

# UVA CS 6316: Machine Learning

## Lecture 22: Review

Dr. Yanjun Qi

University of Virginia  
Department of Computer Science

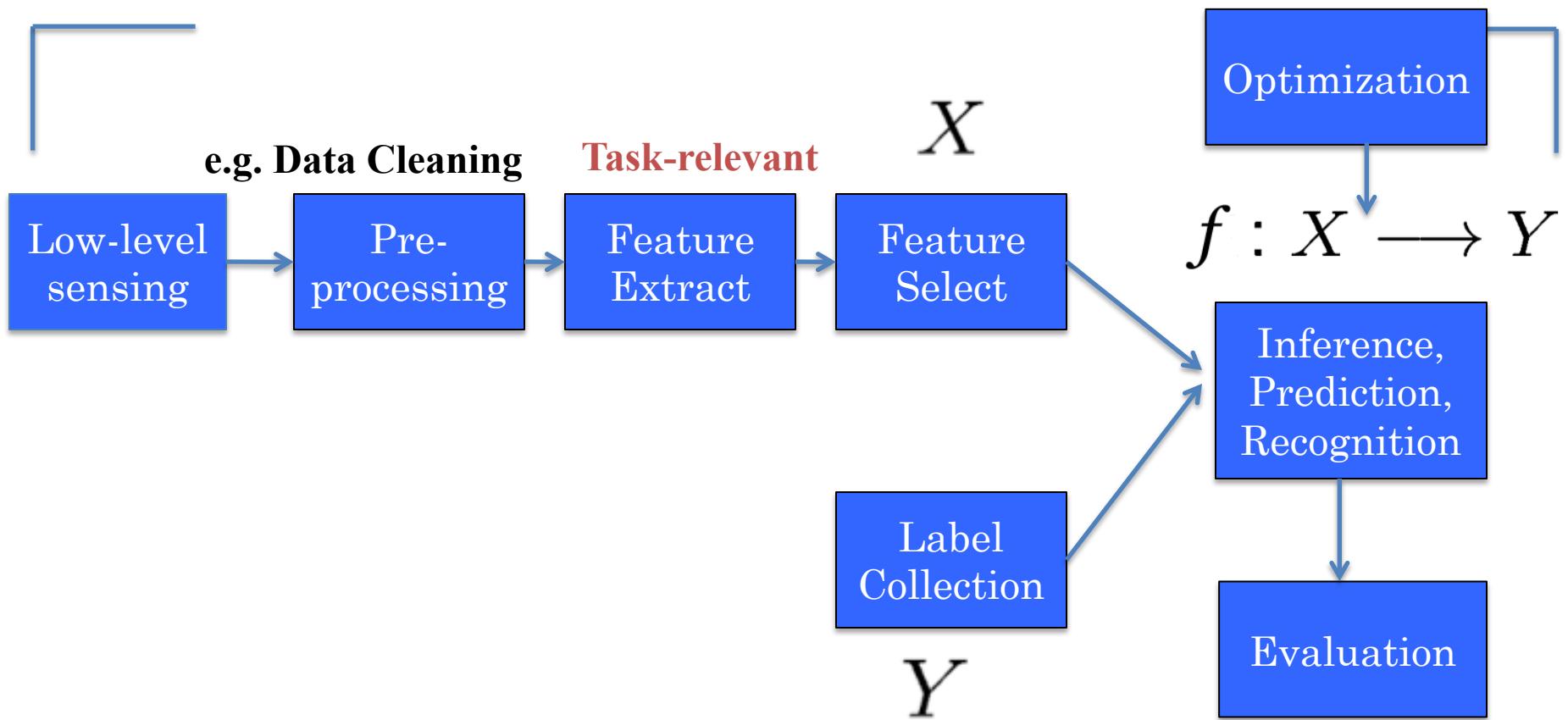
# Objective

- To help students be able to build machine learning tools
  - (not just a tool user!!!)
- Key Results:
  - Able to build a few simple machine learning methods from scratch
  - Able to understand a few complex machine learning methods at the source code level

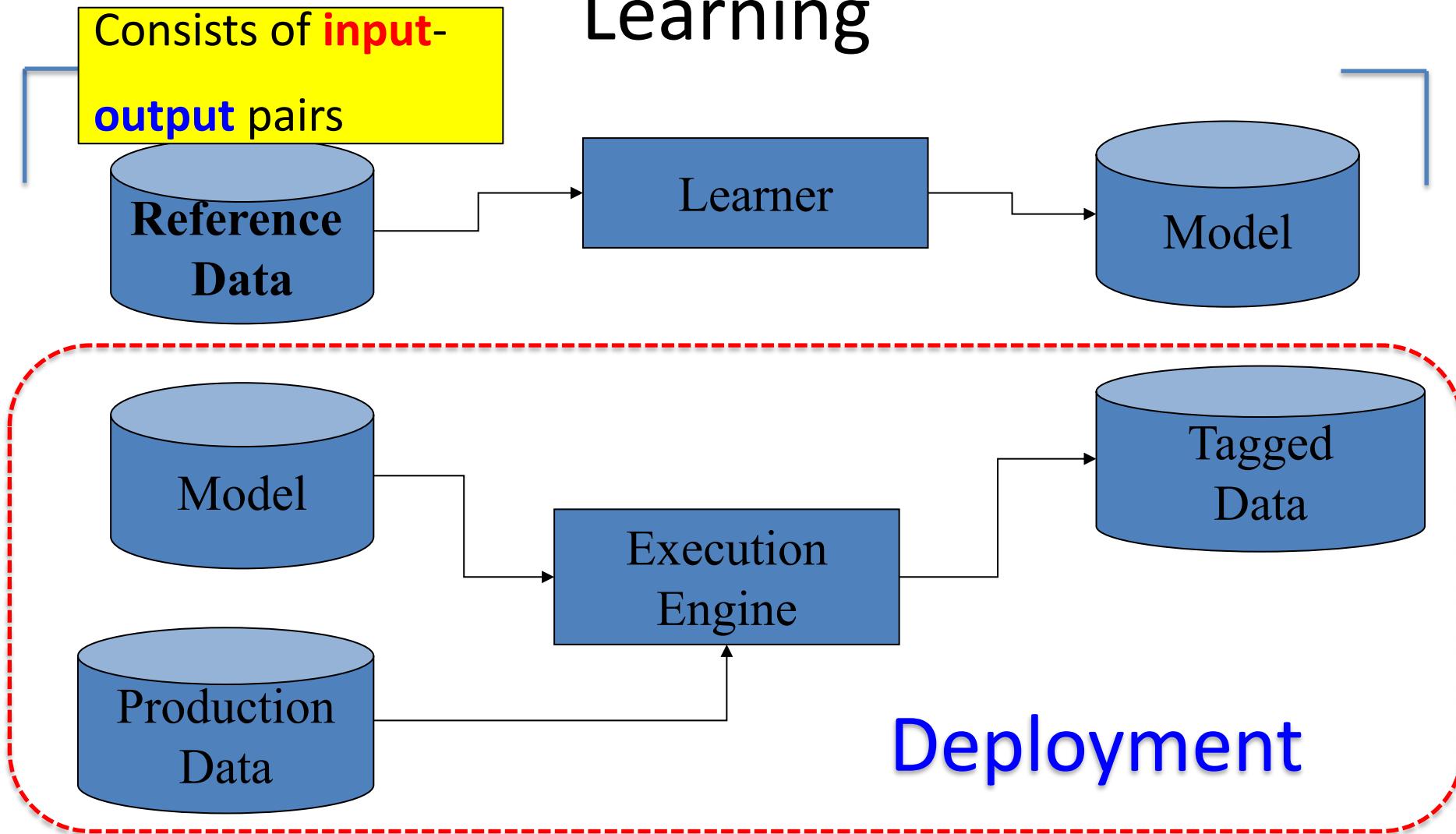
# Recap

- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised)
  - ❑ Unsupervised models
  - ❑ Learning theory
  
- ❑ Review of Assignments covered so far

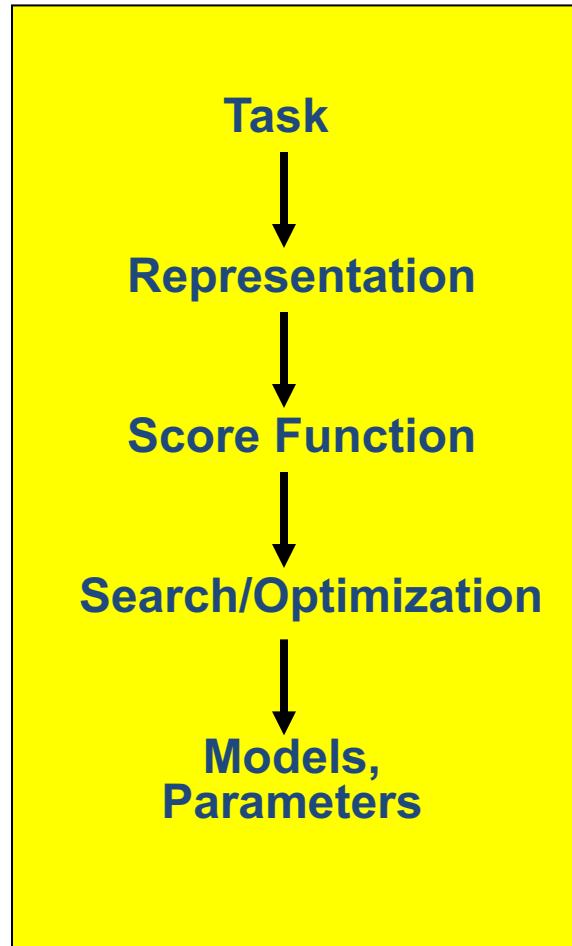
# A Typical Machine Learning Pipeline



# An Operational Model of Machine Learning



# Machine Learning in a Nutshell



ML grew out of work in AI

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

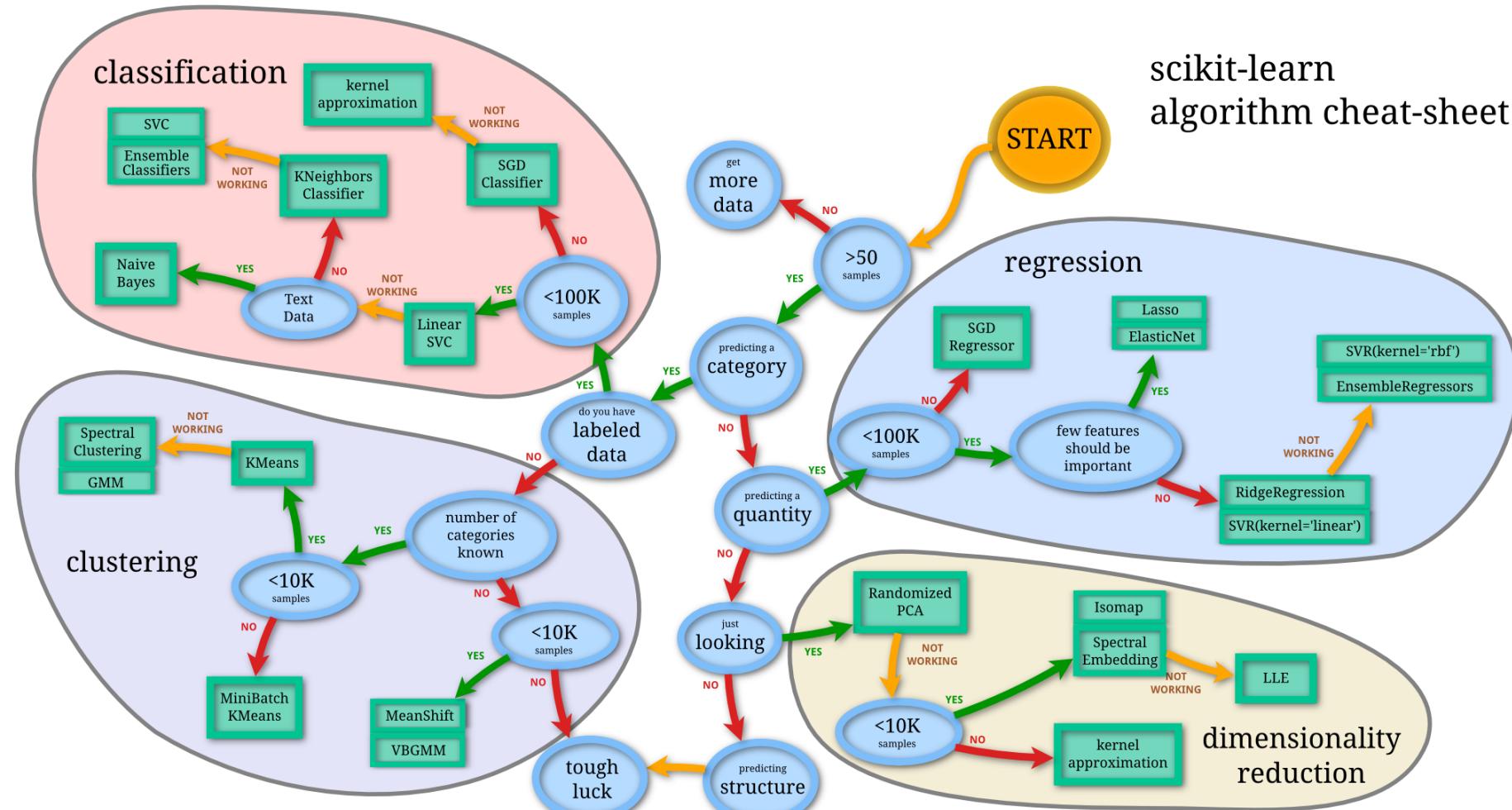
*Aiming to generalize to unseen data*

# What we have covered

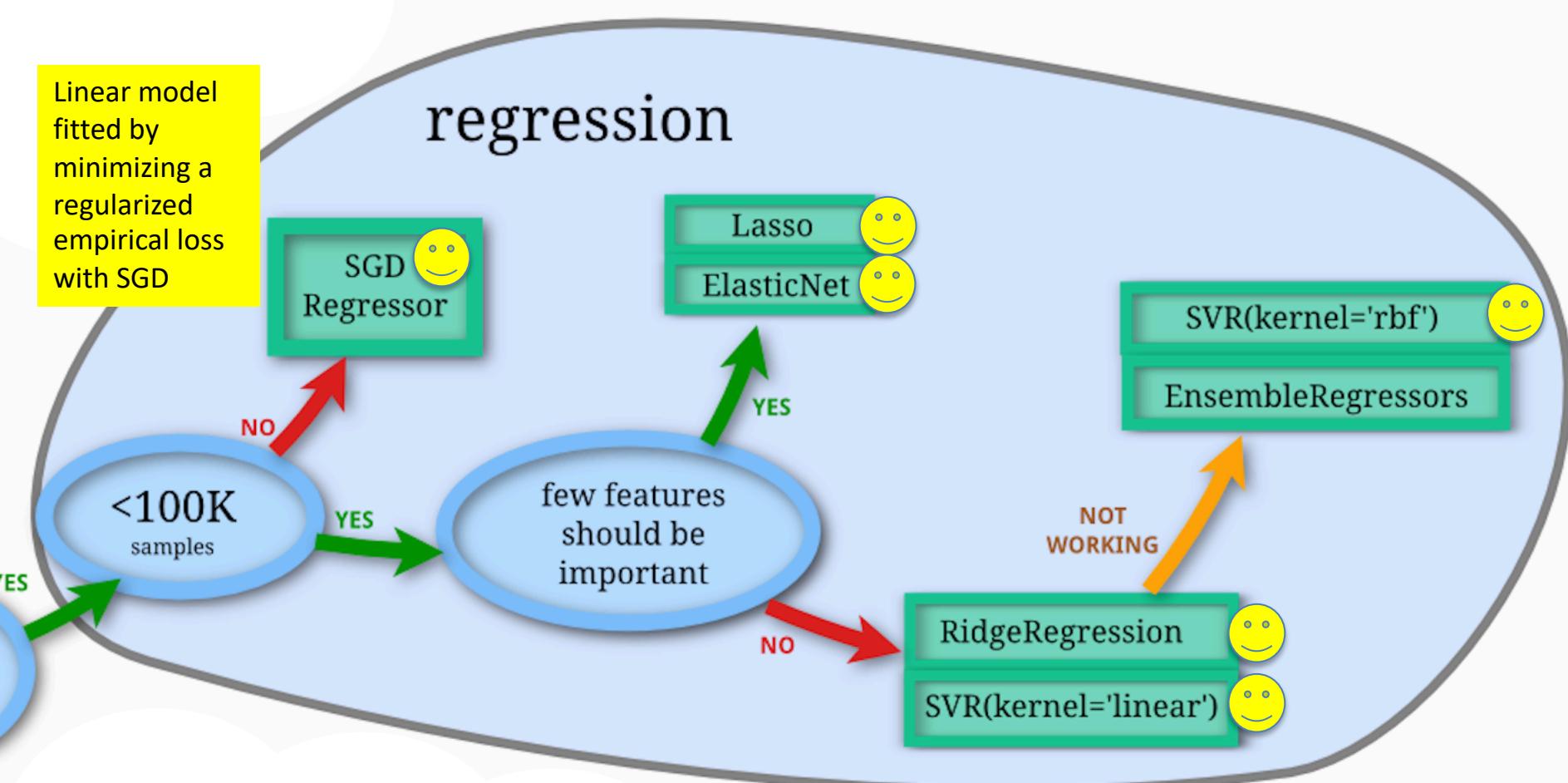
Task	
Representation	
Score Function	
Search/Optimization	
Models, Parameters	

[http://scikit-learn.org/stable/tutorial/machine learning map/](http://scikit-learn.org/stable/tutorial/machine_learning_map/)

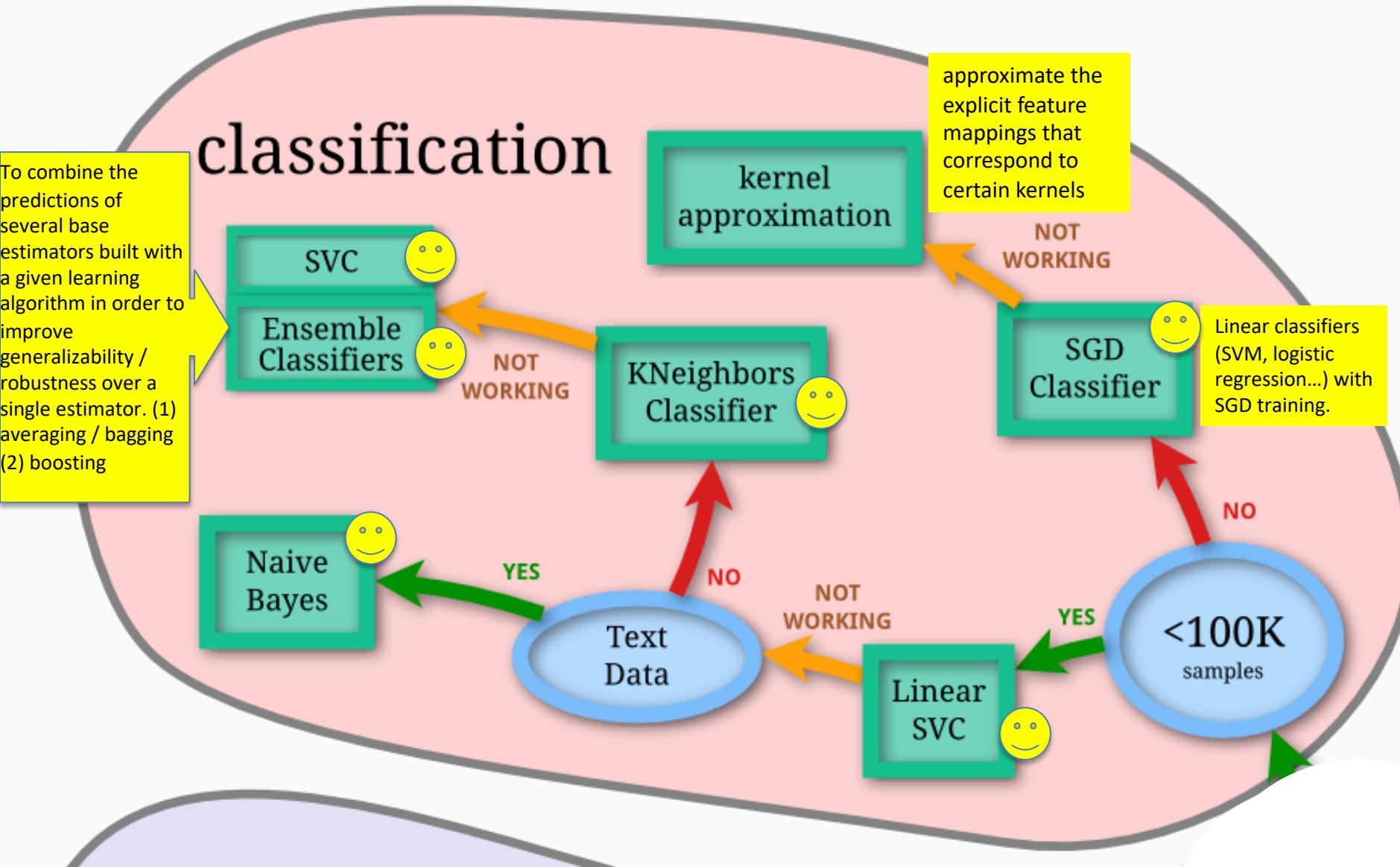
# Scikit-learn algorithm cheat-sheet



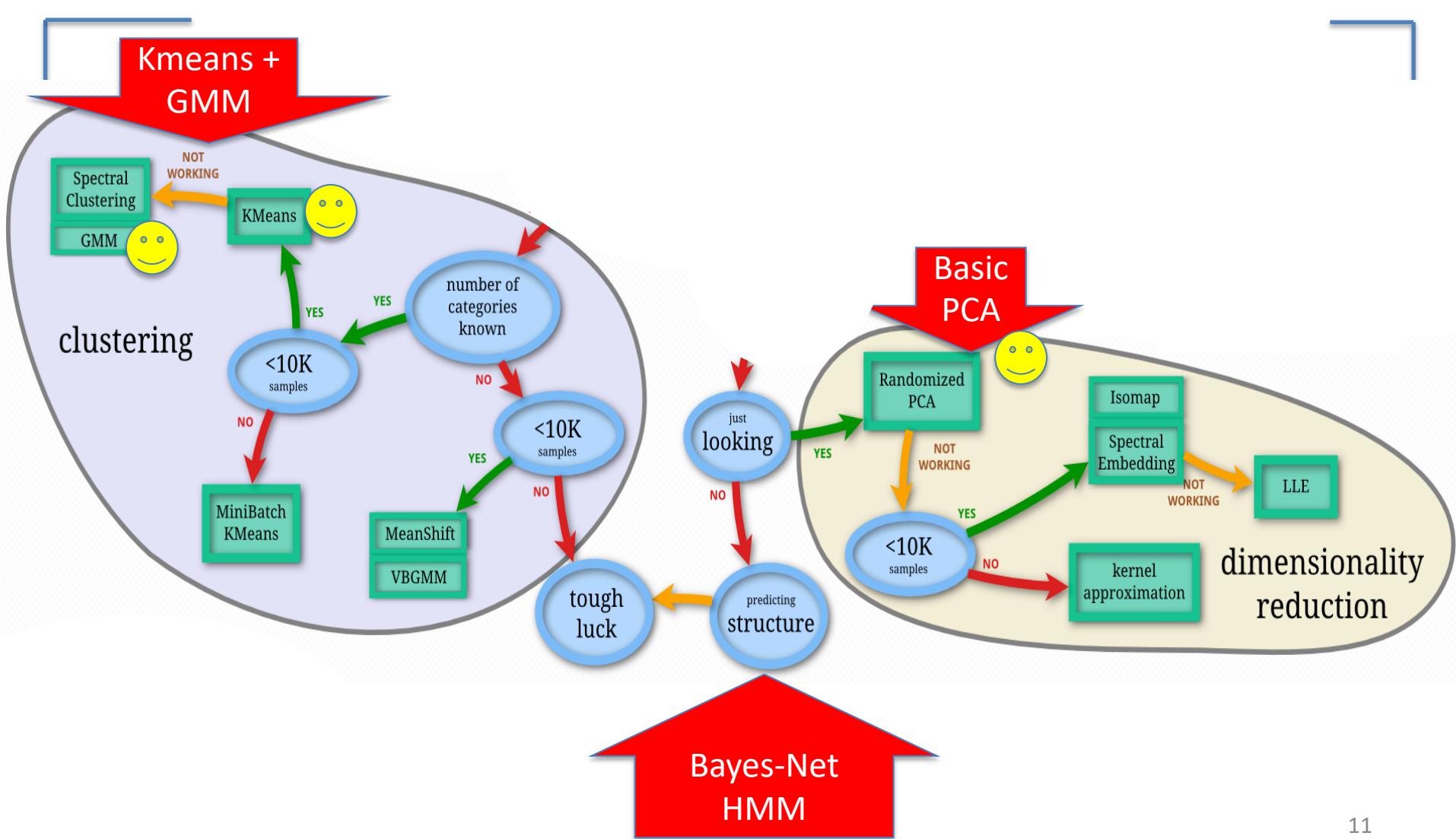
# Scikit-learn : Regression



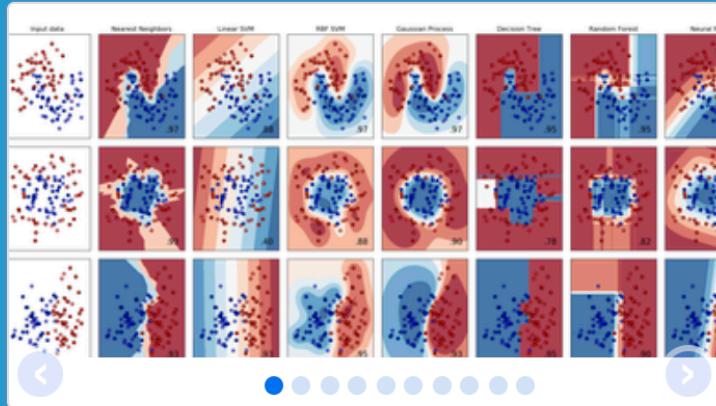
# Scikit-learn : Classification



# Then after classification



<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 category an object belongs to.

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

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

— Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**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

# Recap

- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised)
  - ❑ Unsupervised models
  - ❑ Learning theory
- ❑ Review of Assignments covered so far

# SUPERVISED LEARNING

$$f : X \longrightarrow Y$$

- Find function to map **input** space  $X$  to **output** space  $Y$
- **Generalisation**: learn function / hypothesis from **past data** in order to “explain”, “predict”, “model” or “control” **new** data examples

# What we have covered (I)

## ❑ Supervised Regression models

- Linear regression (LR)
- LR with non-linear basis functions
- ~~Locally weighted LR~~
- LR with Regularizations
- Feature selection \*

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

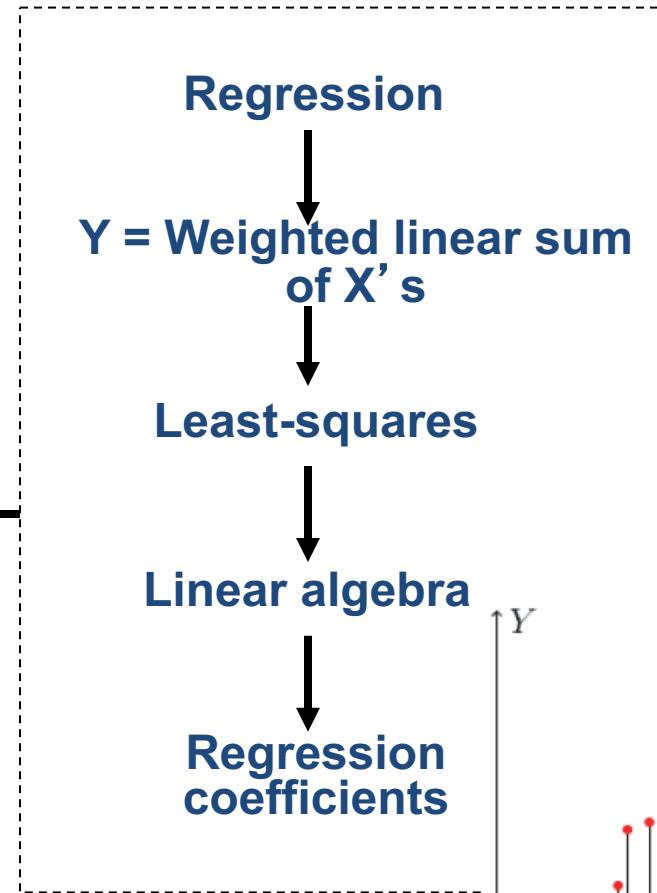
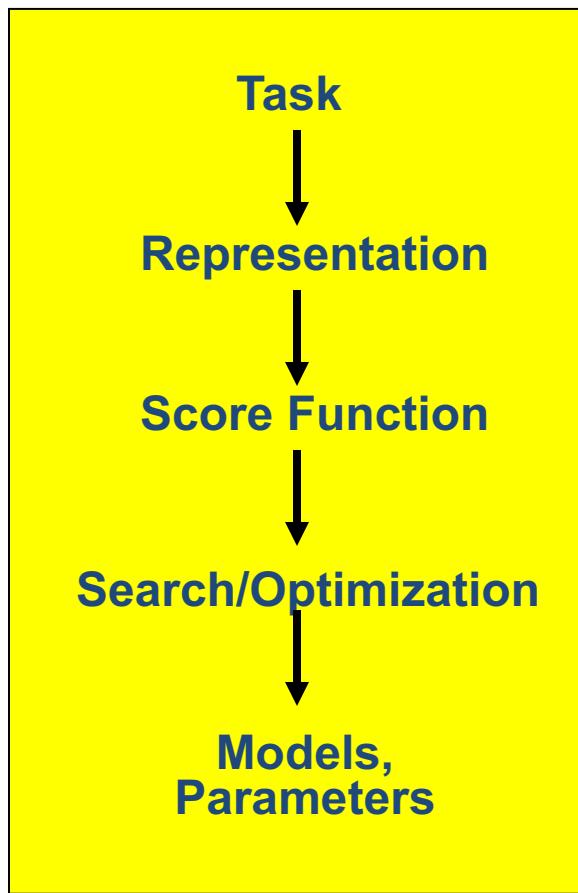
# A Dataset

$$f : \boxed{X} \longrightarrow \boxed{Y}$$

Output Y as  
continuous values

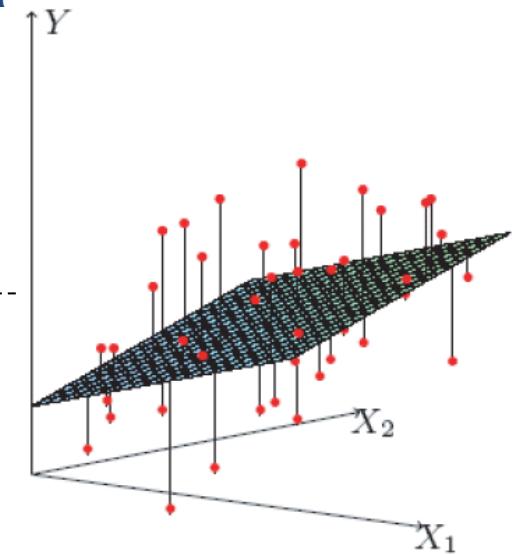
- **Data/points/instances/examples/samples/records:** [ rows ]
- **Features/attributes/dimensions/independent variables/covariates/predictors/regressors:** [ columns, except the last ]
- **Target/outcome/response/label/dependent variable:** special column to be predicted [ last column ]

# (1) Multivariate Linear Regression

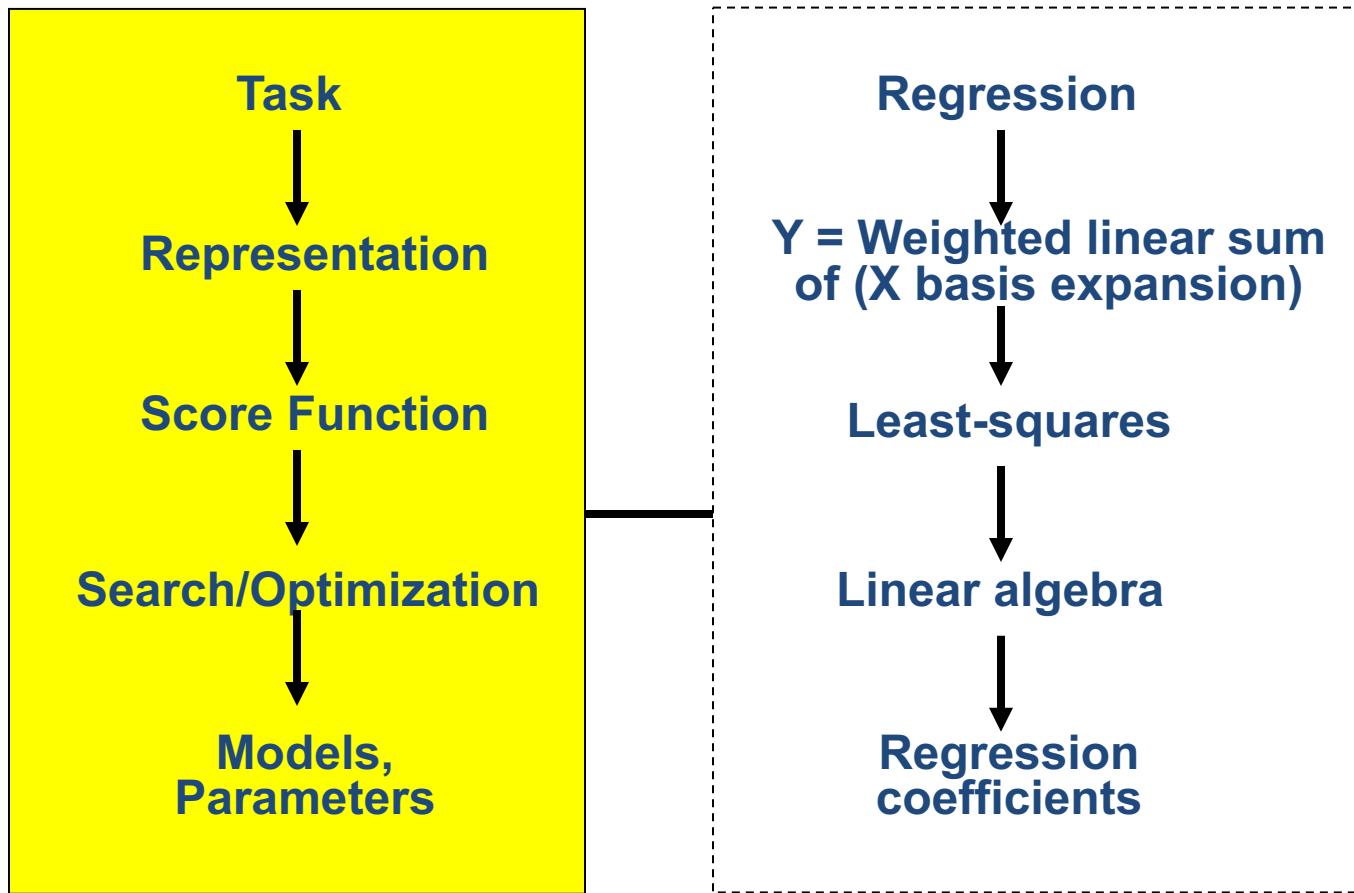


$$\hat{y} = f(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2$$

$$\theta$$

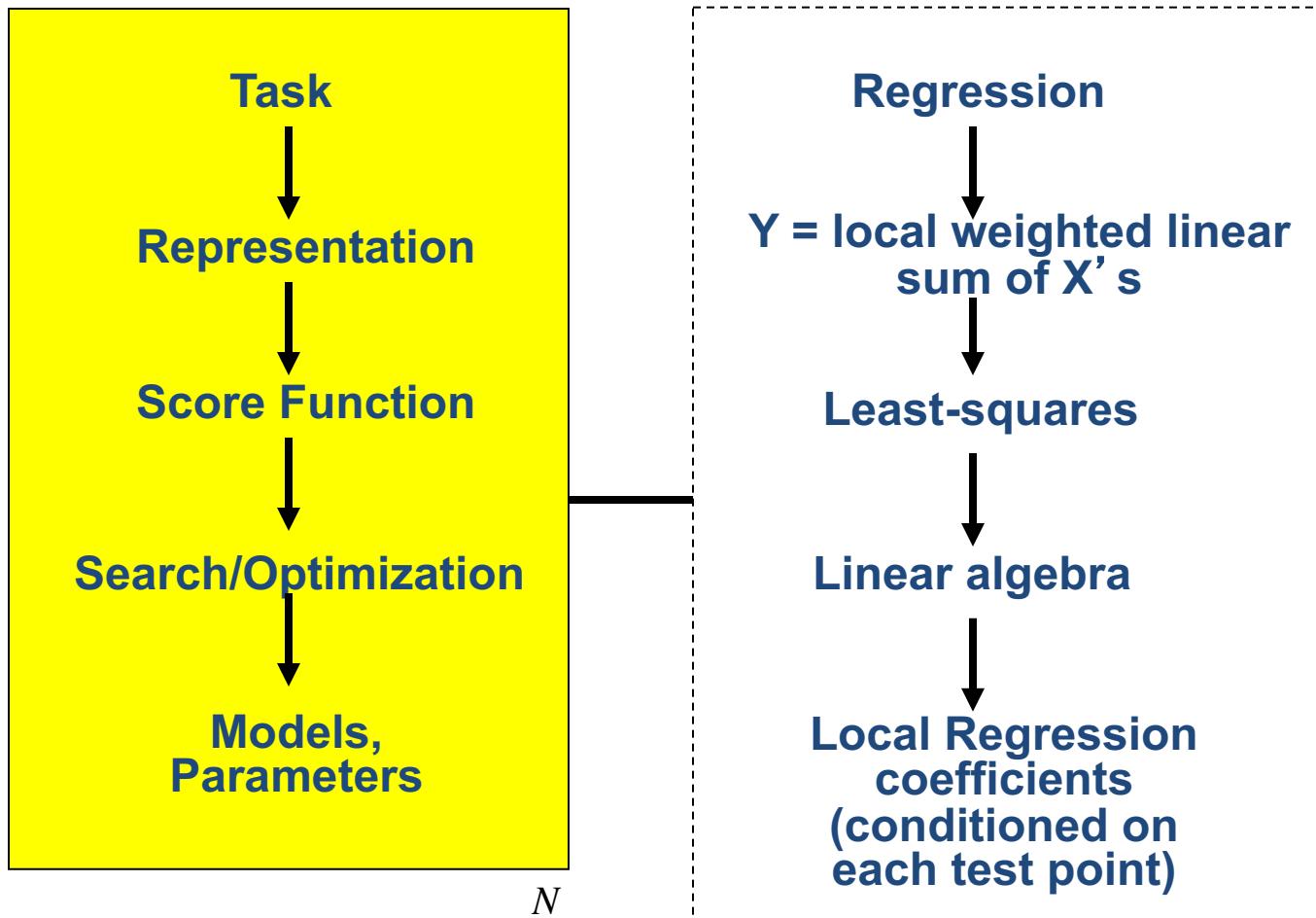


## (2) Multivariate Linear Regression with basis Expansion



$$\hat{y} = \theta_0 + \sum_{j=1}^m \theta_j \varphi_j(x) = \varphi(x)\theta$$

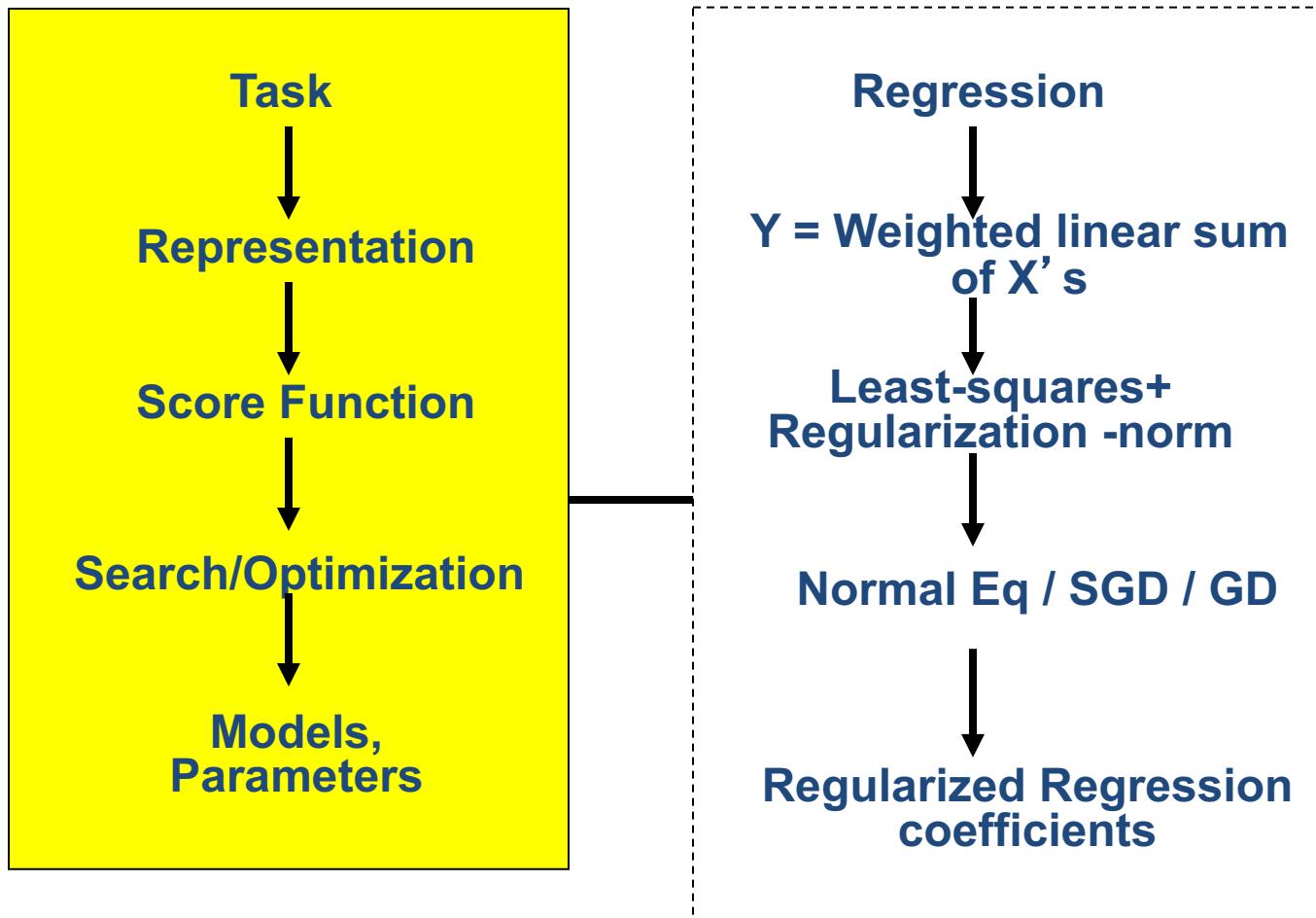
### (3) Locally Weighted / Kernel Regression



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

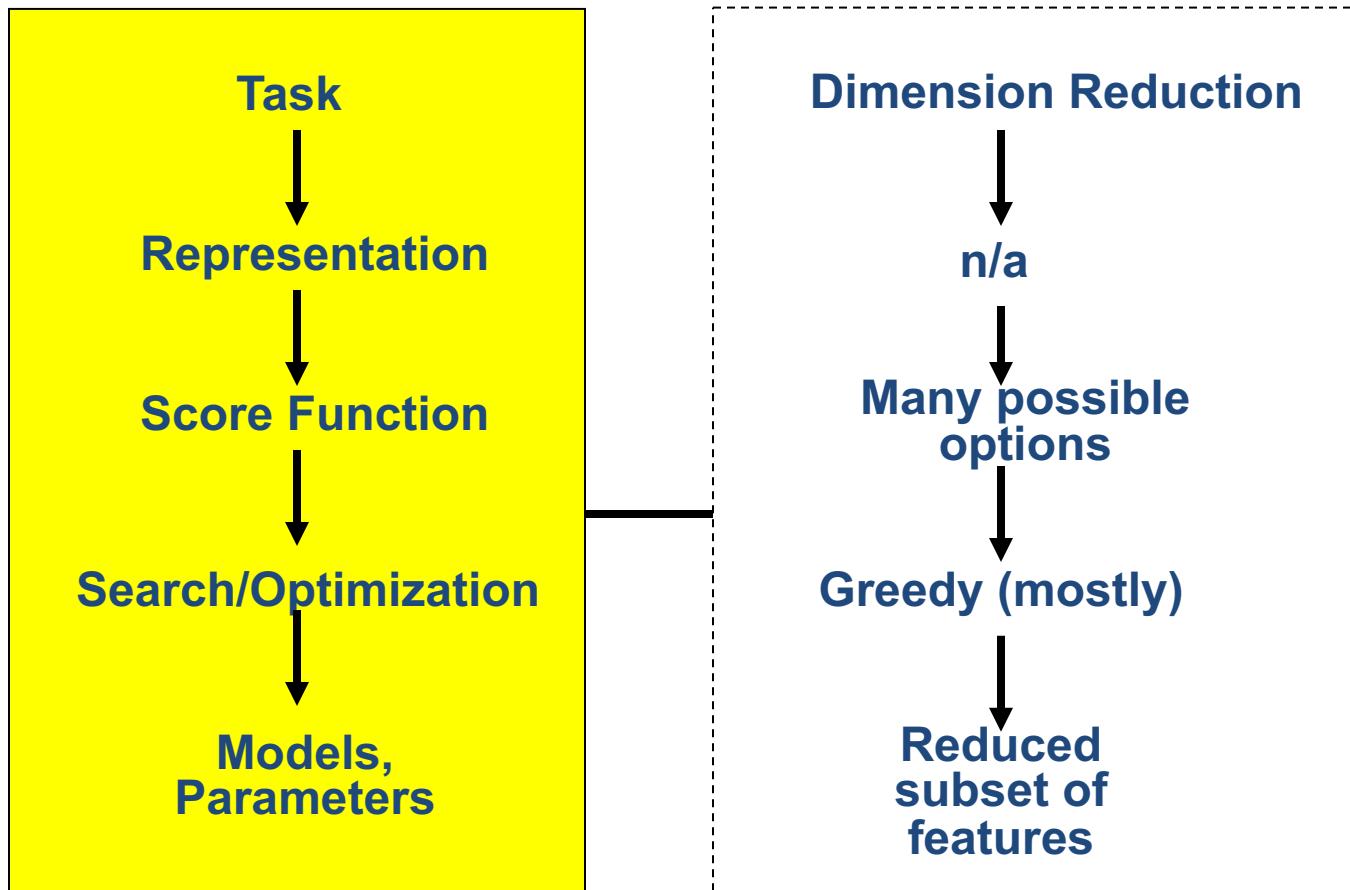
$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$$

## (4) Regularized multivariate linear regression



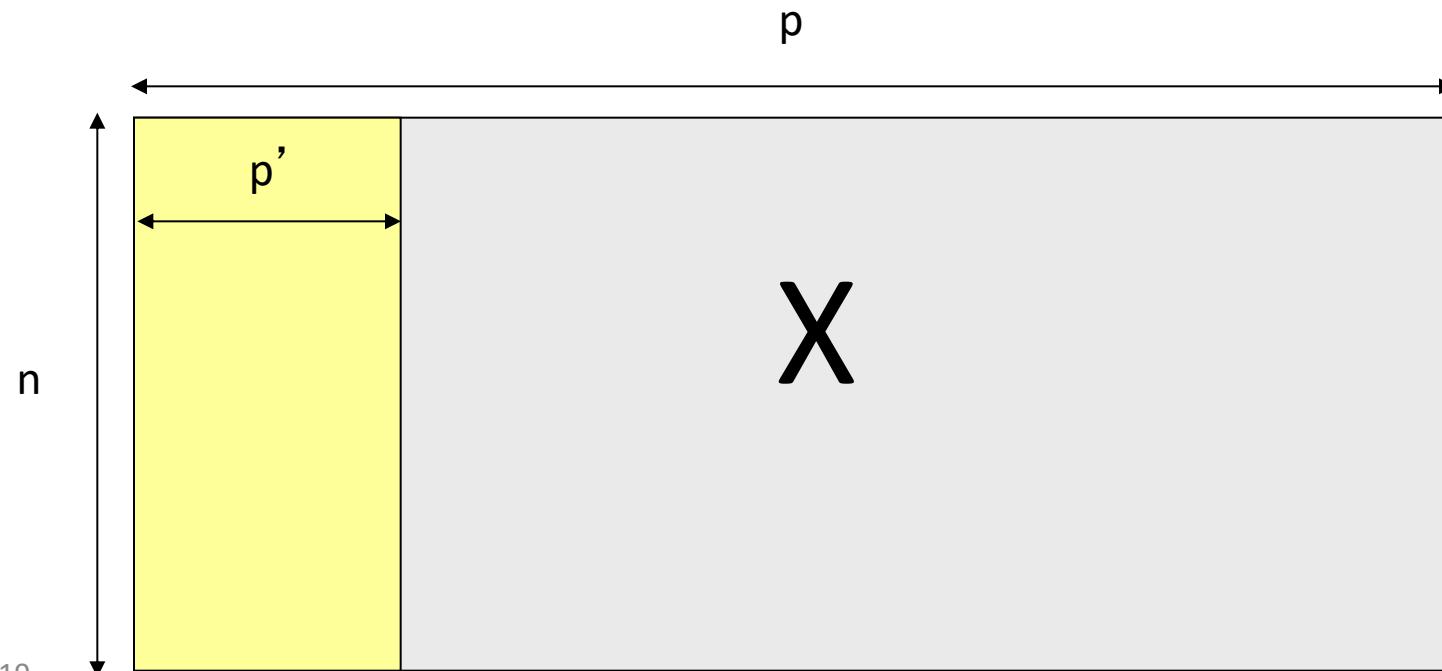
$$\min J(\beta) = \sum_{i=1}^n (Y - \hat{Y})^2 + \lambda \sum_{j=1}^p \beta_j^2$$

## (5) Feature Selection



# (5) Feature Selection

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



# Recap

- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised) 
  - ❑ Unsupervised models
  - ❑ Learning theory
- ❑ Review of Assignments covered so far

# What we have covered (II)

- Supervised Classification models
  - K-nearest Neighbor
  - Support Vector Machine
  - Logistic Regression
  - Neural Network (e.g. MLP, CNN)
  - Generative Bayes Classifier (Naïve, Gaussian LDA, QDA)
  - Random forest / Decision Tree / Boosting

# 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., **logistic regression**, support vector machine, decision Trees
- 2. Generative:
  - build a generative statistical model
  - e.g., **naïve bayes classifier**, **Bayesian networks**
- 3. Instance based classifiers
  - Use observation directly (no models)
  - e.g. **K nearest neighbors**

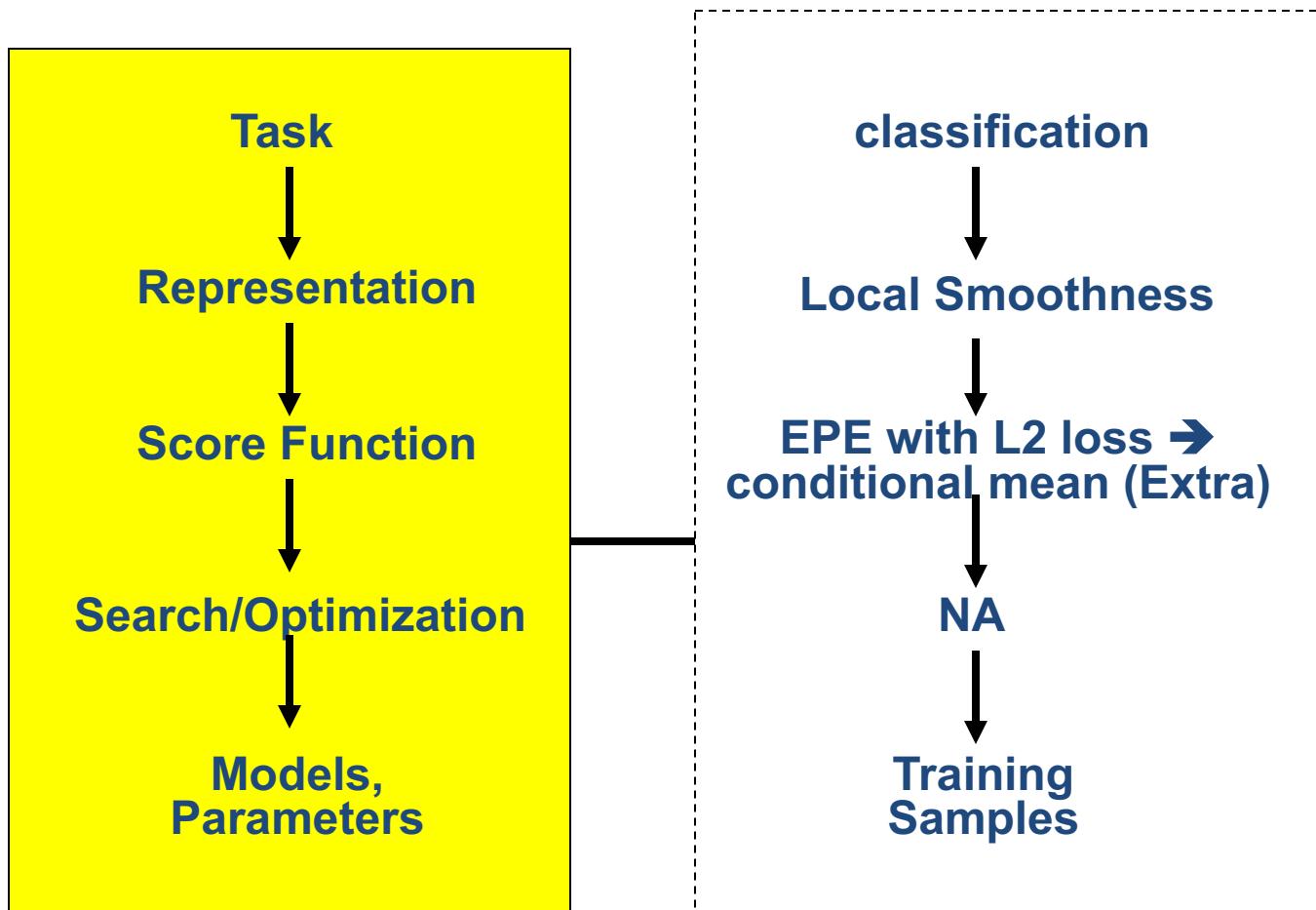
# A Dataset for classification

$$f : [X] \longrightarrow [C]$$

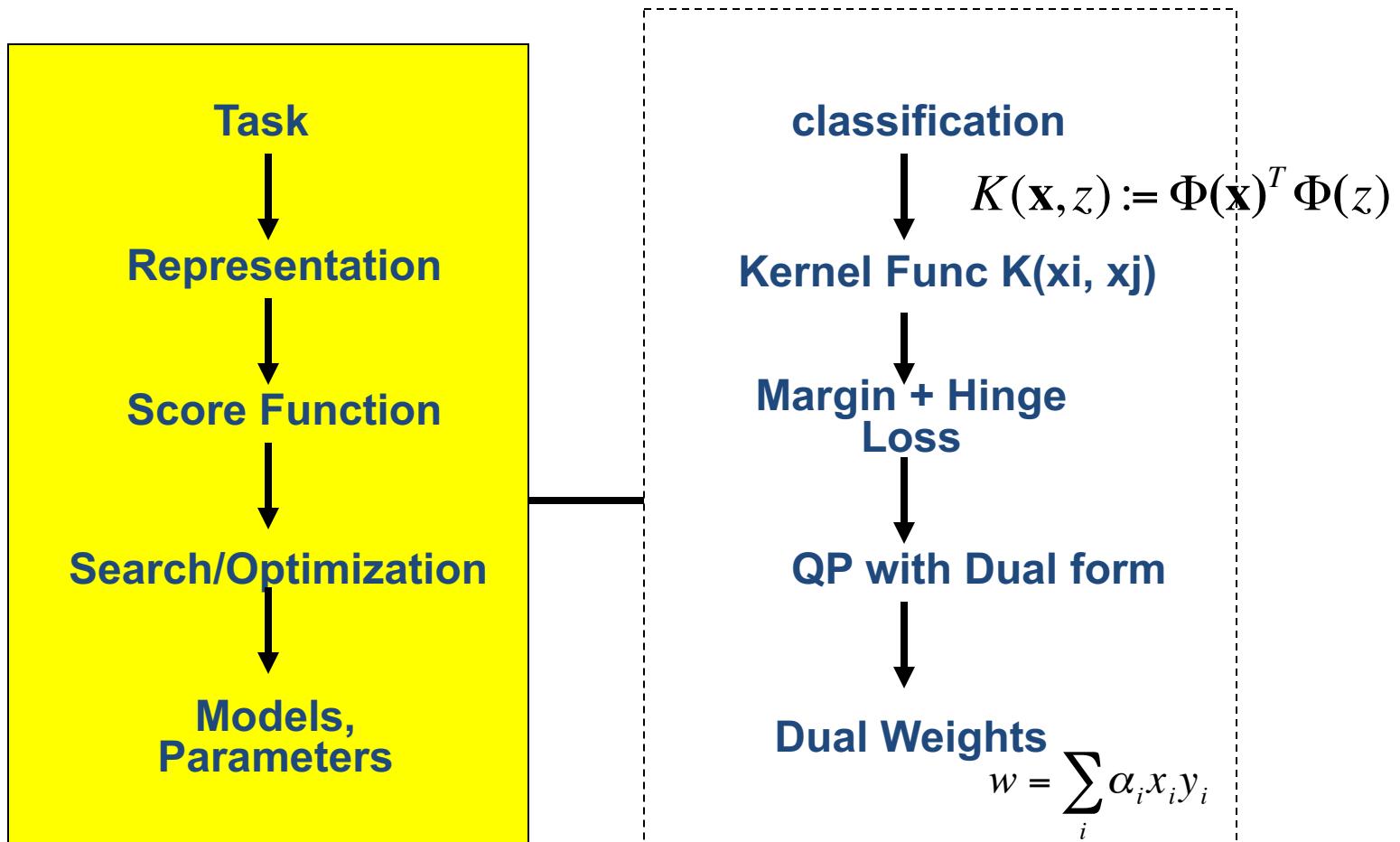
Output as Discrete  
Class Label  
 $C_1, C_2, \dots, C_L$

- 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 ]

# (1) K-Nearest Neighbor



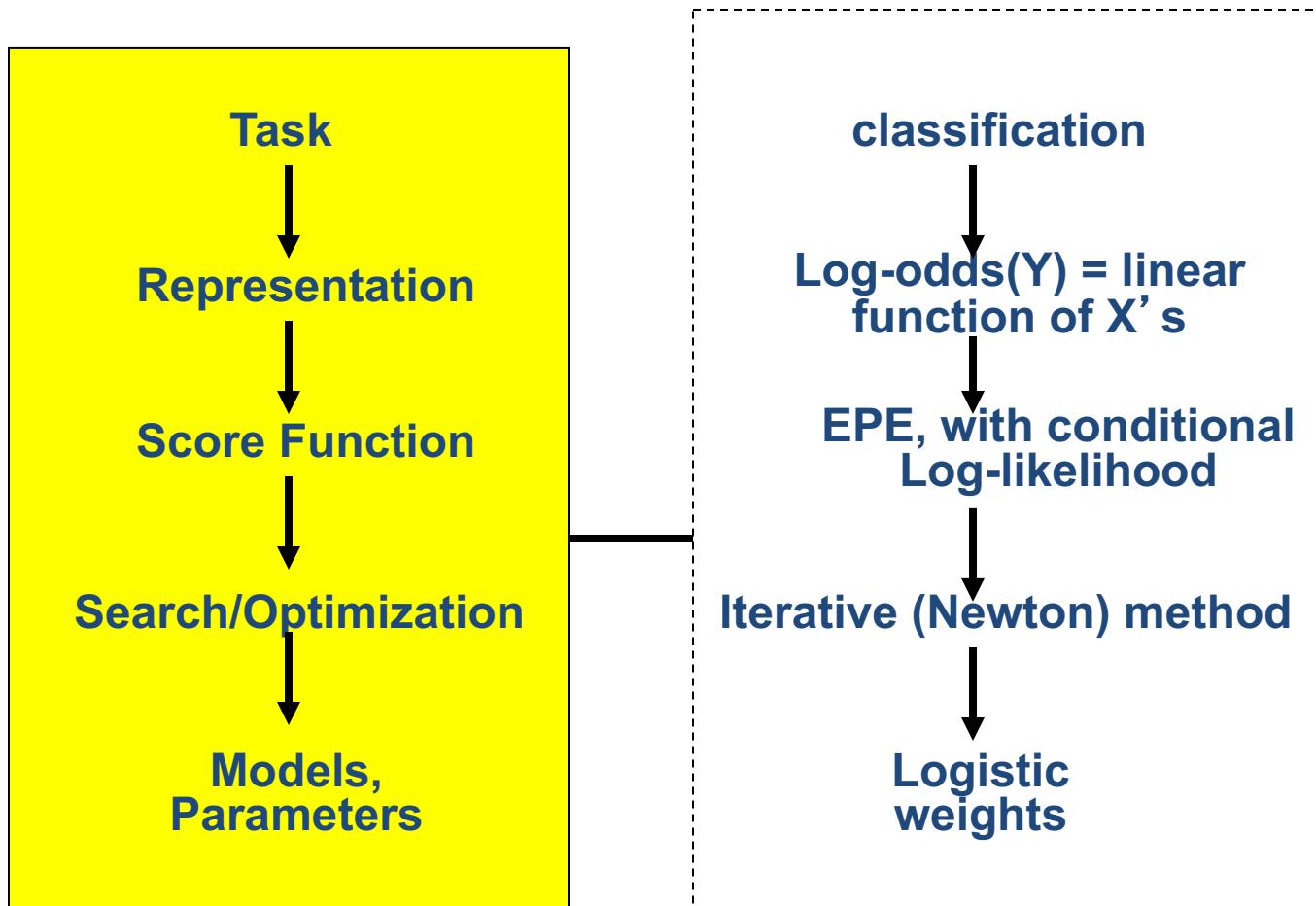
## (2) Support Vector Machine



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

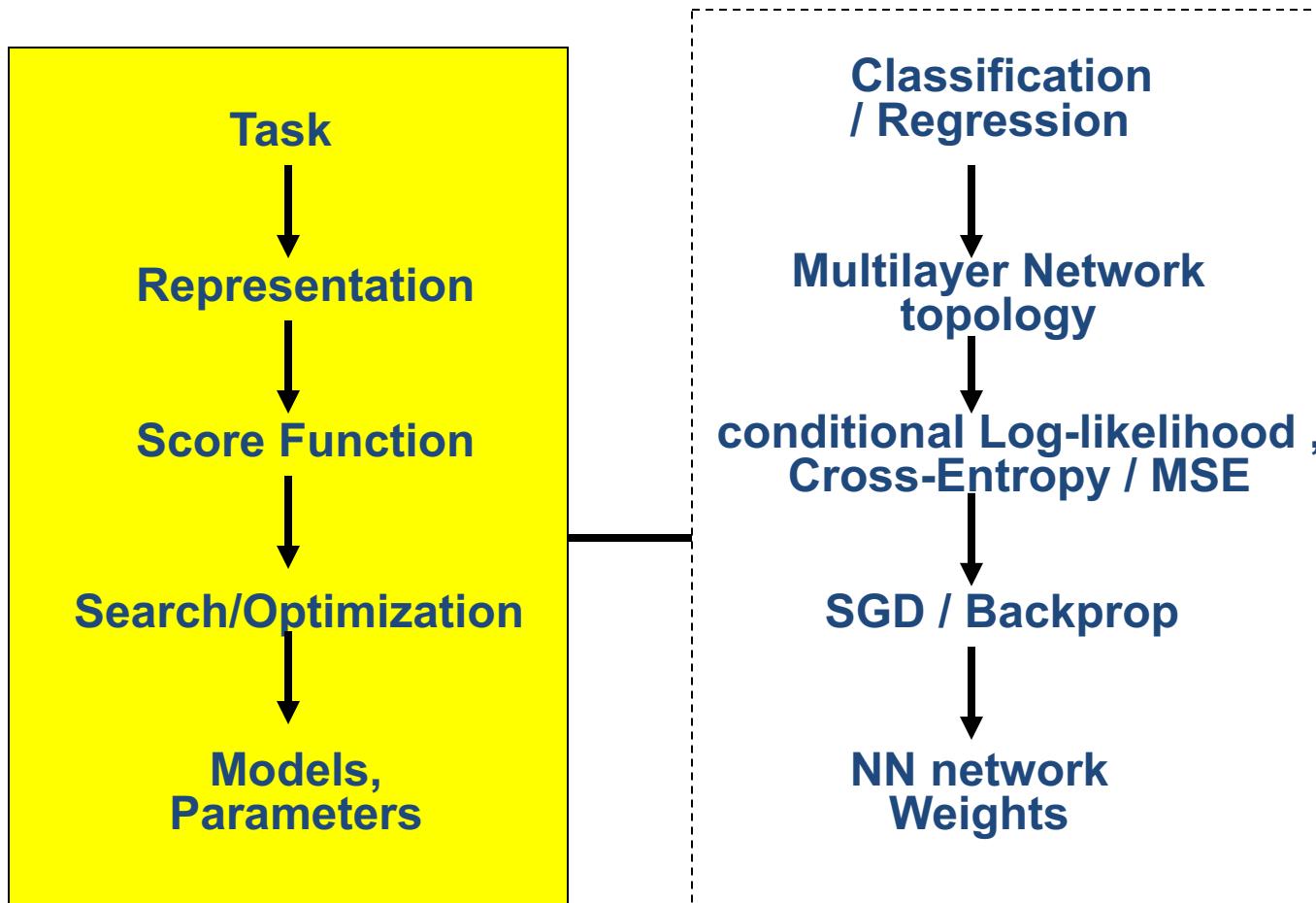
subject to  $\forall \mathbf{x}_i \in D_{train} : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \varepsilon_i$  28

### (3) Logistic Regression



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

## (4) Neural Network

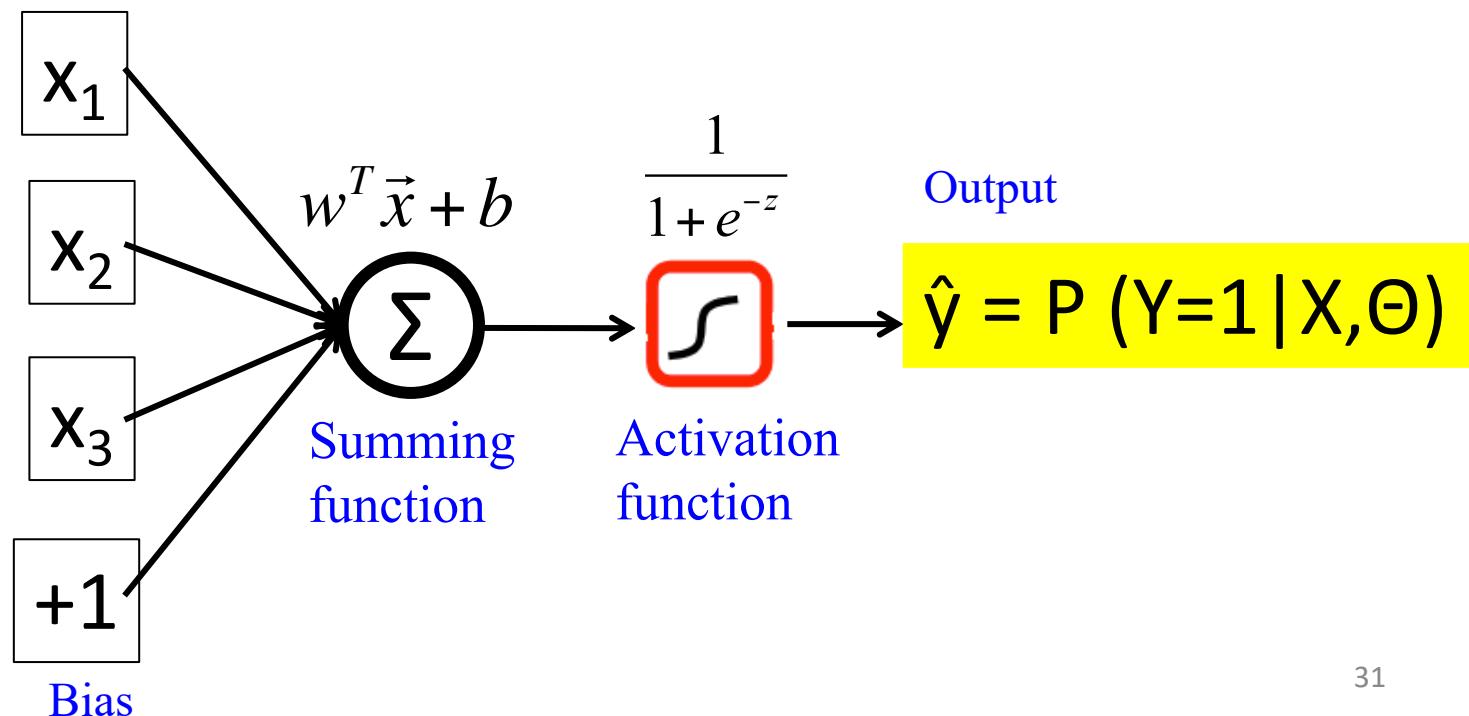


# Logistic regression

Logistic regression could be illustrated as a module

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

where

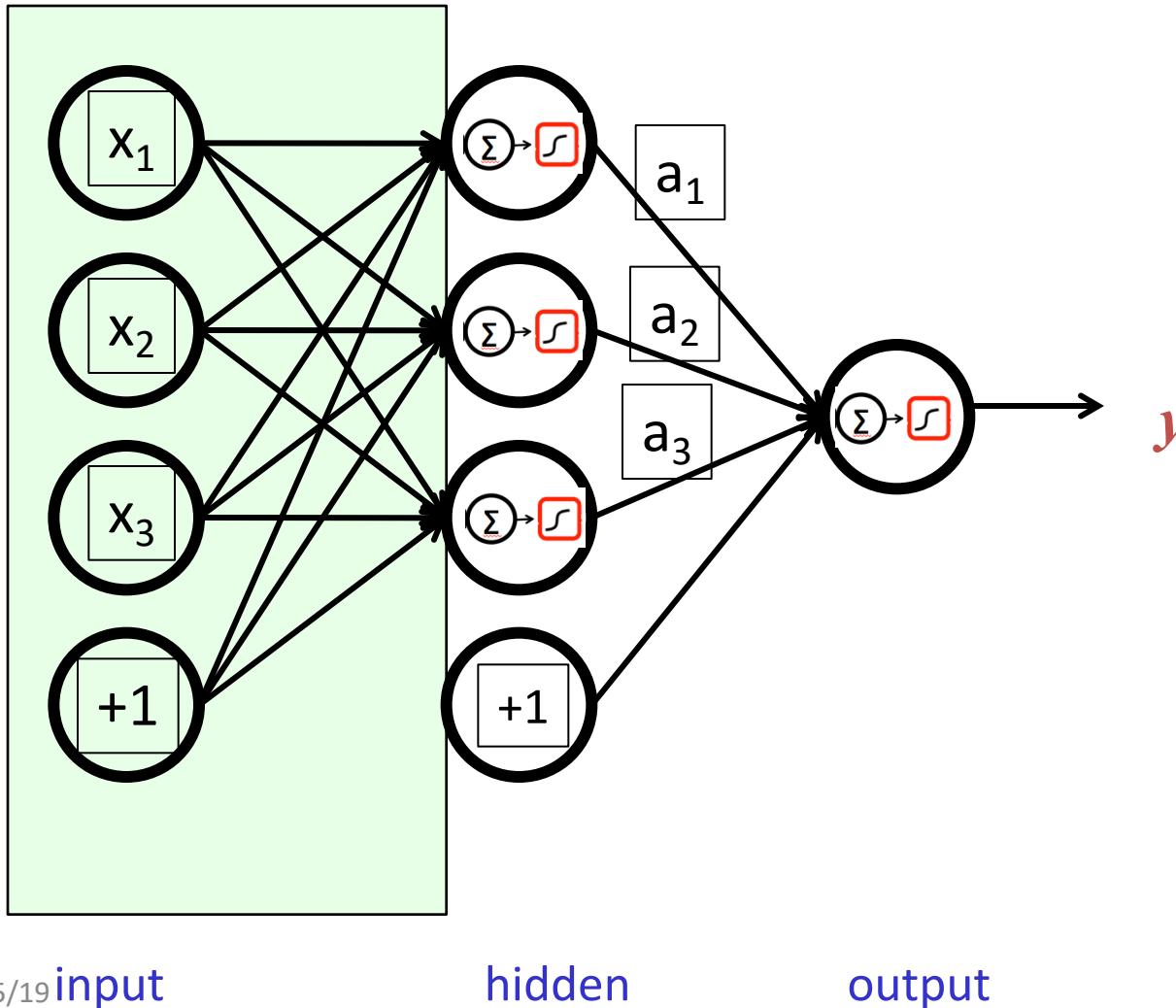


Draw a logistic regression unit as:

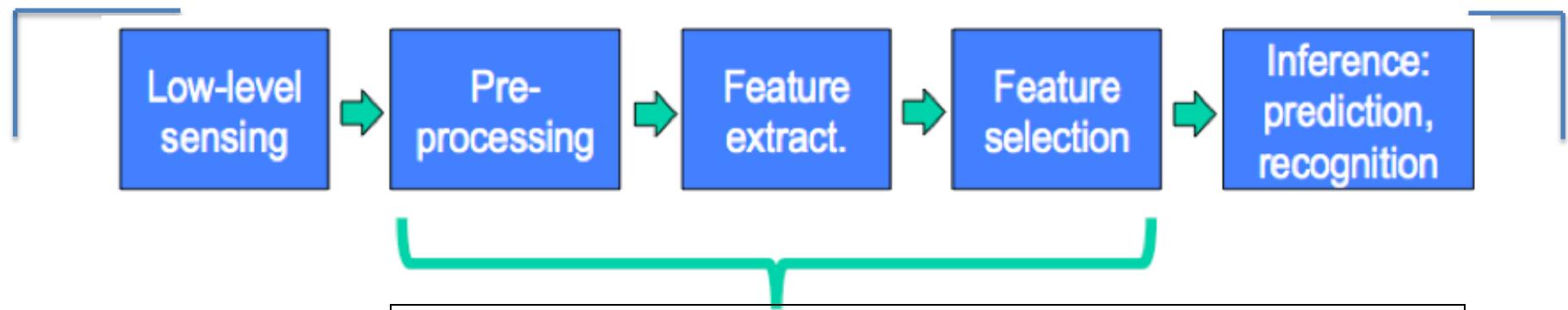
11/25/19

# Multi-Layer Perceptron (MLP)

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

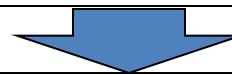


# Deep Learning Way: Learning features / Representation from data



## Feature Engineering

- ✓ Most critical for accuracy
- ✓ Account for most of the computation for testing
- ✓ Most time-consuming in development cycle
- ✓ Often hand-craft and task dependent in practice



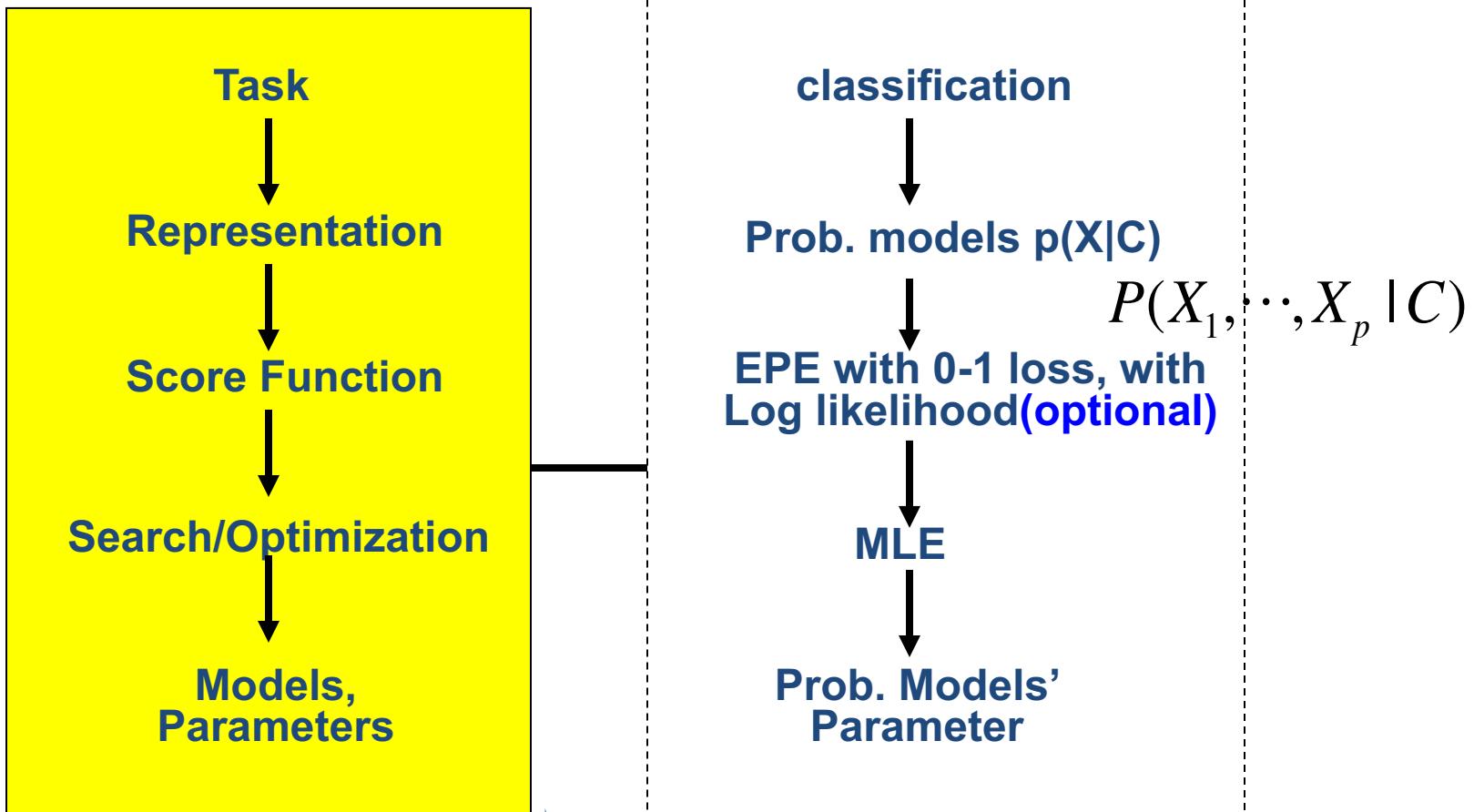
## Feature Learning

- ✓ Easily adaptable to new similar tasks
- ✓ Layerwise representation
- ✓ Layer-by-layer unsupervised training
- ✓ Layer-by-layer supervised training

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

Dr. Yanjun Qi / UDA CS

## (5) Generative Bayes Classifier



*Bernoulli  
Naïve*

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

*Gaussian  
Naïve*

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

*Multinomial*

11/25/19

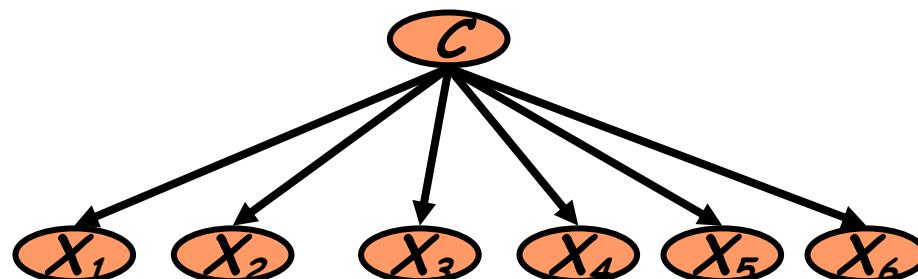
$$P(W_1 = n_1, \dots, W_v = n_v | c_k) = \frac{N!}{n_{1k}! n_{2k}! \dots n_{vk}!} \theta_{1k}^{n_{1k}} \theta_{2k}^{n_{2k}} \dots \theta_{vk}^{n_{vk}}$$

# Naïve Bayes Classifier

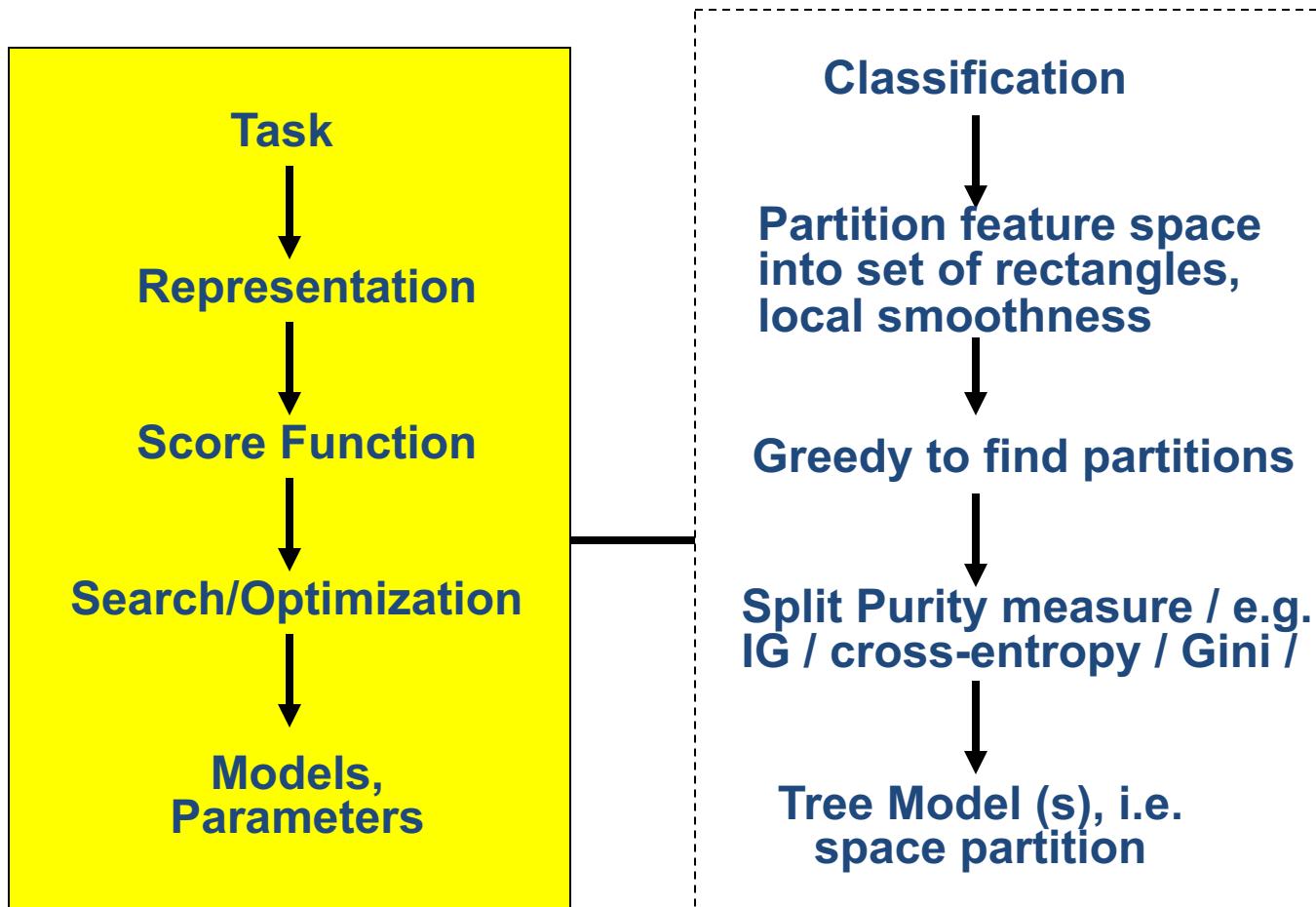
Difficulty: learning the joint probability  $P(X_1, \dots, X_p | C)$

- Naïve Bayes classification
  - Assumption that all input attributes are conditionally independent!

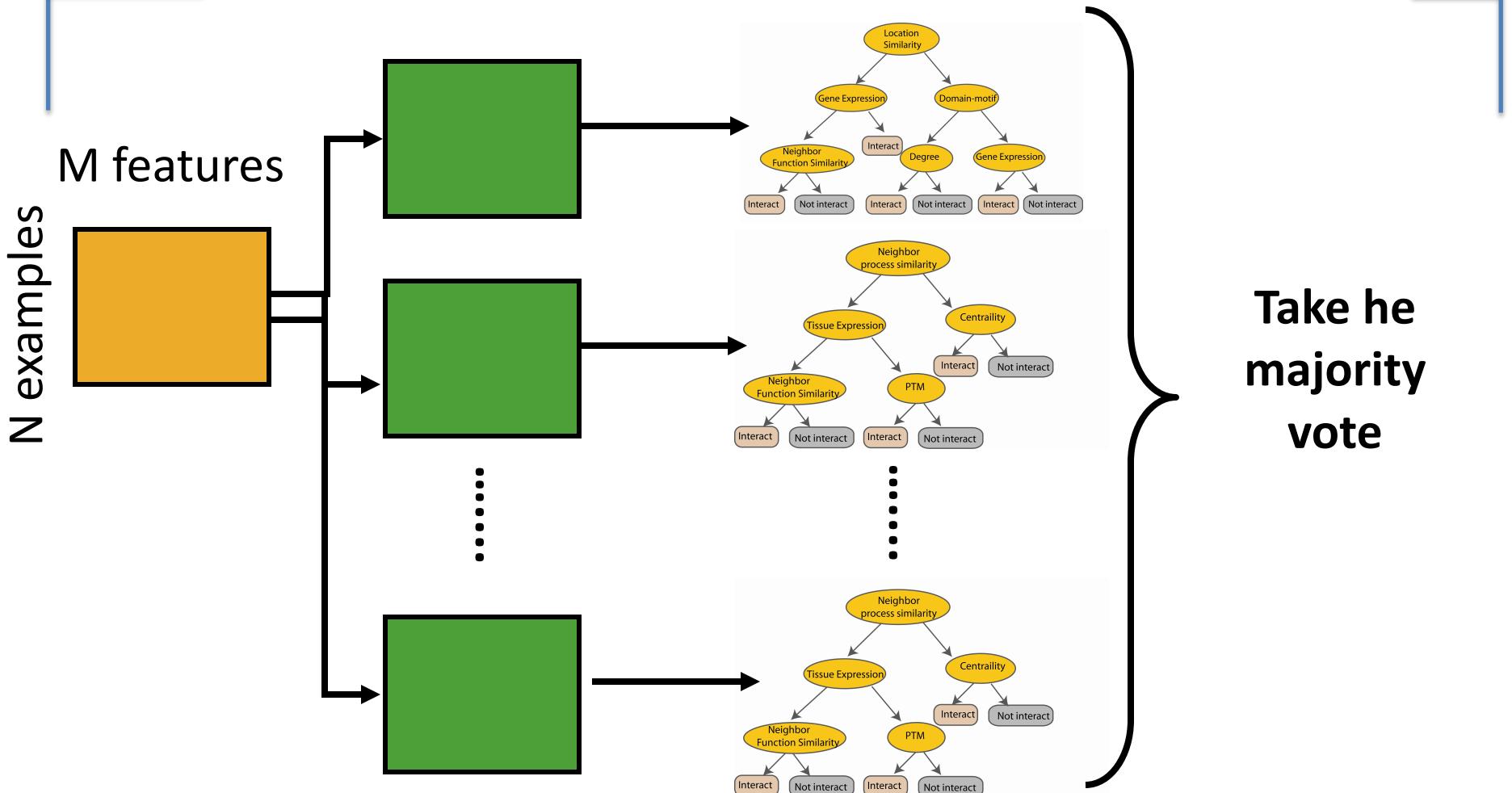
$$\begin{aligned}
 P(X_1, X_2, \dots, X_p | C) &= P(X_1 | X_2, \dots, X_p, C)P(X_2, \dots, X_p | C) \\
 &= \underline{P(X_1 | C)} \underline{P(X_2, \dots, X_p | C)} \\
 &= \underline{P(X_1 | C)} \underline{P(X_2 | C)} \cdots \underline{P(X_p | C)}
 \end{aligned}$$



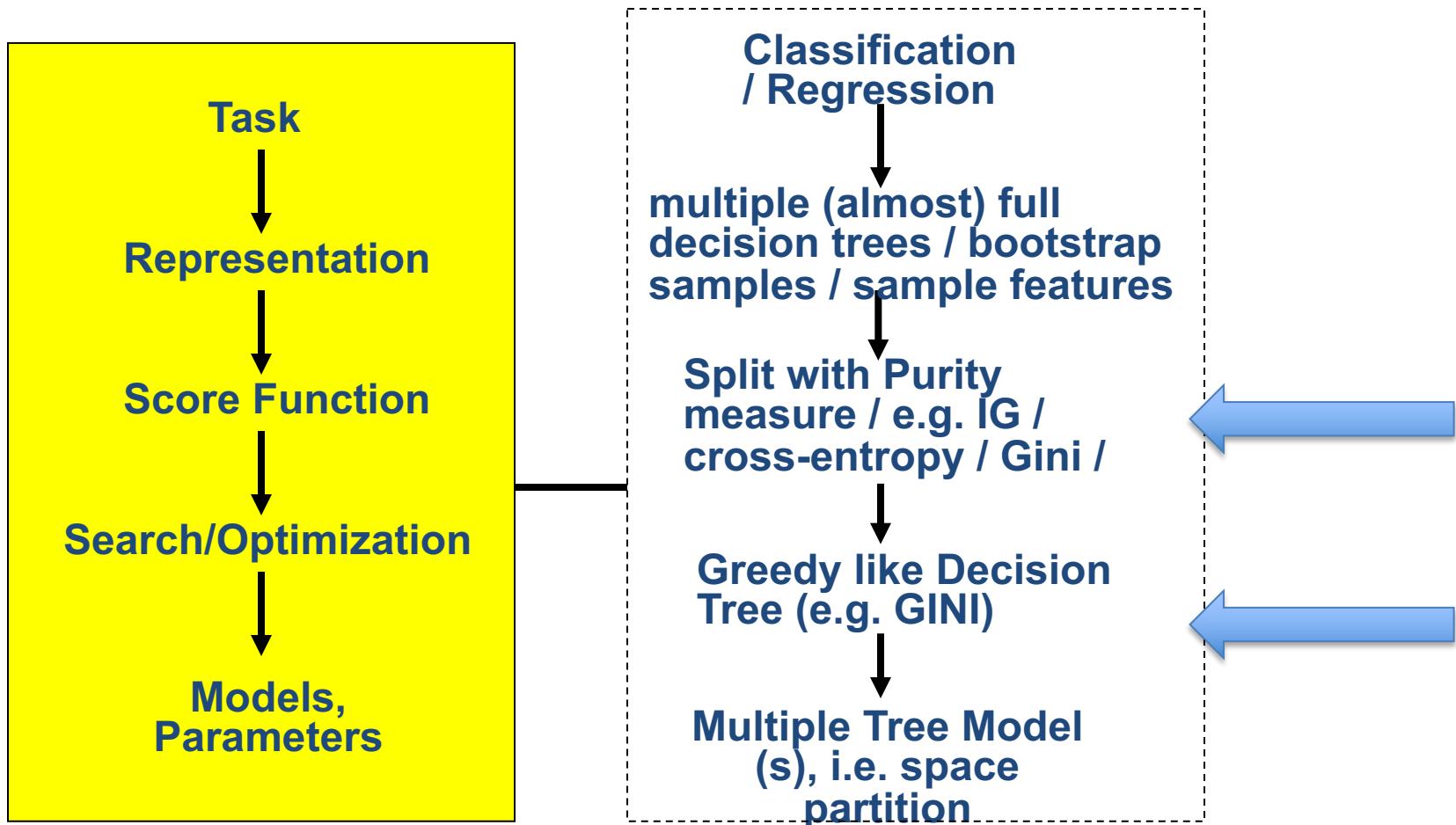
## (6) Decision Tree / Random Forest



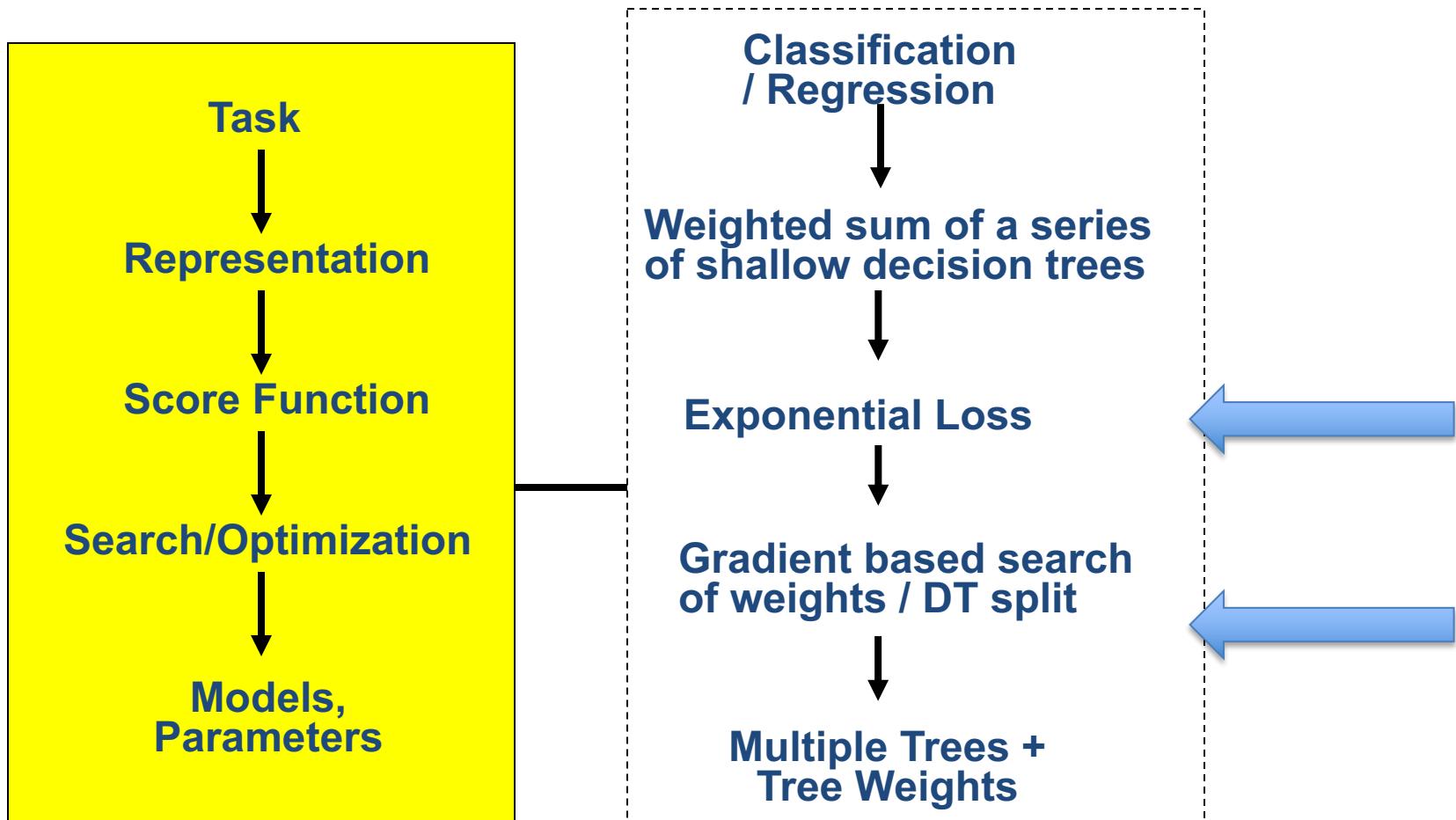
# Random Forest Classifier



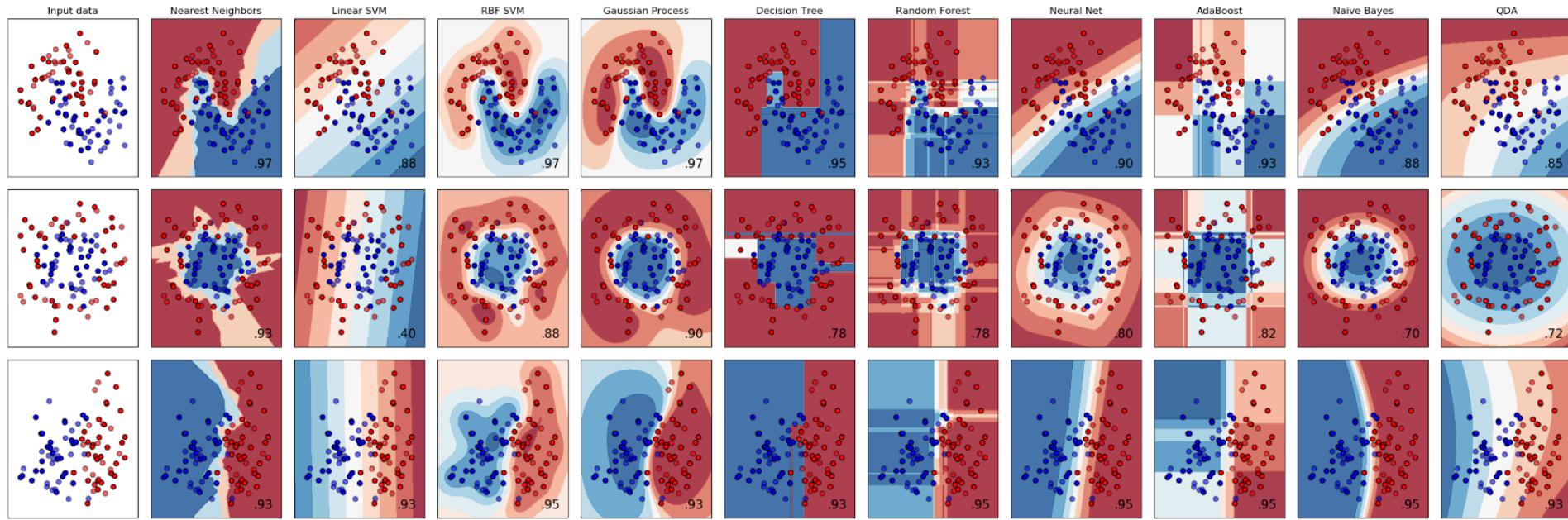
# Random Forest



# Boosting



[https://scikit-learn.org/stable/auto\\_examples/classification/plot\\_classifier\\_comparison.html](https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)



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

# Today

- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised)
  - ❑ Unsupervised models
  - ❑ Learning theory
- ❑ Review of Assignments covered so far

# What we have covered (III)

## ❑ Unsupervised models

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

	$X_1$	$X_2$	$X_3$
$s_1$			
$s_2$			
$s_3$			
$s_4$			
$s_5$			
$s_6$			

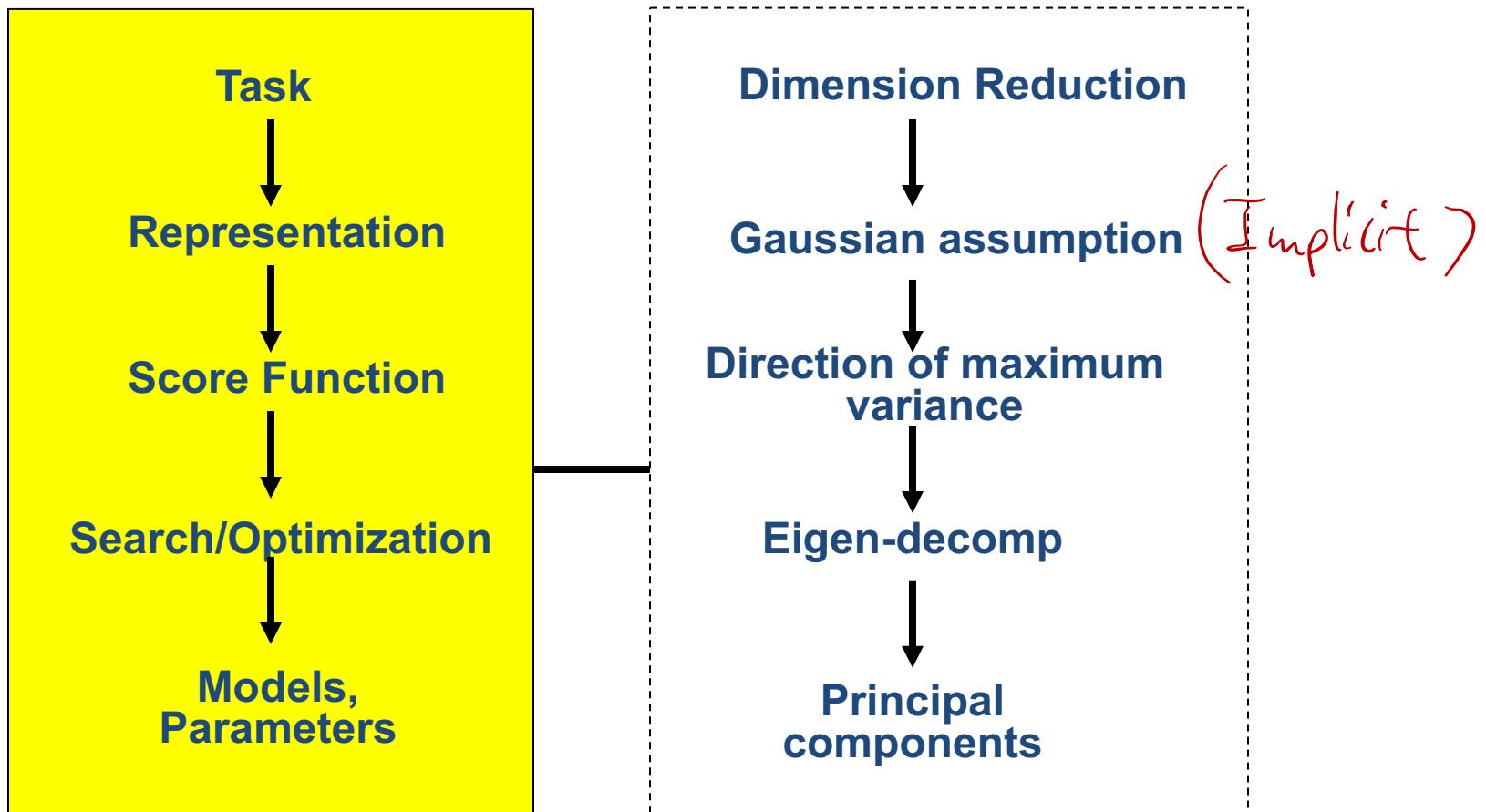
# An unlabeled Dataset X

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

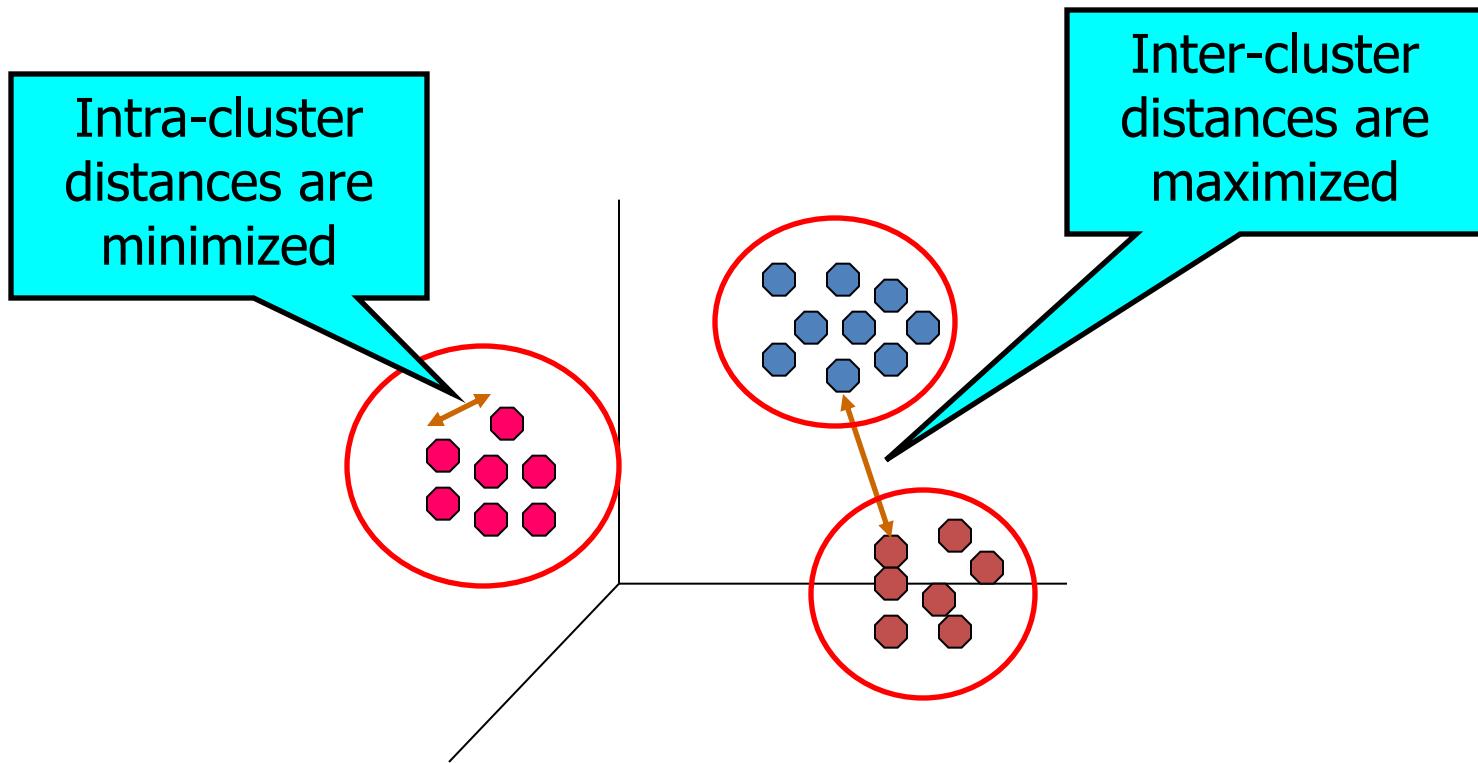
- Data/points/instances/examples/samples/records: [ rows ]
- Features/attributes/dimensions/independent variables/covariates/predictors/regressors: [ columns ]

# (0) Principal Component Analysis (optional)



# What is clustering?

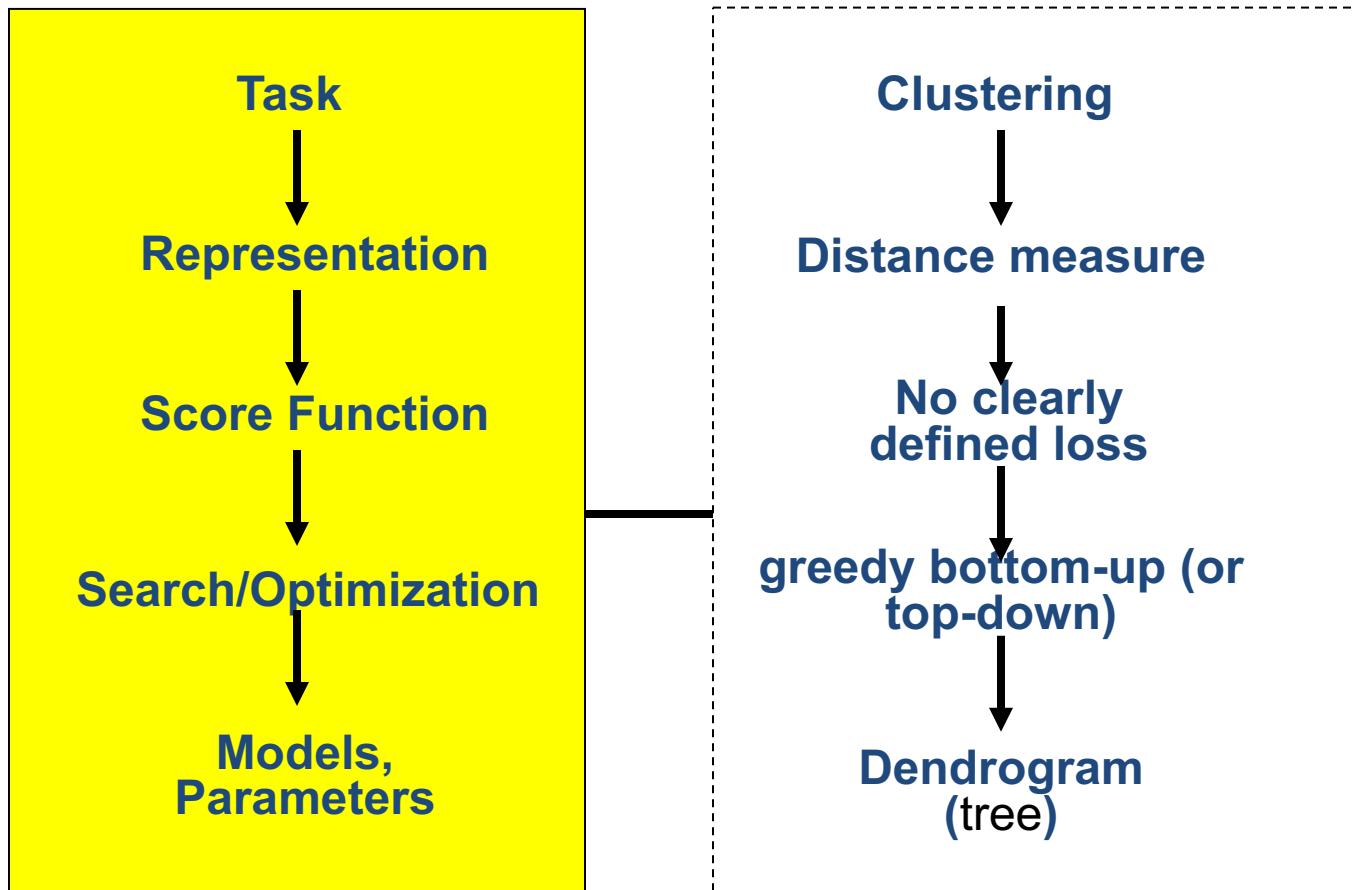
- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups



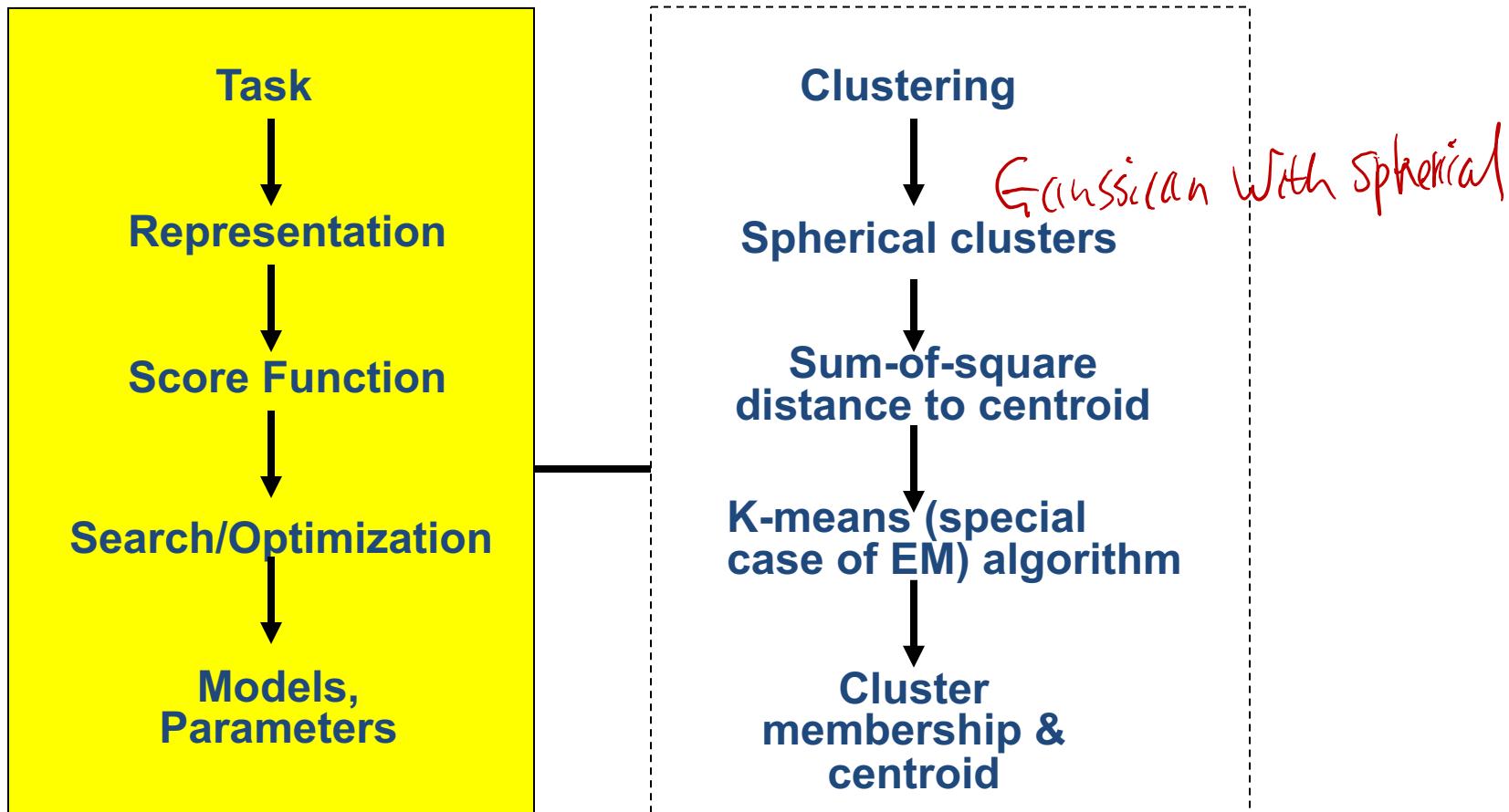
# Issues for clustering

- 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

# (1) Hierarchical Clustering

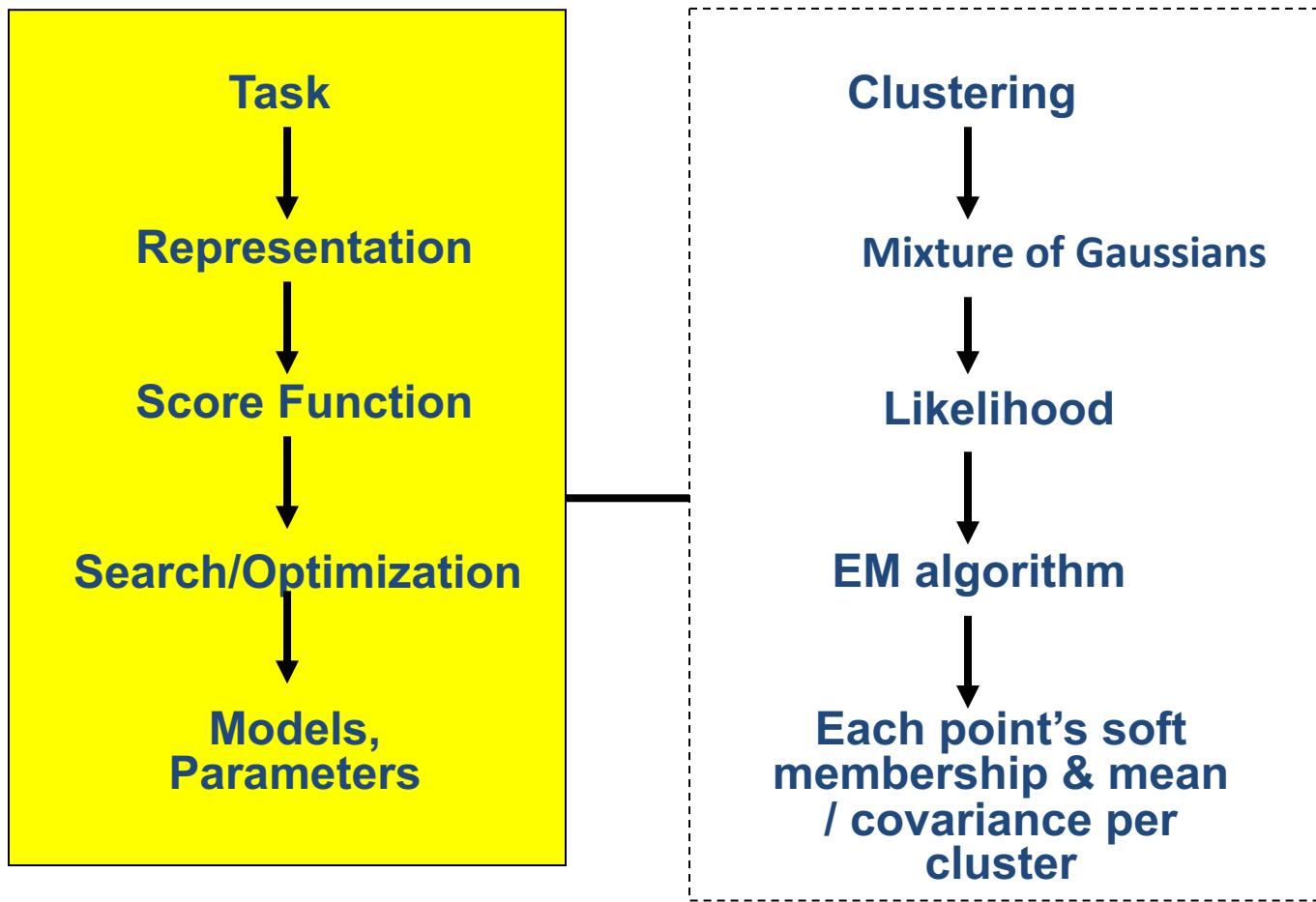


## (2) K-means Clustering



### (3) GMM Clustering

Dr. Yanjun Qi / UVA CS



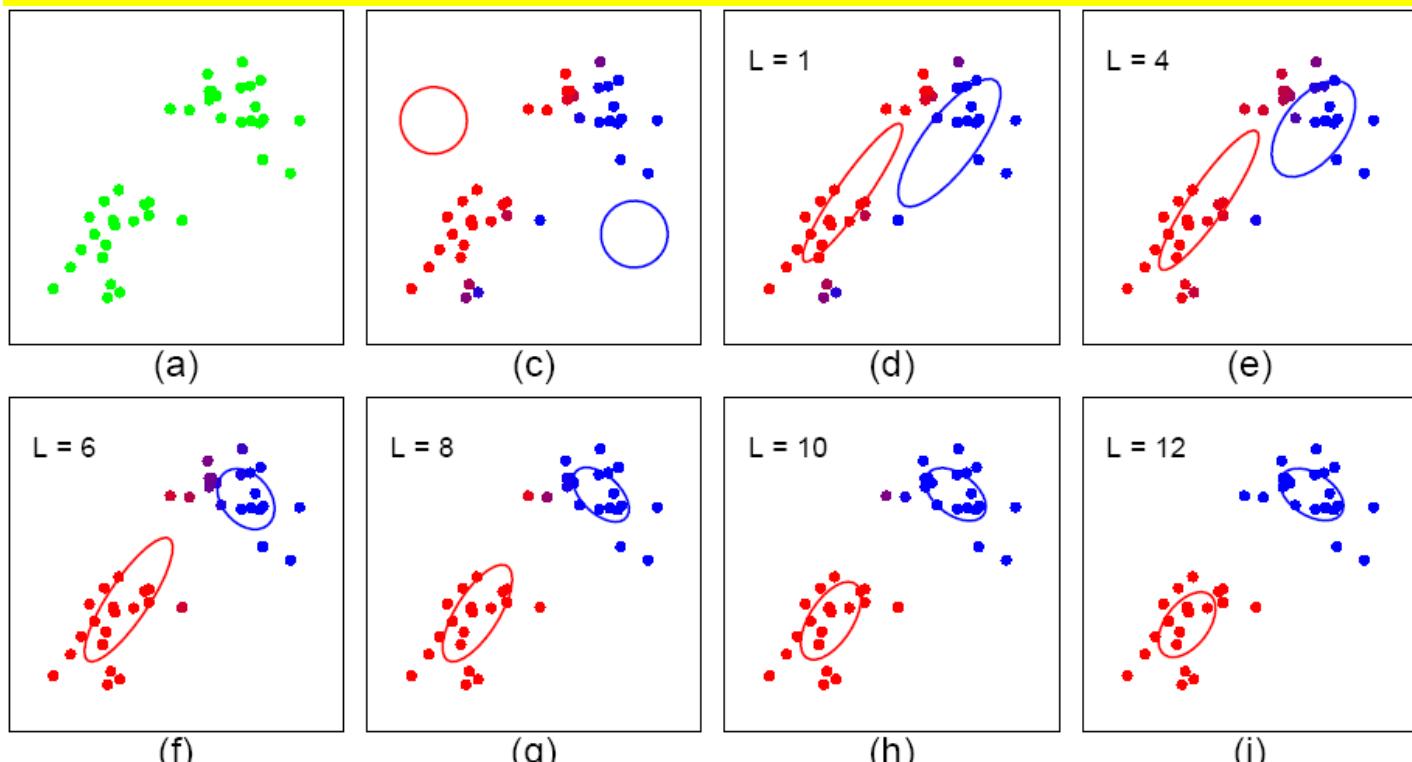
$$\sum_i \log \prod_{i=1}^n p(x = x_i) = \sum_i \log \left[ \sum_{\mu_j} p(\mu = \mu_j) \frac{1}{(2\pi)^{p/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2} (\vec{x} - \vec{\mu}_j)^T \Sigma_j^{-1} (\vec{x} - \vec{\mu}_j)} \right]$$

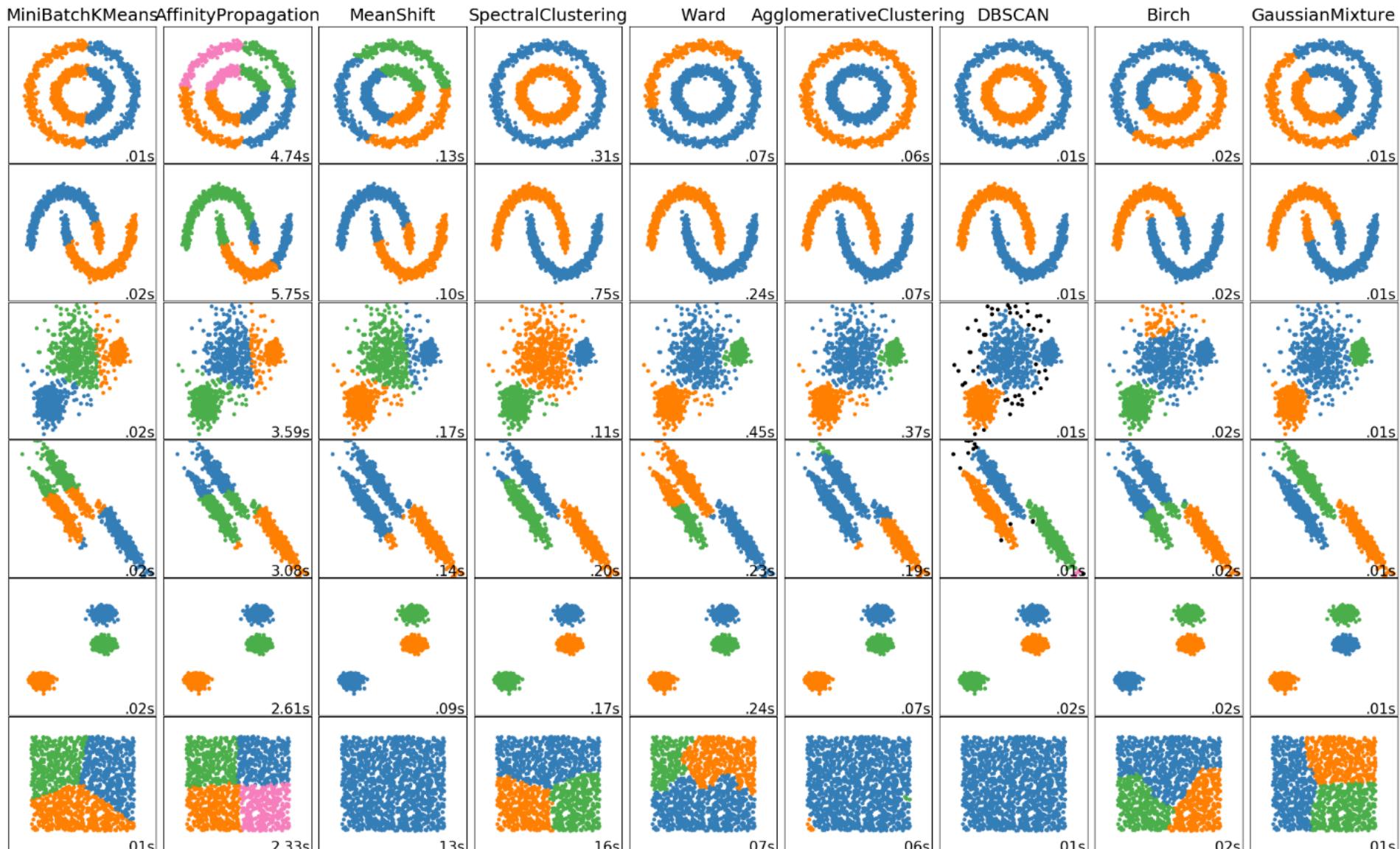
# Expectation-Maximization

Dr. Yanjun Qi / UVA CS

## 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)





- ✓ different assumptions on data
- ✓ different scalability profiles
- ✓ different model **sizes** (embedability in mobile devices)

# Recap

- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised)
  - ❑ Unsupervised models
  - ❑ Learning theory
- 
- ❑ Review of Assignments covered so far

# What we have covered (IV)

- ❑ Learning theory / Model selection
  - K-folds cross validation
  - Expected prediction error
  - Bias and variance tradeoff
  - Generative vs. Discriminative Classifiers

# CV-based Model Selection

Dr. Yanjun Qi / UVA CS

We're trying to decide which algorithm / hyperparameter to use.

- We train each model and make a table...

$i$	$f_i$	TRAINERR	10-FOLD-CV-ERR	Choice
1	$f_1$			
2	$f_2$			
3	$f_3$			✓
4	$f_4$			
5	$f_5$			
6	$f_6$			

Hyperparameter tuning ....

# Statistical Decision Theory (Extra)

- Random input vector:  $X$
  - Random output variable:  $Y$
  - Joint distribution:  $\Pr(X, Y)$
  - Loss function  $L(Y, f(X))$
  - Expected prediction error (EPE):
    - $\text{EPE}(f) = \mathbb{E}(L(Y, f(X))) = \int L(y, f(x)) \Pr(dx, dy)$   
e.g.  $= \int (y - f(x))^2 \Pr(dx, dy)$
- e.g. Squared error loss (also called L2 loss )

Consider population distribution

# Bias-Variance Trade-off for EPE: (Extra)

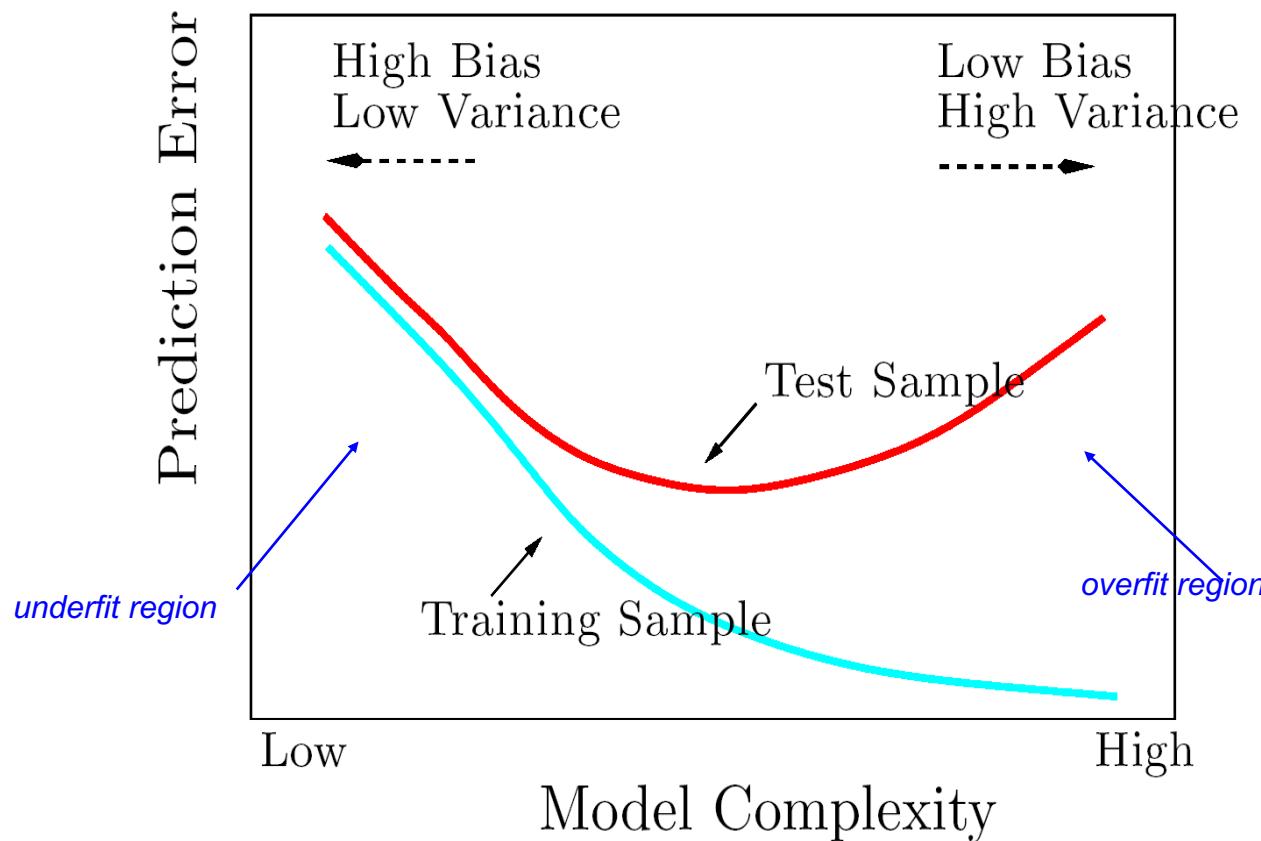
$$\text{EPE} = \text{noise}^2 + \text{bias}^2 + \text{variance}$$

Unavoidable  
error

Error due to  
incorrect  
assumptions

Error due to  
variance of training  
samples

# Bias-Variance Tradeoff / Model Selection



# Model “bias” & Model “variance”

- Middle RED:

- TRUE function

$\theta$   
[middle red]

- Error due to bias:

- How far off in general from the middle red

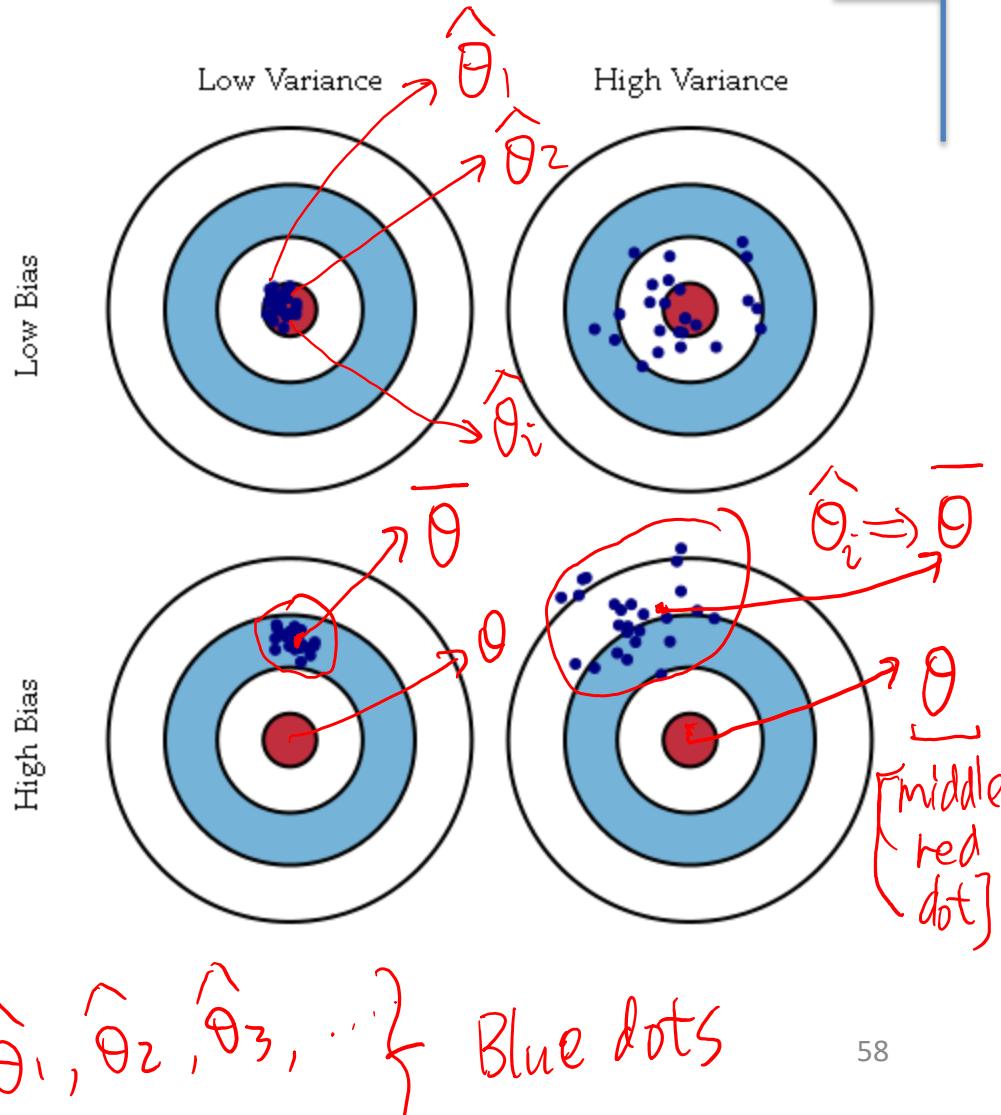
$$E(\theta - \bar{\theta})$$

mean of  $\hat{\theta}$

- Error due to variance:

- How wildly the blue points spread

$$E((\hat{\theta} - \bar{\theta})^2)$$



# 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**

# Logistic Regression vs. Naïve /LDA

## Discriminative classifier (Logistic Regression)

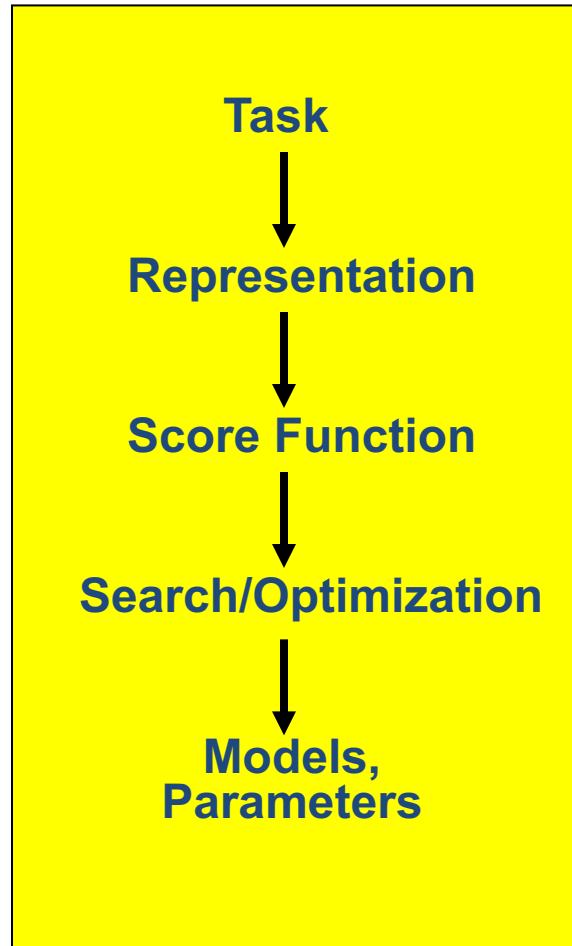
- Smaller asymptotic error
- Slow convergence  $\sim O(p)$

## Generative classifier (Naive Bayes)

- Larger asymptotic error
- Can handle missing data (EM)
- Fast convergence  $\sim O(\lg(p))$

the speed at which a convergent sequence approaches its limit is called the rate of convergence.

# 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

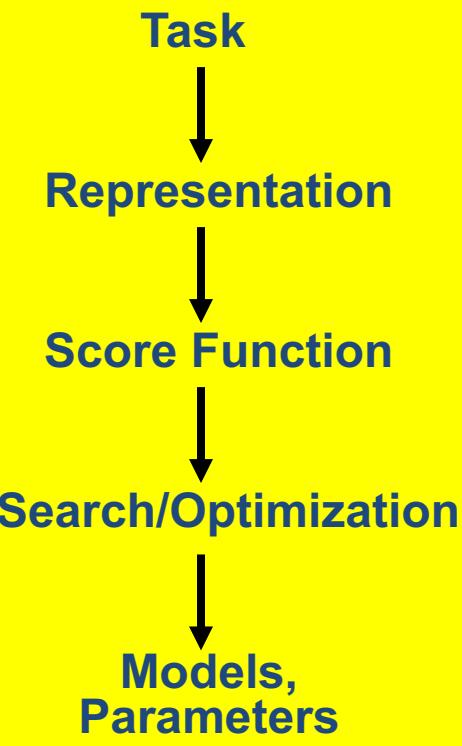
<b>Task</b>	Regression, classification, clustering, dimen-reduction
<b>Representation</b>	Linear func, nonlinear function (e.g. polynomial expansion), local linear, logistic function (e.g. $p(c x)$ ), tree, multi-layer, prob-density family (e.g. Bernoulli, multinomial, Gaussian, mixture of Gaussians), local func smoothness, kernel matrix, local smoothness, partition of feature space,
<b>Score Function</b>	MSE, Margin, log-likelihood, EPE (e.g. L2 loss for KNN, 0-1 loss for Bayes classifier), cross-entropy, cluster points distance to centers, variance, conditional log-likelihood, complete data-likelihood, regularized loss func (e.g. L1, L2) , goodness of inter-cluster similar
<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, backprop
<b>Models, Parameters</b>	Linear weight vector, basis weight vector, local weight vector, dual weights, training samples, tree-dendrogram, multi-layer weights, principle components, member (soft/hard) assignment, cluster centroid, cluster covariance (shape), ...

# Recap

- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised)
  - ❑ Unsupervised models
  - ❑ Learning theory

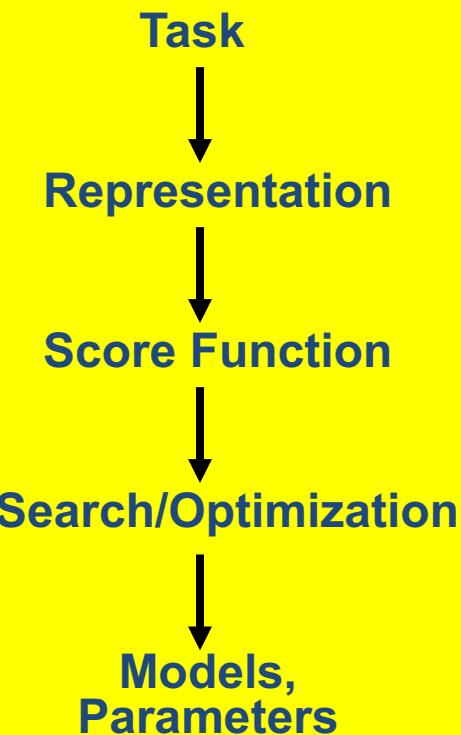
- 
- ❑ Review of Assignments covered so far

# HW1



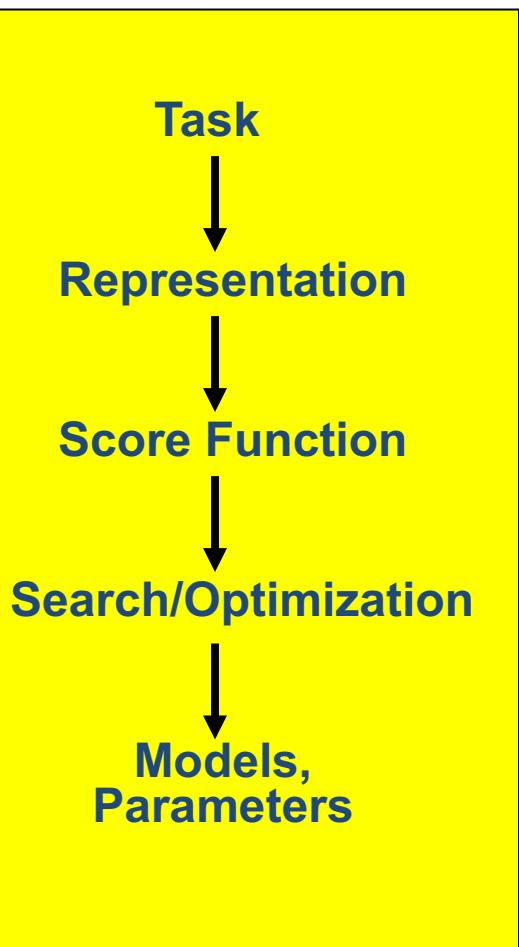
- Q1: Linear algebra review
- Q2: Linear regression model fitting
  - Data loading
  - Basic linear regression
  - Ways to train : Normal equation / SGD / Batch GD
- Sample exam Q:
  - regression model fitting

# HW2



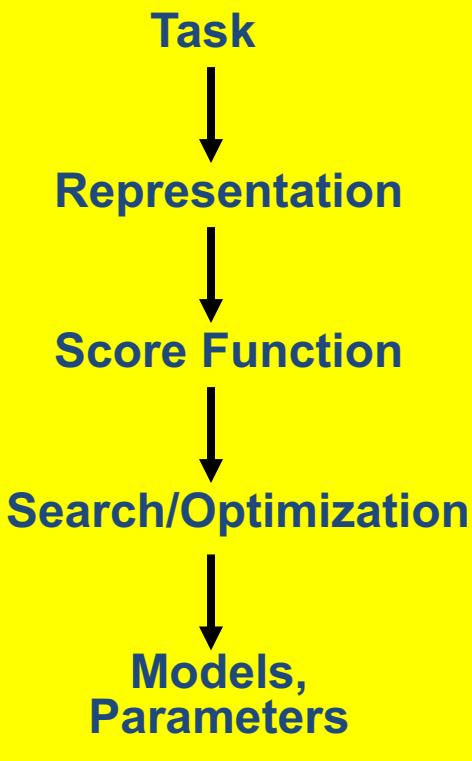
- Q1: Polynomial regression
- Q2: Ridge regression
  - Math derivation of ridge
  - Understand why/how Ridge
  - Model selection of Ridge with KCV
- Sample Qs:
  - Regularization

# HW3



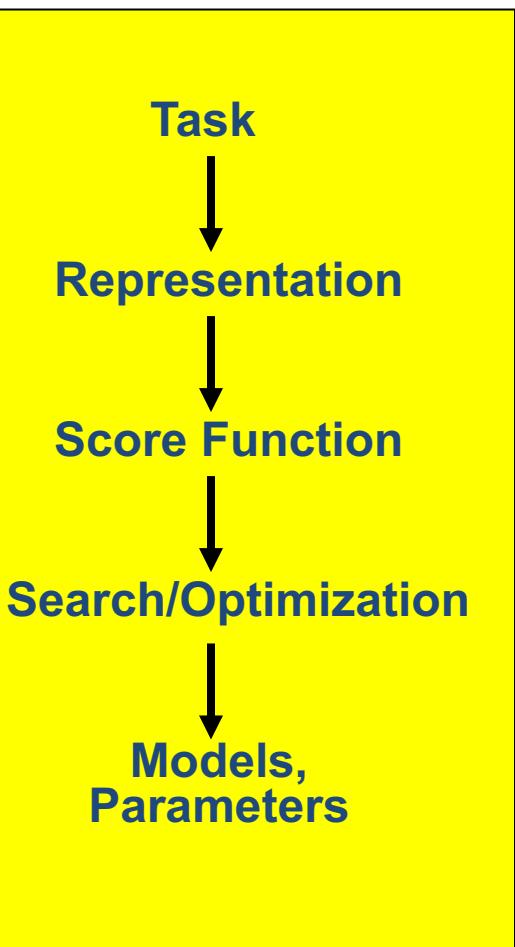
- Q1: KNN and model selection
- Q2: Support Vector Machines with Scikit-Learn
  - Data preprocessing
  - How to use SVM package
  - Model selection for SVM
  - Model selection pipeline with train-vali, or train-CV; then test
- Sample QAs:
  - SVM

# HW4



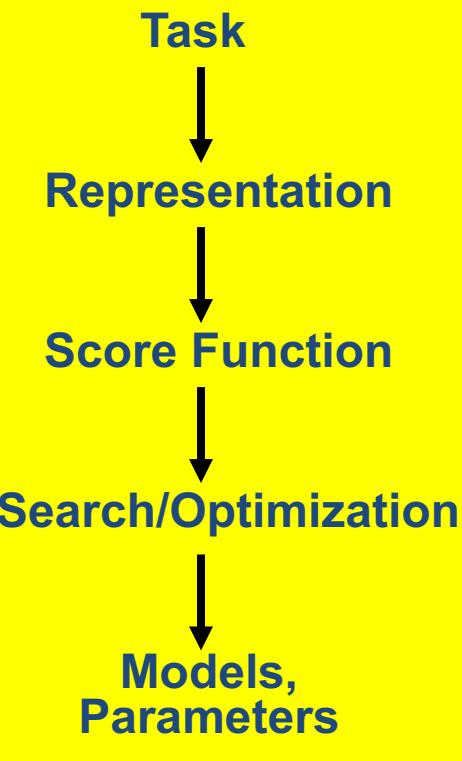
- Q1: Neural Network Tensorflow Playground
  - Interactive learning of MLP
  - Feature engineering vs.
  - Feature learning
- Q2: Image Classification /Keras
  - DNN Tool: Keras using
  - Extra: PCA for image classification
- Sample Qs:
  - Neural Nets

# HW5



- Q1: Naive Bayes Classifier for Text-based Movie Review Classification
  - Preprocessing of text samples
  - BOW Document Representation
  - Multinomial Naive Bayes Classifier BOW way
  - Multivariate Bernoulli Naive Bayes Classifier
- Sample Qs:
  - Bayes Classifier

# HW6



- Q1: Unsupervised Clustering of audio data and consensus data
  - Data loading
  - K-mean clustering
  - How to find K: knee-finding plot
  - How to measure clustering: purityMetric
- Sample Qs:
  - Kmeans and GMM
  - Decision trees

# Highly Recommend

## Four Extra-curriculum books

- 1. Book - Algorithms to Live By: The Computer Science of Human Decisions
  - [https://books.google.com/books/about/Algorithms to Live By The Computer Scien.html?id=xmeJCgAAQBAJ&source=kp\\_book\\_description](https://books.google.com/books/about/Algorithms_to_Live_By_The_Computer_Scien.html?id=xmeJCgAAQBAJ&source=kp_book_description)
  - This book provides a fascinating exploration of how computer algorithms can be applied to our everyday lives.

# Highly Recommend

## Four Extra-curriculum books

- 2. A Mind for Numbers: How to Excel at Math and Science (Even If You Flunked Algebra)
  - <https://www.goodreads.com/book/show/18693655-a-mind-for-numbers>
  - A good summary <https://fastertomaster.com/a-mind-for-numbers-barbara-oakley/>
  - The book is geared toward helping students study well. Despite the title for math, most of the advice in this book is appropriate for just about any subject.

# Highly Recommend

## Four Extra-curriculum books

- 3. Book: **So Good They Cannot Ignore You - Why Skills Trump Passion in the Quest for Work You Love**
  - [https://www.amazon.com/Good-They-Cant-Ignore-You-ebook/dp/B0076DDBJ6/ref=tmm\\_kin\\_swatch\\_0?encoding=UTF8&qid=1497747881&sr=1-1](https://www.amazon.com/Good-They-Cant-Ignore-You-ebook/dp/B0076DDBJ6/ref=tmm_kin_swatch_0?encoding=UTF8&qid=1497747881&sr=1-1)
  - The idea of Career capital - rare and valuable skills need deliberate practice
  - 10,000 hours of deliberate practice → Expert!

# Highly Recommend

## Four Extra-curriculum books

- 4. Book: Homo Deus- A Brief History of Tomorrow
  - <https://www.goodreads.com/book/show/3113856-homo-deus>
  - “Homo Deus explores the projects, dreams and nightmares that will shape the twenty-first century—from overcoming death to creating artificial life. It asks the fundamental questions: Where do we go from here? And how will we protect this fragile world from our own destructive powers? This is the next stage of evolution. This is Homo Deus.””

# Highly Recommend

## Four Extra-curriculum books

- 4. Book: Homo Deus- A Brief History of Tomorrow
  - <https://www.goodreads.com/book/show/3113856-homo-deus>
  - “Homo Deus explores the projects, dreams and nightmares that will shape the twenty-first century—from overcoming death to creating artificial life. It asks the fundamental questions: Where do we go from here? And how will we protect this fragile world from our own destructive powers? This is the next stage of evolution. This is Homo Deus.””

# Highly Recommend One more:

## 5. Machine Learning book from Dr. Domingos: Master Algorithm

So How Do Computers Discover New Knowledge?

1. **Symbolists**--Fill in gaps in existing knowledge
2. **Connectionists**--Emulate the brain
3. **Evolutionists**--Simulate evolution
4. **Bayesians**--Systematically reduce uncertainty
5. **Analogizers**--Notice similarities between old and new

SRC: Pedro Domingos ACM Webinar Nov 2015

<http://learning.acm.org/multimedia.cfm>

# Combining The Five Tribes of Machine Learning: Single algorithm or Master Algorithm

Tribe	Origins	Key Algorithm
Symbolists	Logic, philosophy	Inverse deduction
Connectionists	Neuroscience	Backpropagation
Evolutionists	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

SRC: Pedro Domingos ACM Webinar Nov 2015

<http://learning.acm.org/multimedia.cfm>

# 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