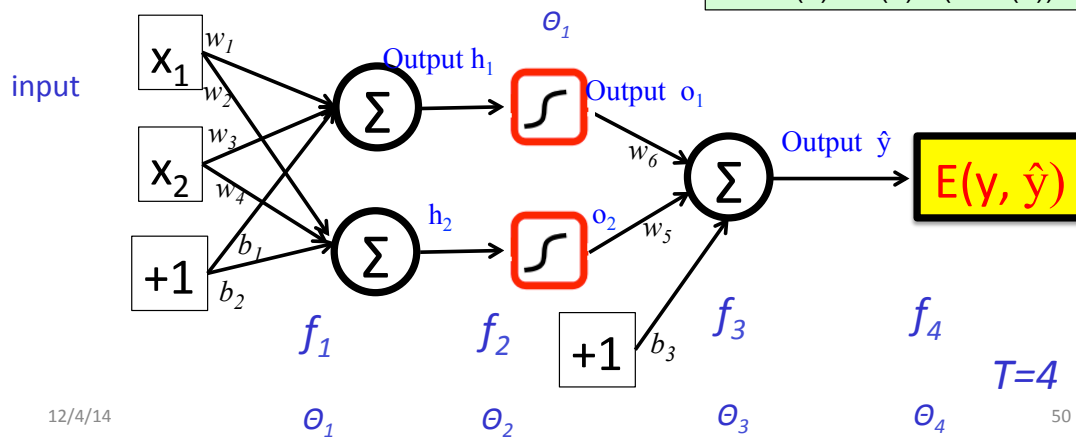
to train this layered network. The stacked layers in our network can be written in a more general form of multi-level functions:

$$l_{\mathbf{x}} = \mathbf{f}_T(\mathbf{f}_{T-1}(\dots(\mathbf{f}_1(\mathbf{x}))\dots)),$$

where  $l_{\mathbf{x}}$  denotes the loss on a single example  $\mathbf{x}$

For instance → for regression

for sigmoid unit, its derivative is,  $o'(h) = o(h) * (1 - o(h))$



12/4/14

50

$\mathbf{f}_i, i \in [1, T]$ , the derivative for updating its parameter set  $\theta_i$  is using the delta rule:

$$\frac{\partial l}{\partial \theta_i} = \frac{\partial f_T}{\partial f_i} \times \frac{\partial f_i}{\partial \theta_i},$$

and the first factor on the right can be recursively calculated:

$$\frac{\partial f_T}{\partial f_i} = \frac{\partial f_T}{\partial f_{i+1}} \times \frac{\partial f_{i+1}}{\partial f_i}.$$

Note that  $\mathbf{f}$  and  $\theta$  are usually vectors

so  $\frac{\partial f_T}{\partial f_{i+1}}$  and  $\frac{\partial f_i}{\partial \theta_i}$  are Jacobian matrices, and “ $\times$ ” is matrix multiplication.

e.g.

$$\frac{\partial f_4}{\partial f_3} = \frac{\partial (y - \hat{y})^2}{\partial f_3} = \frac{\partial (y - \hat{y})}{\partial f_3} = \frac{\partial (y - \hat{y})}{\partial f_3}$$

output error

12/4/14

Dr. Qi's CIKM 2012 paper/talk <sup>51</sup>

**for**  $j = 1$  to MaxIter **do**

**if** converge **then**

    break

**end if**

$\mathbf{x}, y \leftarrow$  random sampled data point and label

  calculate loss  $l(\mathbf{x}; y)$

  cumulative  $\leftarrow 1$

**for**  $i = T$  to 1 **do**

$$\frac{\partial l}{\partial \theta_i} \leftarrow \text{cumulative} * \frac{\partial f_i}{\partial \theta_i}$$

$$\theta_i \leftarrow \theta_i - \lambda \frac{\partial l}{\partial \theta_i}$$

$$\text{cumulative} \leftarrow \text{cumulative} * \frac{\partial f_{i+1}}{\partial f_i}$$

**end for**

**end for**

Yanjun Qi / UVA CS 4501-01-6501-07

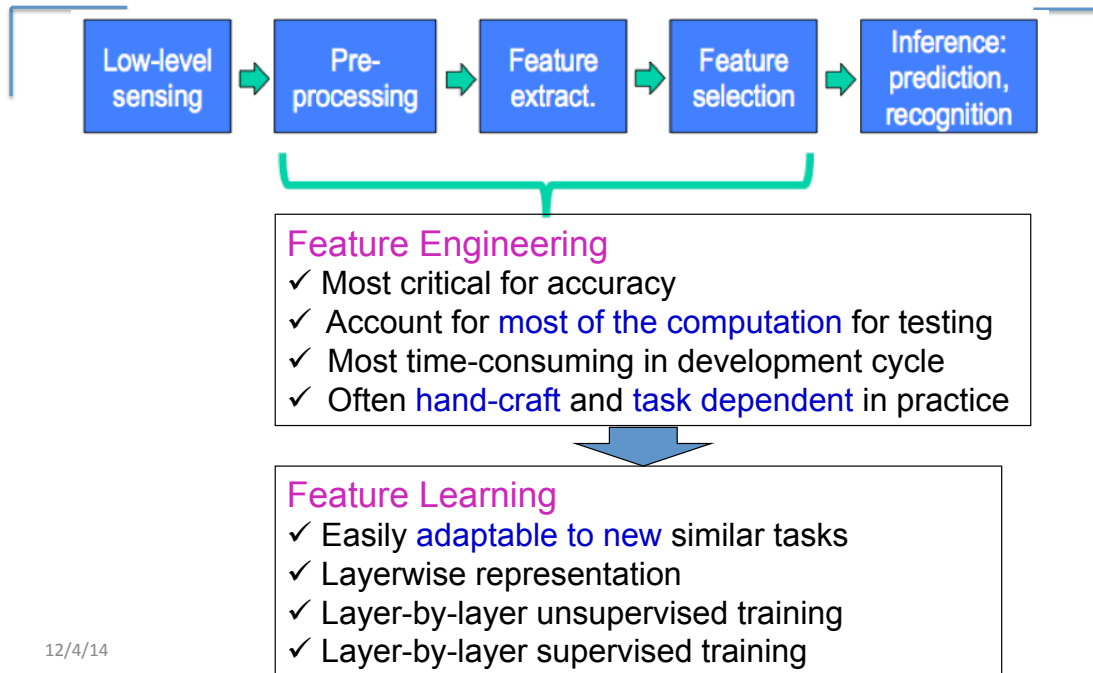
*Error propagation  
Backward Step*



12/4/14

Dr. Qi's CIKM 2012 paper/talk <sup>52</sup>

## Deep Learning Way: Learning features / Representation from data



12/4/14

## DESIGN ISSUES for Deep NN

- Data representation
- Network Topology
- Network Parameters
- Training
  - Scaling up with **graphics processors**
  - Scaling up with **Asynchronous SGD**
- Validation

12/4/14

54

# ImageNet Challenge 2014

- In the mean time Pierre Sermanet had joined other people from Google Brain
- Monster model: GoogLeNet now at **6.7% error rate**

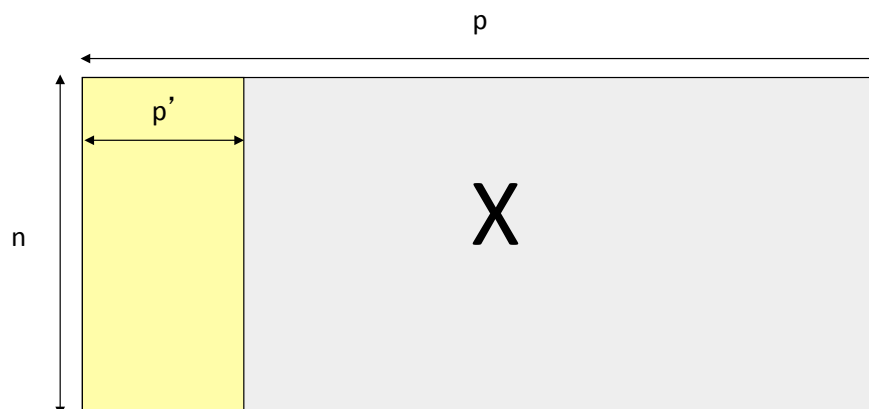


Dr. Jeff Dean's talk

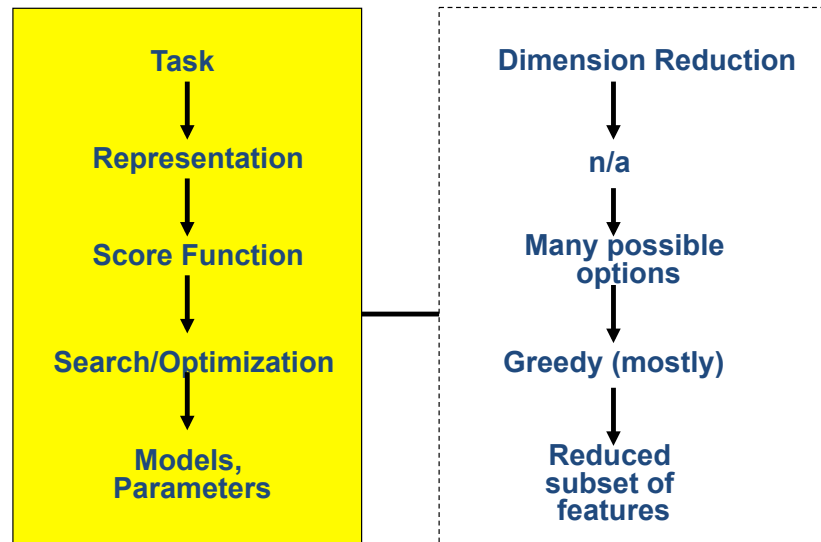
Olivier Grisel's talk

## (7) Feature Selection

- **Thousands to millions of low level features:** select the most relevant one to build **better, faster, and easier to understand** learning machines.



## (7) Feature Selection



12/4/14

57

## Feature Selection

- **Filtering approach:**  
ranks features or feature subsets independently of the predictor (classifier).
  - ...using univariate methods: consider **one** variable at a time
  - ...using multivariate methods: consider **more than one** variables at a time
- **Wrapper approach:**  
uses a classifier to assess (many) features or feature subsets.
- **Embedding approach:**  
uses a classifier to build a (single) model with a subset of features that are internally selected.

12/4/14

58/54

## What we have covered (III)

### □ Unsupervised models

- Dimension Reduction (PCA)
- Hierarchical clustering
- K-means clustering
- GMM/EM clustering

12/4/14

59

	$X_1$	$X_2$	$X_3$
$S_1$			
$S_2$			
$S_3$			
$S_4$			
$S_5$			
$S_6$			

## An unlabeled Dataset X

a data matrix of  $n$  observations on  $p$  variables  $x_1, x_2, \dots, x_p$

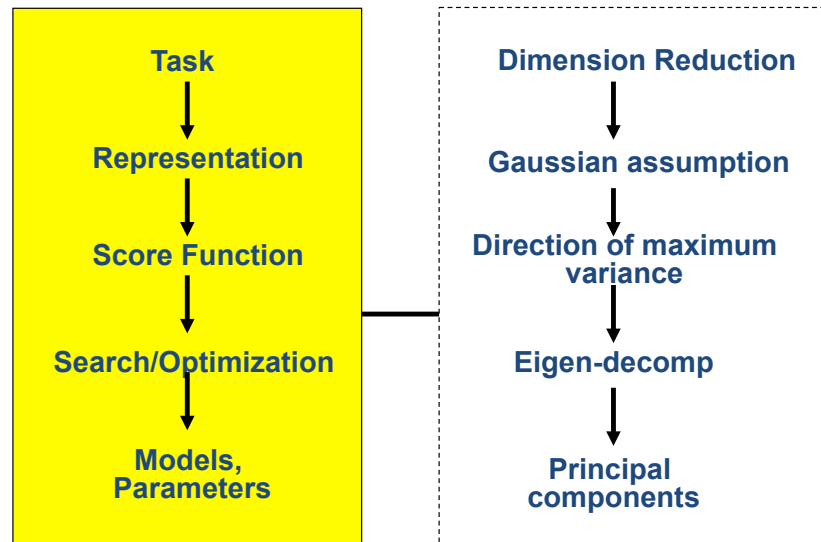
**Unsupervised learning** = learning from raw (unlabeled, unannotated, etc) data, as opposed to supervised data where a label of examples is given

- **Data/points/instances/examples/samples/records:** [ rows ]
- **Features/attributes/dimensions/independent variables/covariates/predictors/regressors:** [ columns ]

12/4/14

60

## (0) Principal Component Analysis

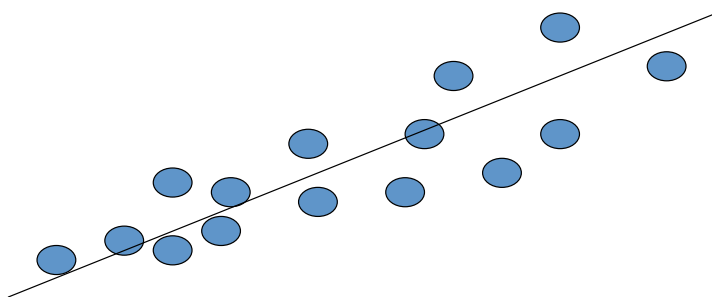


12/4/14

61

## Algebraic Interpretation – 1D

- Given  $n$  points in a  $p$  dimensional space, for large  $p$ , how does one project on to a 1 dimensional space?



- Choose a line that fits the data so **the points are spread out well along the line**

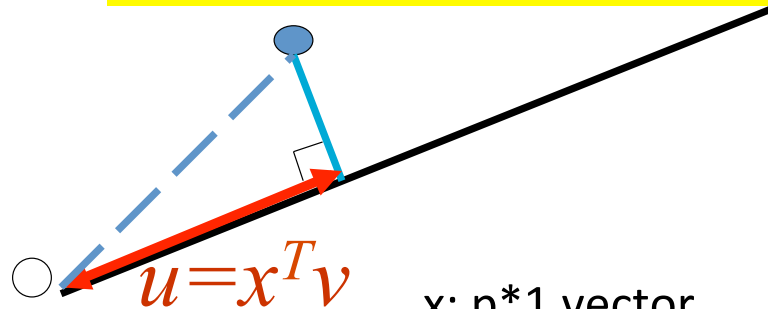
12/4/14

From Dr. S. Narasimhan<sup>62</sup>

## Algebraic Interpretation – 1D

- Minimizing sum of squares of distances to the line is the same as maximizing the sum of squares of the projections on that line, thanks to Pythagoras.

$$\max(v^T X^T X v), \text{ subject to } v^T v = 1$$



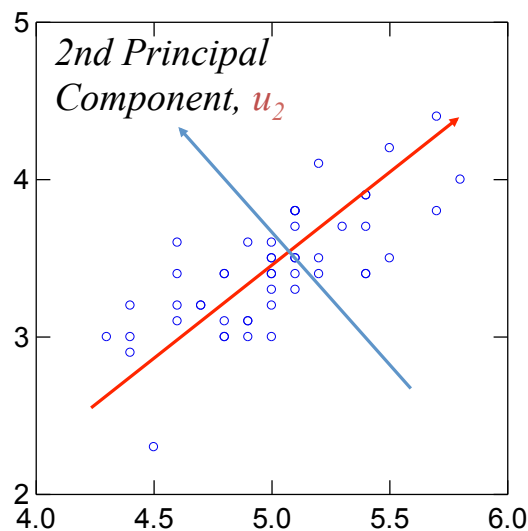
$x$ :  $p \times 1$  vector  
 $v$ :  $p \times 1$  vector

assuming data  
is centered

12/4/14

63

## PCA Eigenvectors → Principal Components



*1st Principal  
Component,  $u_1$*

12/4/14

64



# PCA & Gaussian Distributions.

- PCA is similar to learning a Gaussian distribution for the data.
- Dimension reduction occurs **by ignoring the directions in which the covariance is small.**

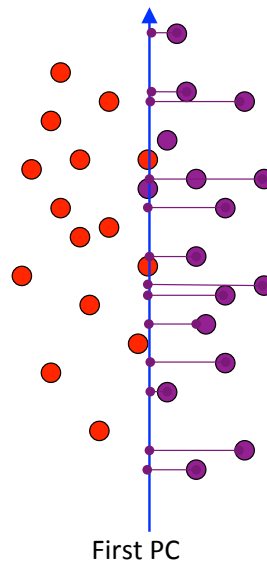
12/4/14

65

# PCA Limitations

- The direction of maximum variance is not always good for classification

not ideal for discrimination



12/4/14

From Prof. Derek Hoiem<sup>66</sup>

## What we have covered (III)

### □ Unsupervised models

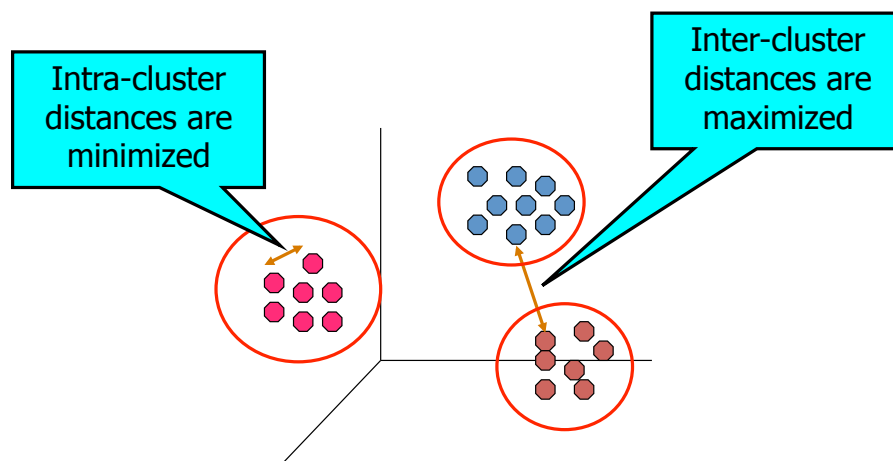
- Dimension Reduction (PCA)
- Hierarchical clustering
- K-means clustering
- GMM/EM clustering

12/4/14

67

## What is clustering?

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups



12/4/14

68

# Issues for clustering

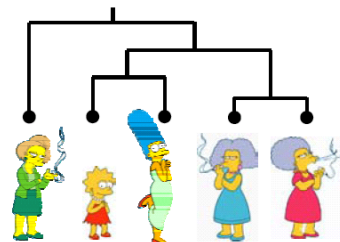
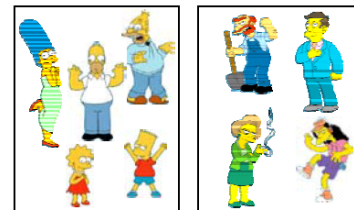
- What is a natural grouping among these objects?
  - Definition of "groupness"
- What makes objects "related"?
  - Definition of "similarity/distance"
- **Representation** for objects
  - Vector space? Normalization?
- **How many** clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid "trivial" clusters - too large or small
- Clustering **Algorithms**
  - Partitional algorithms
  - Hierarchical algorithms
- **Formal** foundation and convergence

12/4/14

69

# Clustering Algorithms

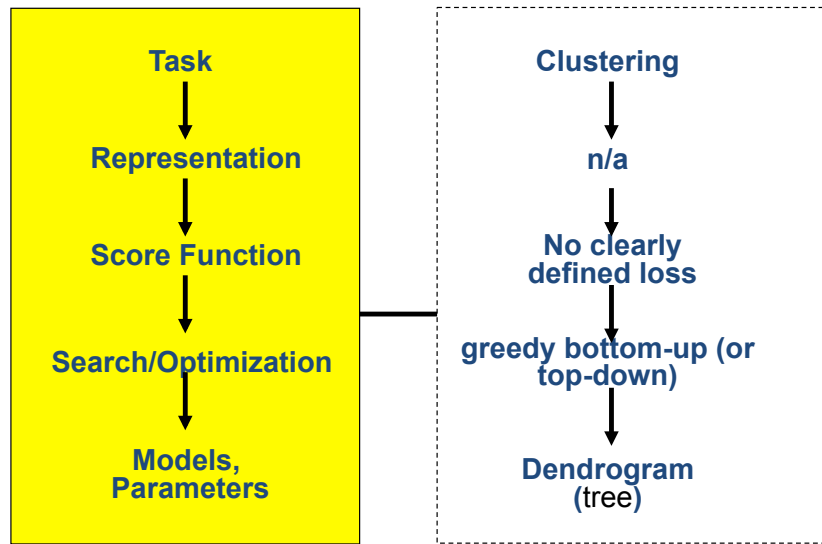
- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering
- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive



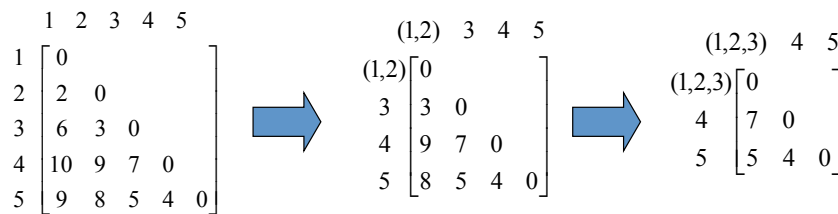
12/4/14

70

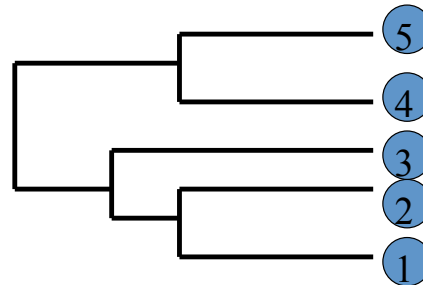
# (1) Hierarchical Clustering



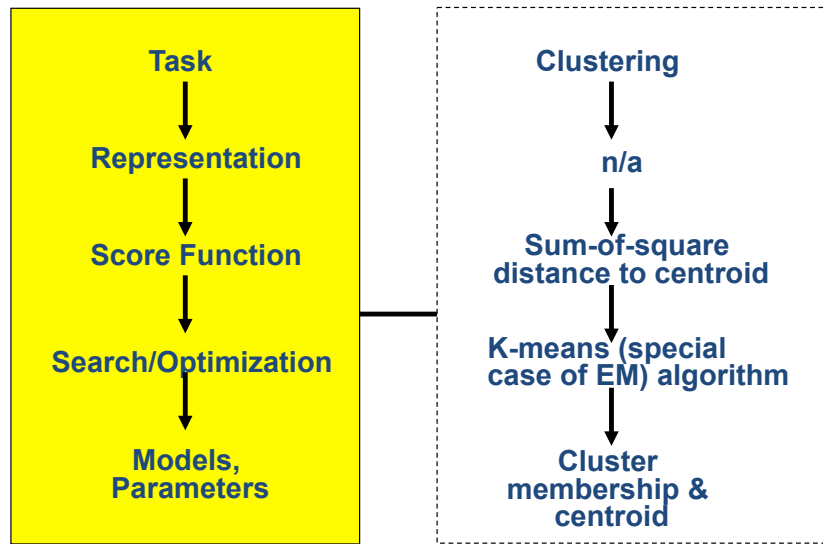
# Example: single link



$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$



## (2) K-means Clustering

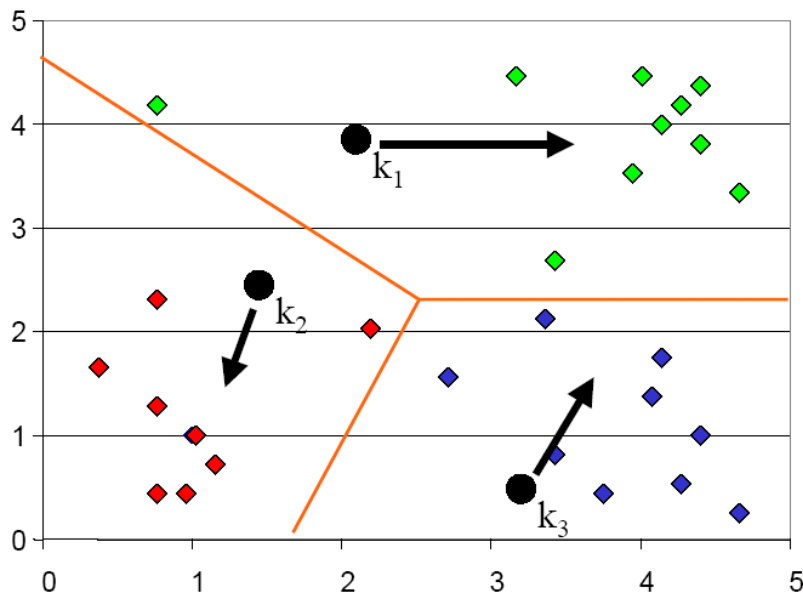


12/4/14

73

## K-means Clustering: Step 2

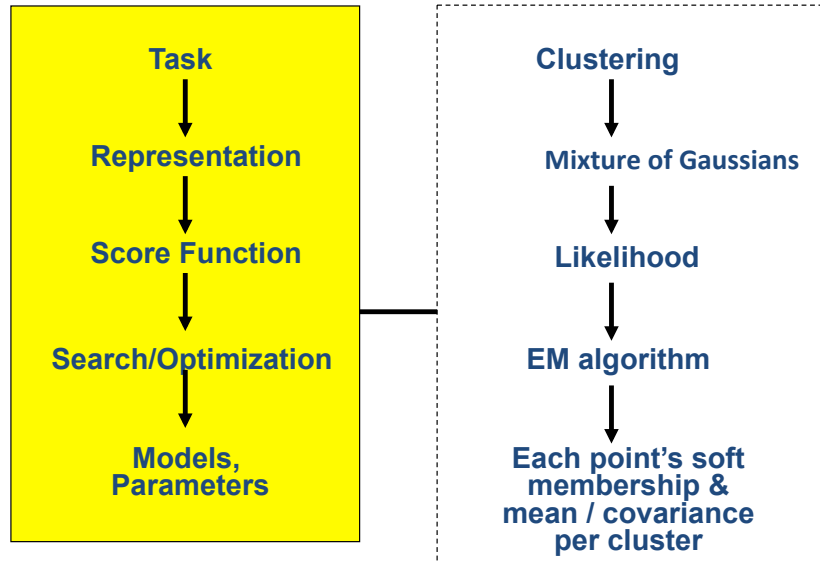
- Determine the membership of each data points



12/4/14

74

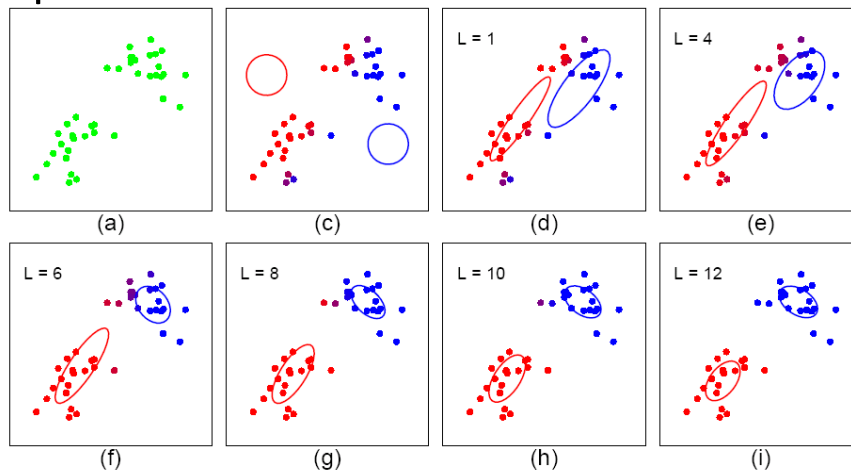
### (3) GMM Clustering



$$\begin{aligned}
 p(\vec{x} = x_i) &= \sum_{\mu_j} p(\vec{x} = x_i, \vec{\mu} = \mu_j) = \sum_{\mu_j} p(\vec{\mu} = \mu_j) p(\vec{x} = x_i | \vec{\mu} = \mu_j) \\
 &= \sum_{\mu_j} p(\vec{\mu} = \mu_j) \frac{1}{(2\pi)^{p/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x} - \mu_j) \Sigma_j^{-1} (\vec{x} - \mu_j)\right)
 \end{aligned}$$

## Expectation-Maximization for training GMM

- Start:
  - "Guess" the centroid  $\mu_k$  and covariance  $\Sigma_k$  of each of the K clusters
- Loop each cluster, revising both the mean (centroid position) and covariance (shape)



## What we have covered (IV)

### □ Learning theory / Model selection

- K-folds cross validation
- Expected prediction error
- Bias and variance tradeoff

12/4/14

77

## Evaluation Choice: e.g. 10 fold Cross Validation

- Divide data into 10 equal pieces
- 9 pieces as training set, the rest 1 as test set
- Collect the scores from the diagonal

model	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	train	train	train	train	train	train	train	train	train	test
2	train	train	train	train	train	train	train	train	test	train
3	train	train	train	train	train	train	train	test	train	train
4	train	train	train	train	train	train	test	train	train	train
5	train	train	train	train	train	test	train	train	train	train
6	train	train	train	train	test	train	train	train	train	train
7	train	train	train	test	train	train	train	train	train	train
8	train	train	test	train	train	train	train	train	train	train
9	train	test	train	train	train	train	train	train	train	train
10	test	train	train	train	train	train	train	train	train	train

12/4/14

78


# Expected prediction error (EPE)

Consider  
sample  
population  
distribution

$$\text{EPE}(f) = \mathbb{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$$

- For L2 loss: e.g.  $\int (y - f(x))^2 \Pr(dx, dy)$

under L2 loss, best estimator for EPE (Theoretically) is :

e.g. KNN  Conditional mean  $f(x) = \mathbb{E}(Y | X = x)$   
NN methods are the direct implementation (approximation)

- For 0-1 loss:  $L(k, \ell) = 1 - \delta_{kl}$   
 $\hat{G}(X) = C_k$  if  
 $\Pr(C_k | X = x) = \max_{g \in C} \Pr(g | X = x)$

 Bayes Classifier

12/4/14

79

# Bias-Variance Trade-off

$$\text{E}(\text{MSE}) = \text{noise}^2 + \text{bias}^2 + \text{variance}$$

Unavoidable  
error

Error due to  
incorrect  
assumptions

Error due to  
variance of training  
samples

See the ESL book for explanations of bias-variance ...

12/4/14

80  
Slide credit: D. Hoiem



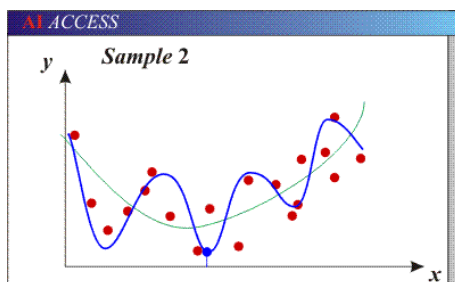
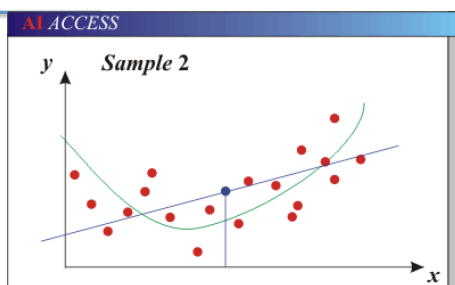
# need to make assumptions that are able to generalize

- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

12/4/14

81  
Slide credit: L. Lazebnik

## Bias-Variance Trade-off

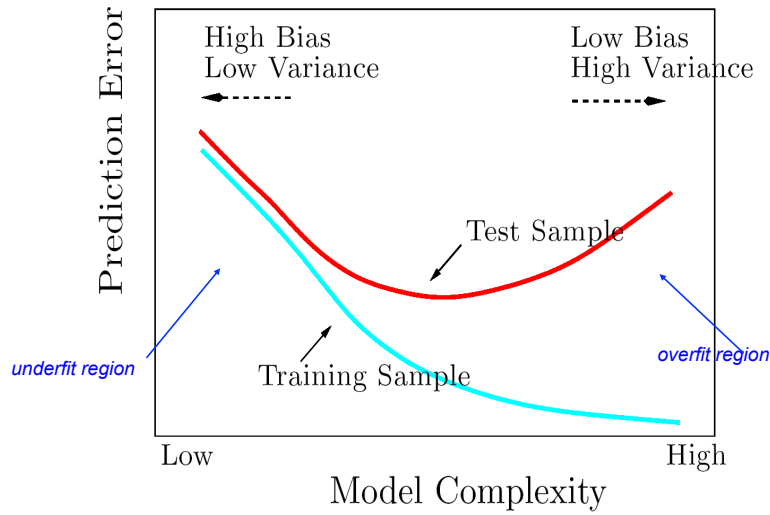


- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample randomness).

12/4/14

82  
Slide credit: D. Hoiem

# Bias-Variance Tradeoff / Model Selection

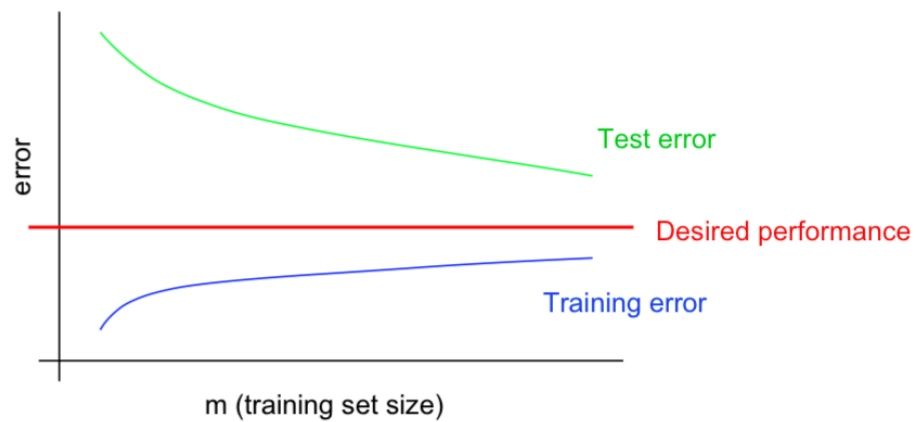


12/4/14

83

## High variance

Typical learning curve for high variance:



- Test error still decreasing as  $m$  increases. Suggests larger training set will help.
- Large gap between training and test error.
- **Low training error and high test error**

12/4/14

34  
Slide credit: A. Ng

# How to reduce variance?

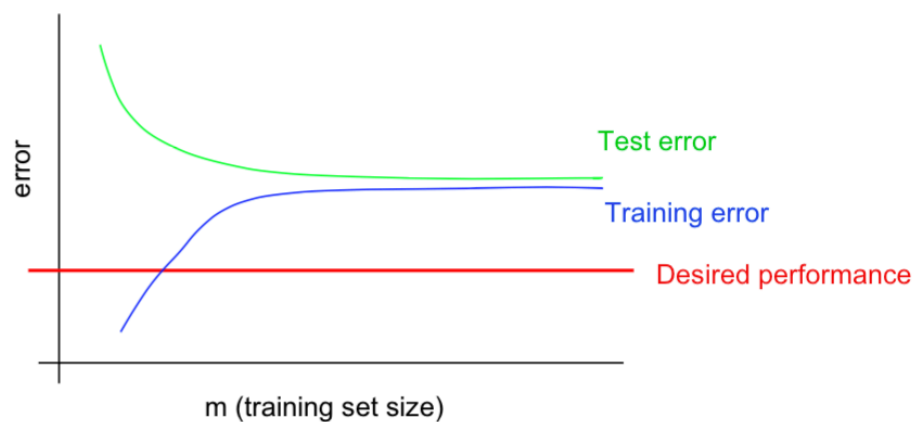
- Choose a simpler classifier
- Regularize the parameters
- Get more training data
- Try smaller set of features

12/4/14

85  
Slide credit: D. Hoiem

# High bias

Typical learning curve for high bias:



- Even training error is unacceptably high.
- Small gap between training and test error.

**High training error and high test error**

12/4/14

86  
Slide credit: A. Ng

## How to reduce Bias ?

- E.g.
  - Get additional features
  - Try adding basis expansions, e.g. polynomial
  - Try more complex learner

## For instance, if trying to solve “spam detection” using

L2 - logistic regression, implemented with gradient descent.

Fixes to try: **If performance is not as desired**

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>- Try getting more training examples.</li> <li>- Try a smaller set of features.</li> <li>- Try a larger set of features.</li> <li>- Try email header features.</li> <li>- Run gradient descent for more iterations.</li> <li>- Try Newton’s method.</li> <li>- Use a different value for <math>\lambda</math>.</li> <li>- Try using an SVM.</li> </ul> | <ul style="list-style-type: none"> <li>Fixes high variance.</li> <li>Fixes high variance.</li> <li>Fixes high bias.</li> <li>Fixes high bias.</li> <li>Fixes optimization algorithm.</li> <li>Fixes optimization algorithm.</li> <li>Fixes optimization objective.</li> <li>Fixes optimization objective.</li> </ul> |
|---|--|

## References

- ❑ Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.
- ❑ Prof. M.A. Papalaskar's slides
- ❑ Prof. Andrew Ng's slides