UVA CS 6316: Machine Learning

Lecture 20: Review

Dr. Yanjun Qi

University of Virginia
Department of Computer Science

Final Review

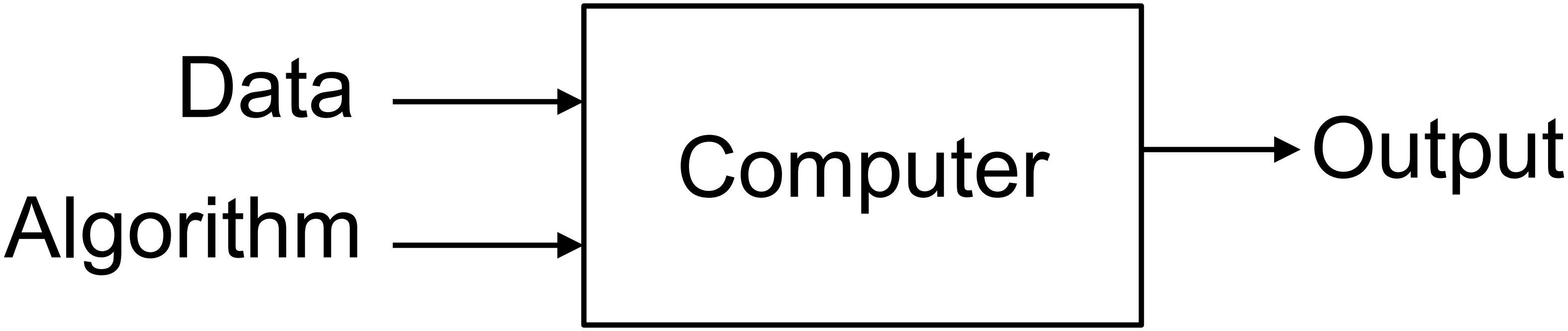
- Five Tribes of Machine Learning
- Review of ML methods covered so far
  - Regression (supervised)
  - Classification (supervised)
  - Unsupervised models
  - Learning theory
  - Review of Six Assignments
- Four books to recommend

Objective

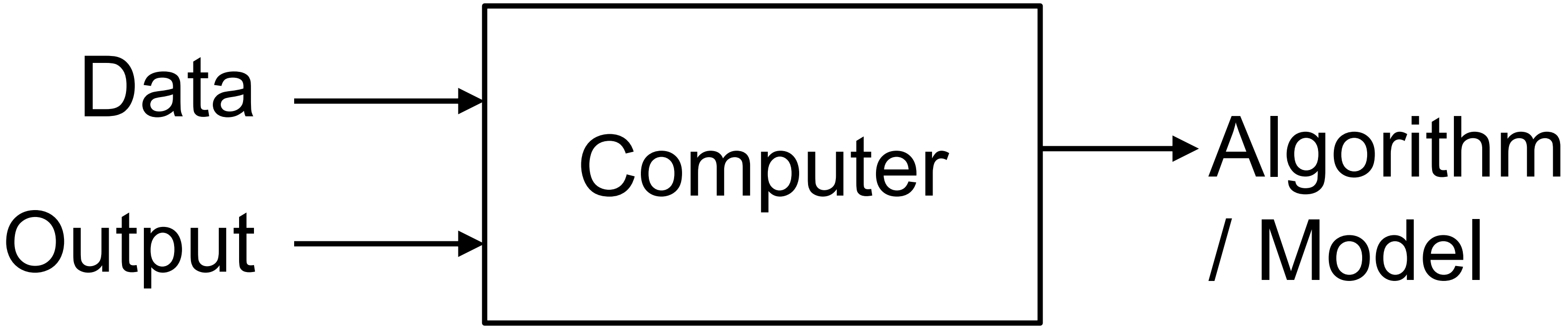
- To help students be able to build machine learning tools
  - (not just a tool user!!!)

- 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 and equation level

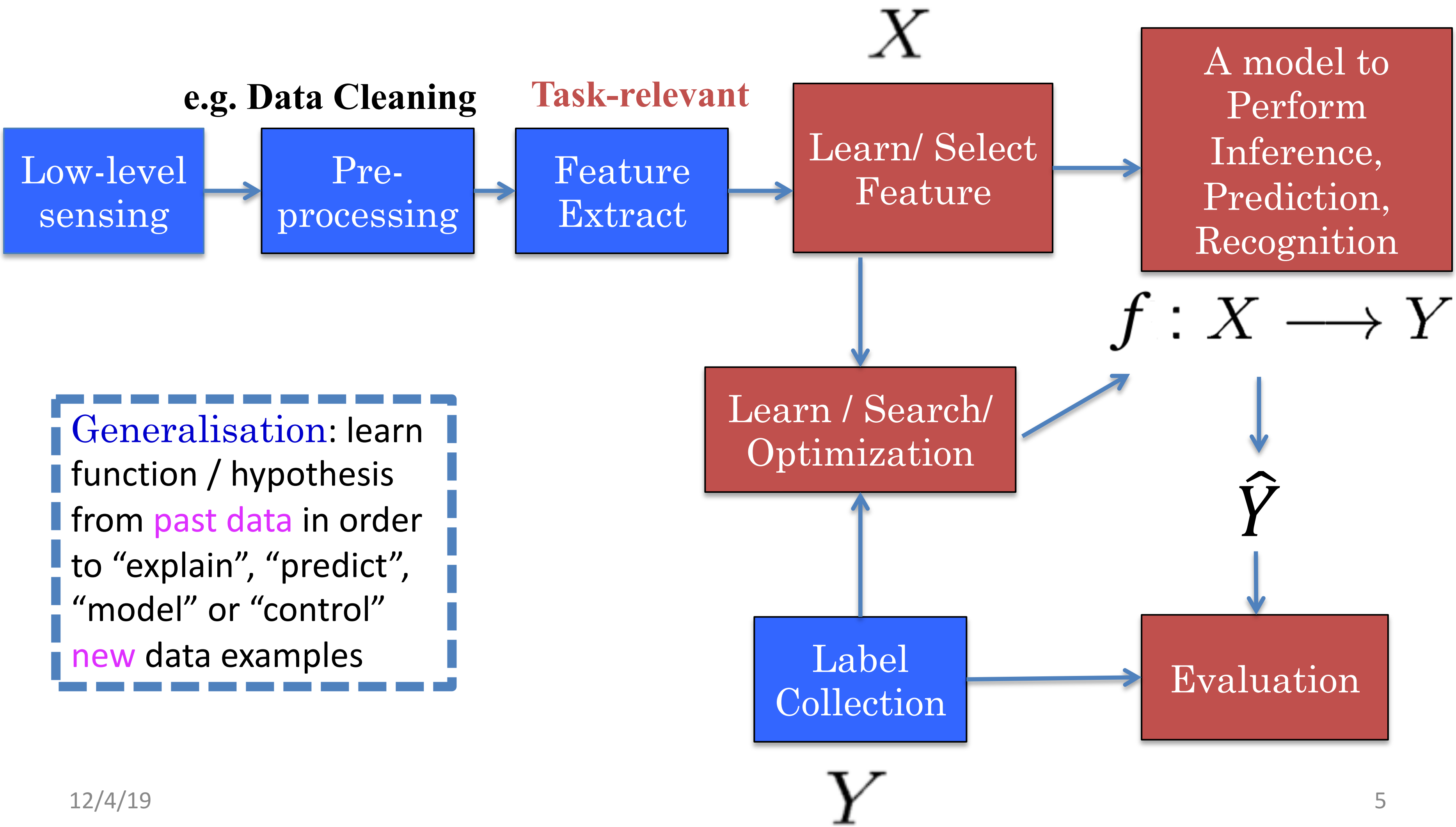
Traditional Programming



Machine Learning

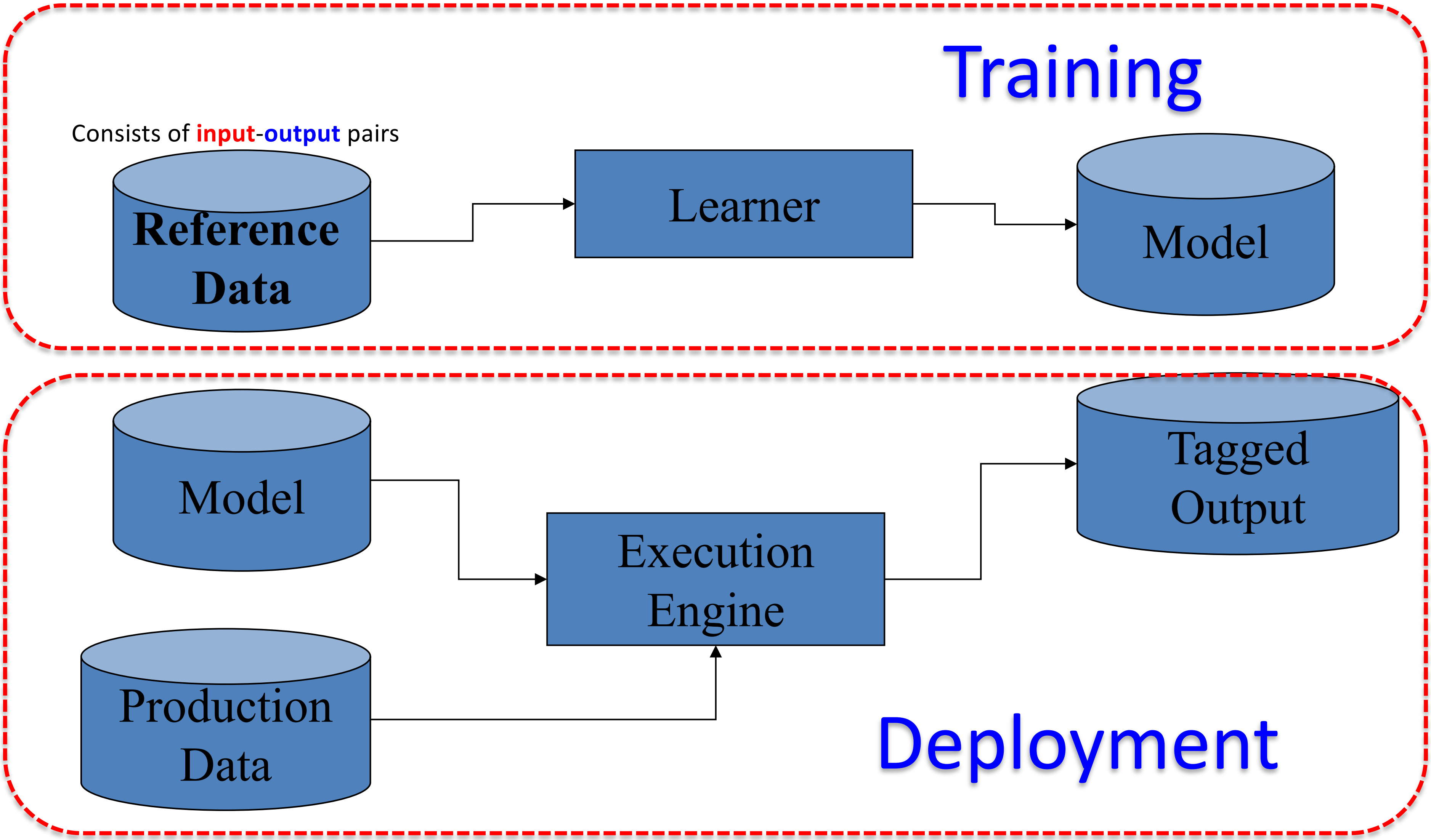


A Typical Machine Learning Application's Pipeline



Generalisation: learn function / hypothesis from past data in order to "explain", "predict", "model" or "control" new data examples

Two Modes of Machine Learning



Course Content Plan → Six major sections of this course

- ❑ ~~Regression (supervised)~~
- ❑ ~~Classification (supervised)~~
  - ❑ ~~Feature Selection~~
- ❑ Unsupervised models
  - ❑ ~~Dimension Reduction (PCA)~~
  - ❑ ~~Clustering (K-means, GMM/EM, Hierarchical)~~
- ❑ Learning theory
- ❑ Graphical models
- ❑ Reinforcement Learning

Y is a continuous

Y is a discrete

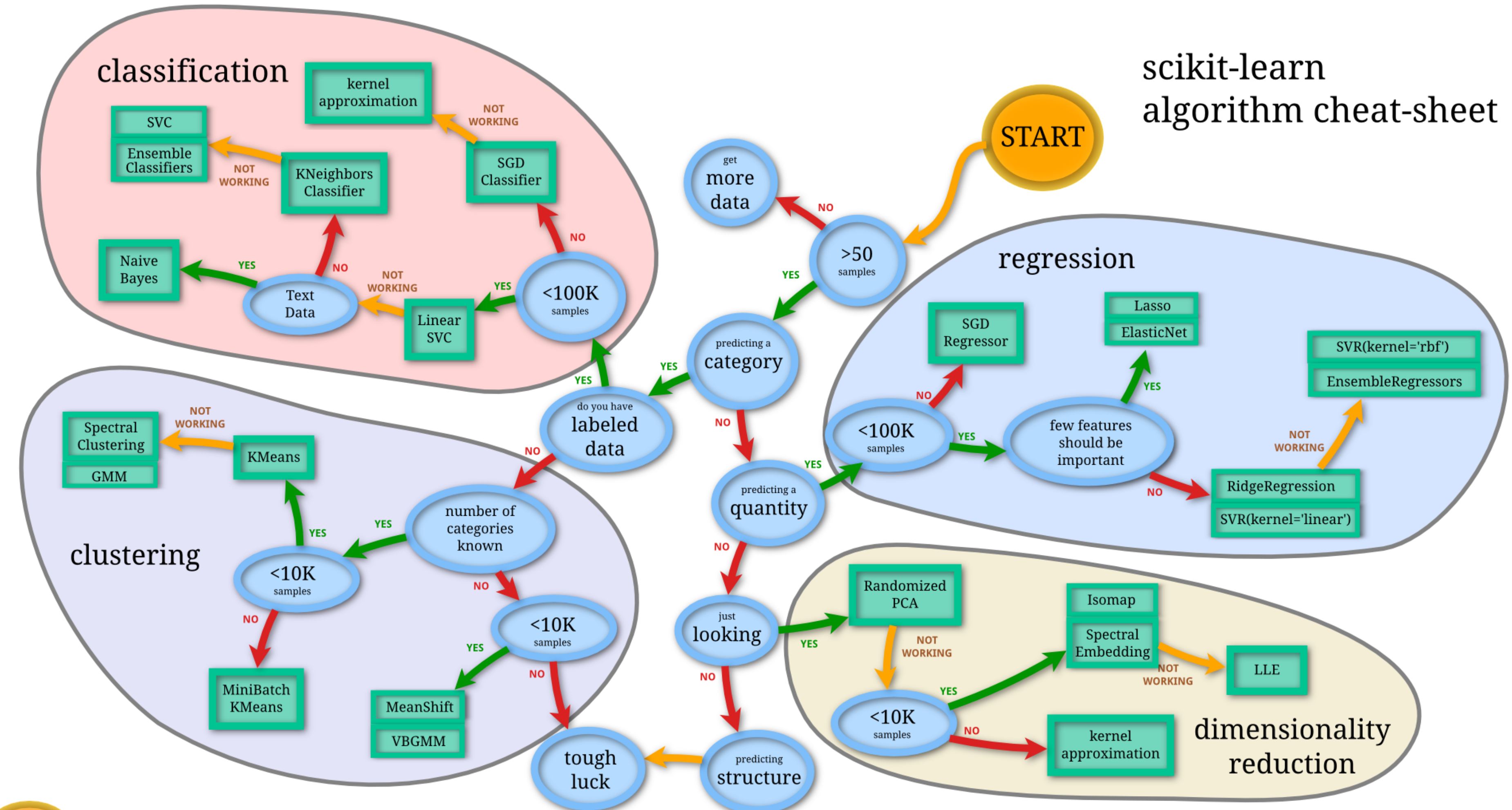
NO Y

About f()

About interactions among X1,… Xp

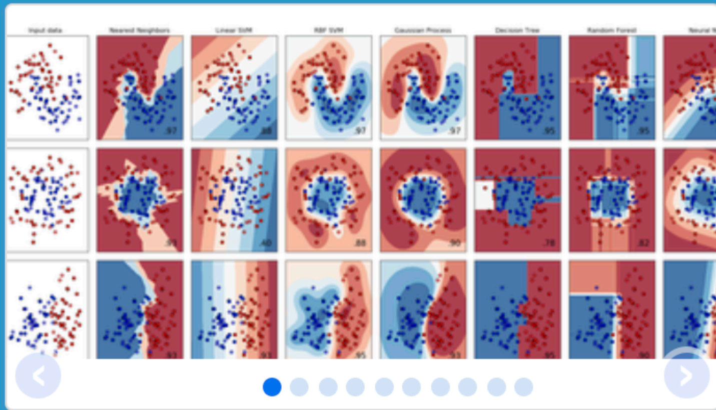
Learn program to Interact with its environment http://scikit-learn.org/stable/tutorial/machine_learning_map/

Scikit-learn algorithm cheat-sheet



Back

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

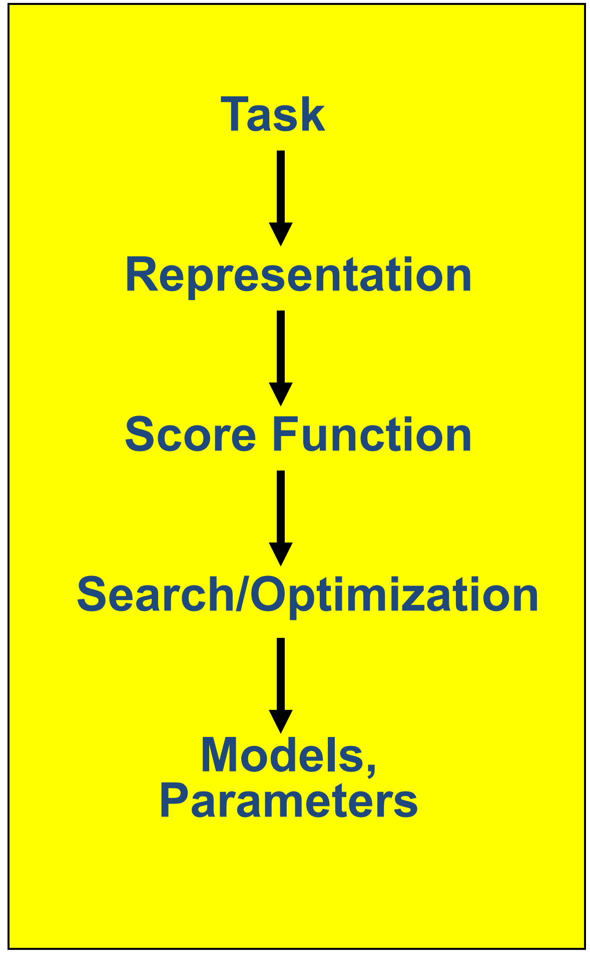
Highly Recommend One Book: 0. By Dr. Domingos: Master Algorithm

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

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

The Five Tribes of Machine Learning:

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

From: Dr. Pedro Domingos

Symbolists

Tom Mitchell

Steve Muggleton

Ross Quinlan

Tribe	Origins	Key Algorithm
Symbolists	Logic, philosophy	Inverse deduction

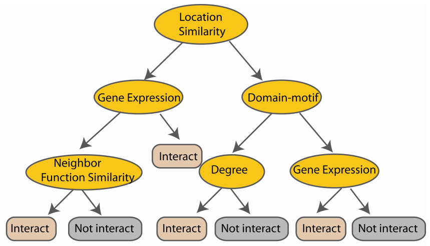
From: Dr. Pedro Domingos

e.g., Decision Tree-building algorithms (1990s)

ID3: Iterative Dichotomiser 3. Developed in the 80s by Ross Quinlan.

C4.5: Successor of ID3, also developed by Quinlan ('93). Main improvements over I3D:

Adaboost: by Robert Schapire (1999)



Connectionists

Yann LeCun

Geoff Hinton

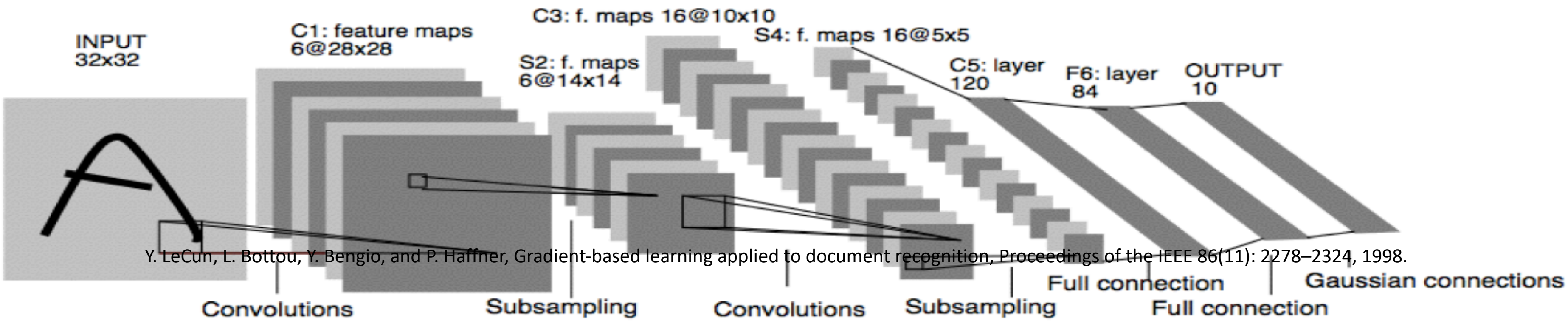
Yoshua Bengio

Tribe	Origins	Key Algorithm
Connectionists	Neuroscience	Backpropagation

From: Dr. Pedro Domingos

Deep Learning (CNN) in the 90's

- Prof. Yann LeCun invented Convolutional Neural Networks (CNN) in 1998
- First NN successfully trained with many layers



Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.

Evolutionaries

John Koza

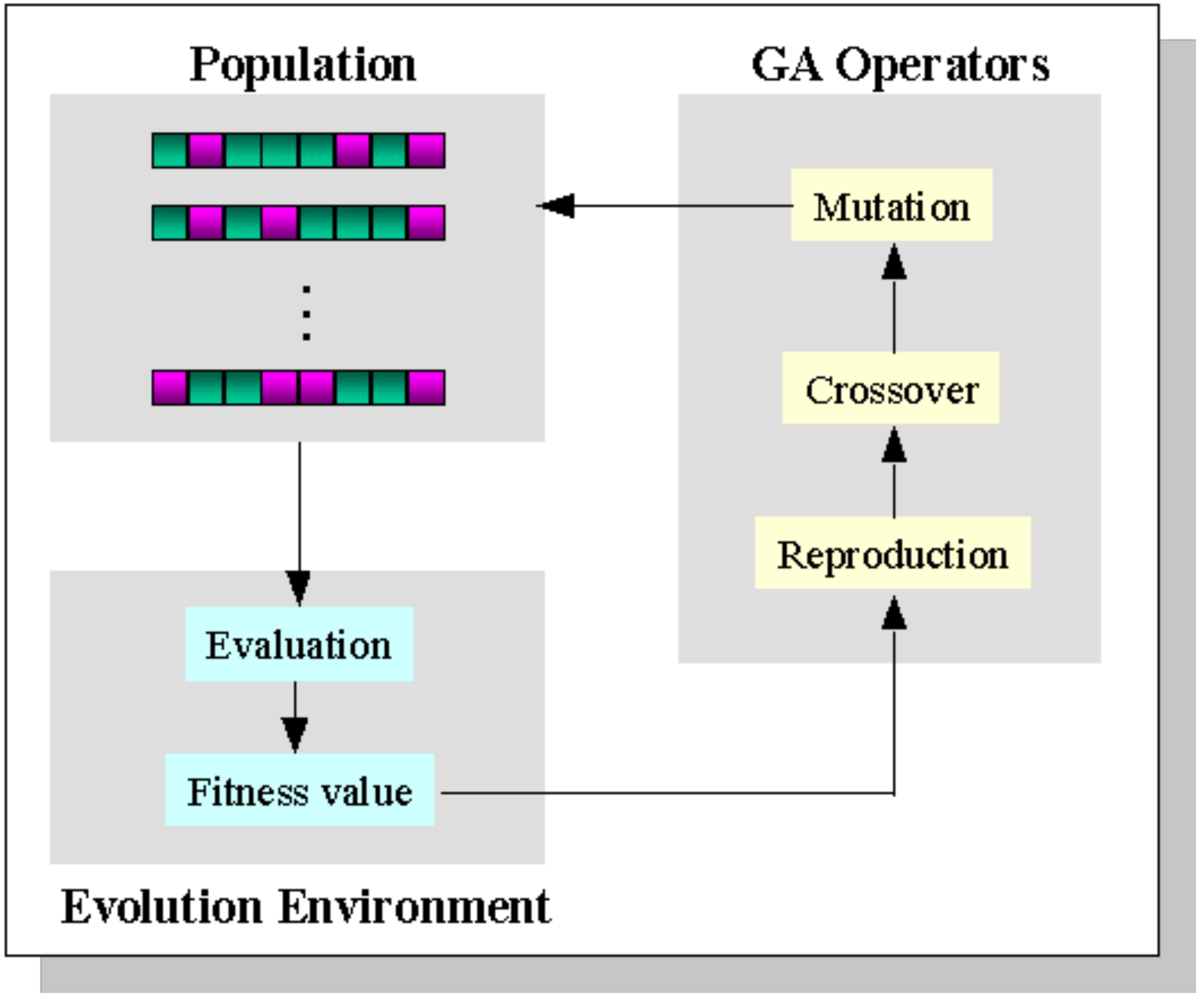
John Holland

Hod Lipson

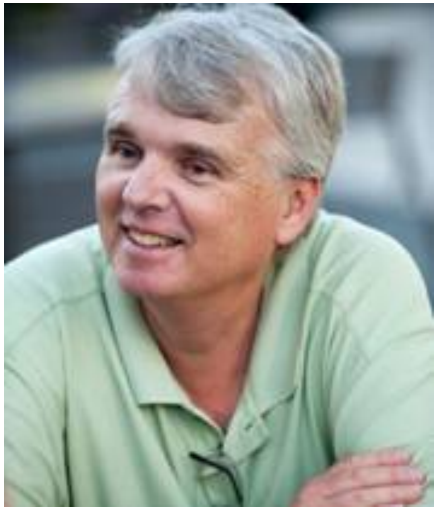
Tribe	Origins	Key Algorithm
Evolutionists	Evolutionary biology	Genetic programming

From: Dr. Pedro Domingos

Genetic Algorithms



Bayesians



David Heckerman　　Judea Pearl　　Michael Jordan

Tribe	Origins	Key Algorithm
Bayesians	Statistics	Probabilistic inference

From: Dr. Pedro Domingos

Probabilistic Inference

Likelihood

How probable is the evidence given that our hypothesis is true?

Prior

How probable was our hypothesis *before* observing the evidence?

$$P(H \mid e) = \frac{P(e \mid H)\, P(H)}{P(e)}$$

Posterior

How probable is our hypothesis given the observed evidence? (Not directly computable)

Marginal

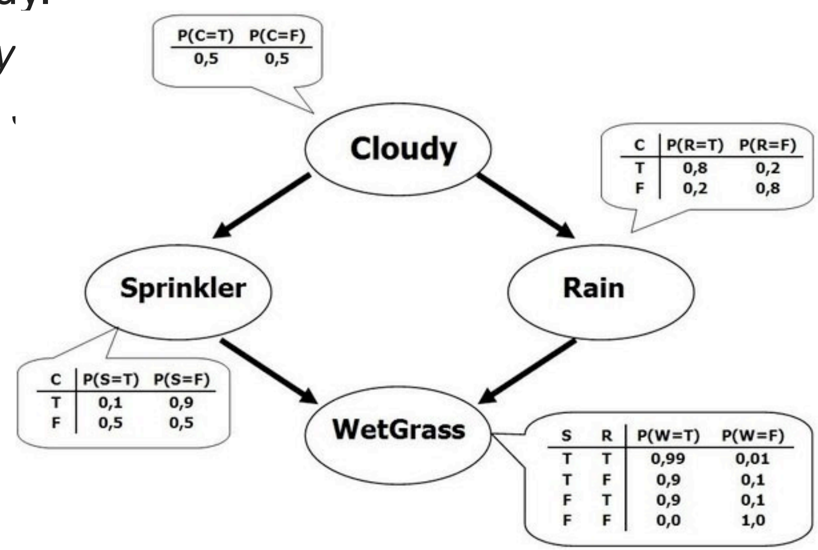
How probable is the new evidence under all possible hypotheses?

$P(e) = \sum P(e \mid H_i)\, P(H_i)$

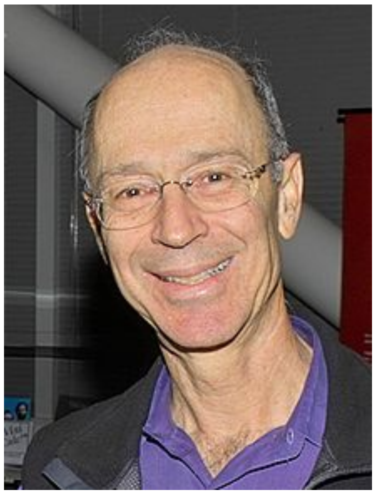
From: Dr. Pedro Domingos

Reasoning with uncertainty

- "Bayesian network" was termed by Judea Pearl in 1985
- Bayes' conditioning is the basis for updating information in the graph
- The distinction between causal and evidential modes of reasoning
- In the late 1980s, established as a field of study.
  - Pearl's *Probabilistic Reasoning in Intelligent Sy*
  - Neapolitan's *Probabilistic Reasoning in Expert*

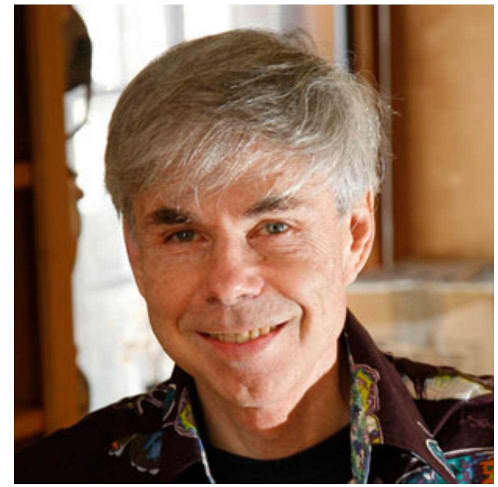


Analogizers



Peter Hart

Vladimir Vapnik



Douglas Hofstadter

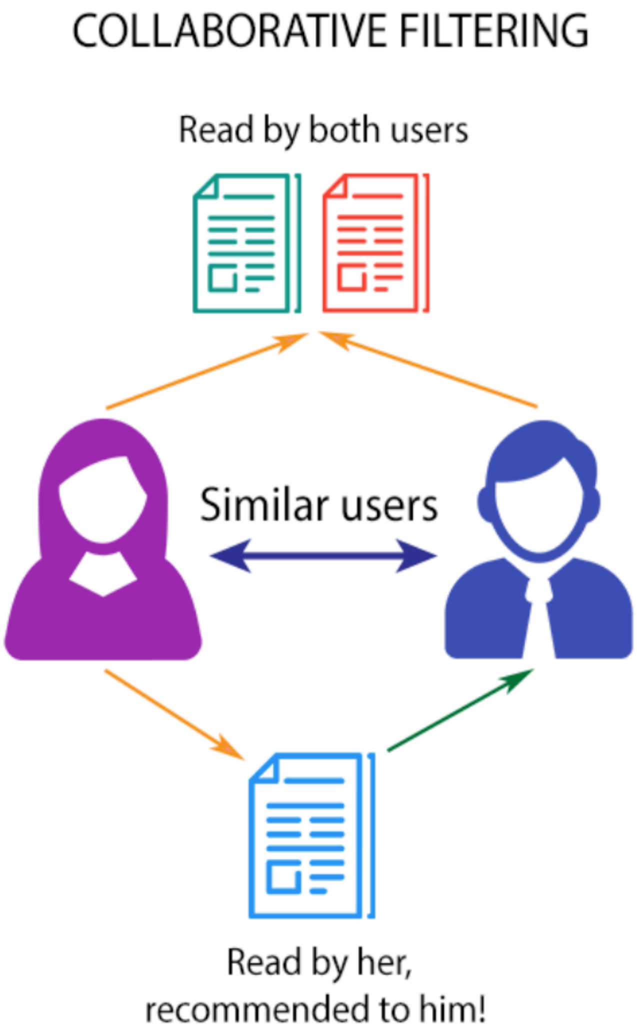
Tribe	Origins	Key Algorithm
Analogizers	Psychology	Kernel machines

From: Dr. Pedro Domingos

A little bit History

- **SVM :** first introduced in 1992, popular because of its success in handwritten digit recognition (1994); Regarded as an important example of "kernel methods"

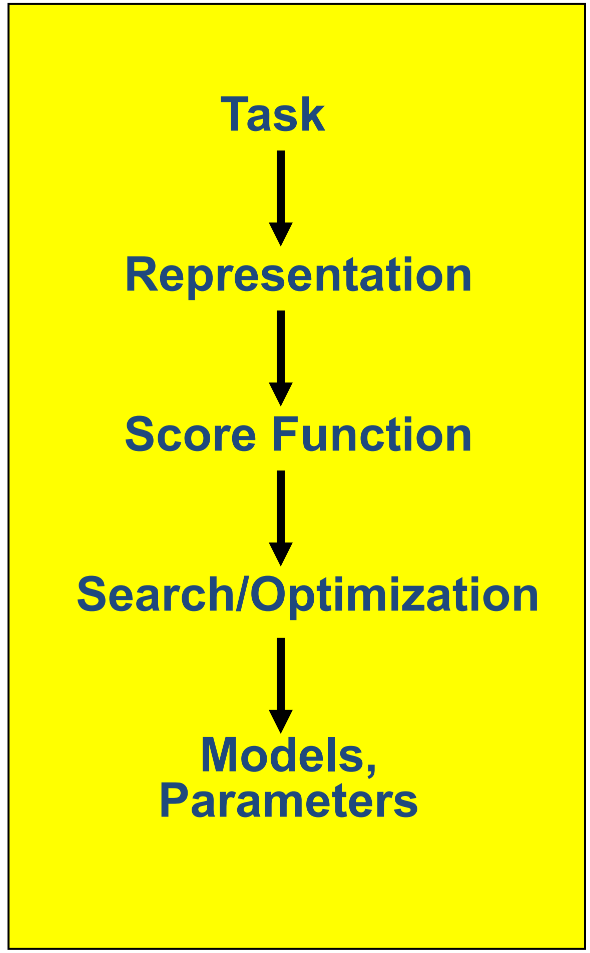
- **Recommender Systems:**
  - E.g., Matrix Factorization



Recommender Systems

From: Dr. Pedro Domingos

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

The Big Picture

Tribe	Problem	Origins	Solution	Module in Nutshell
Symbolists	Knowledge composition	Logic, philosophy	Inverse deduction	Representations;
Connectionists	Credit assignment	Neuroscience	Backpropagation	Representations; Optimization
Evolutionaries	Structure discovery	Evolutionary biology	Genetic programming	Optimization;
Bayesians	Uncertainty	Statistics	Probabilistic inference	Score function;
Analogizers	Similarity	Psychology	Kernel machines	Representations;

Your UVA Email ID:
Your Name:

Q12: Summarize what we have covered

Task	
Representation	
Score Function	
Search/Optimization	
Models, Parameters	

Final Review

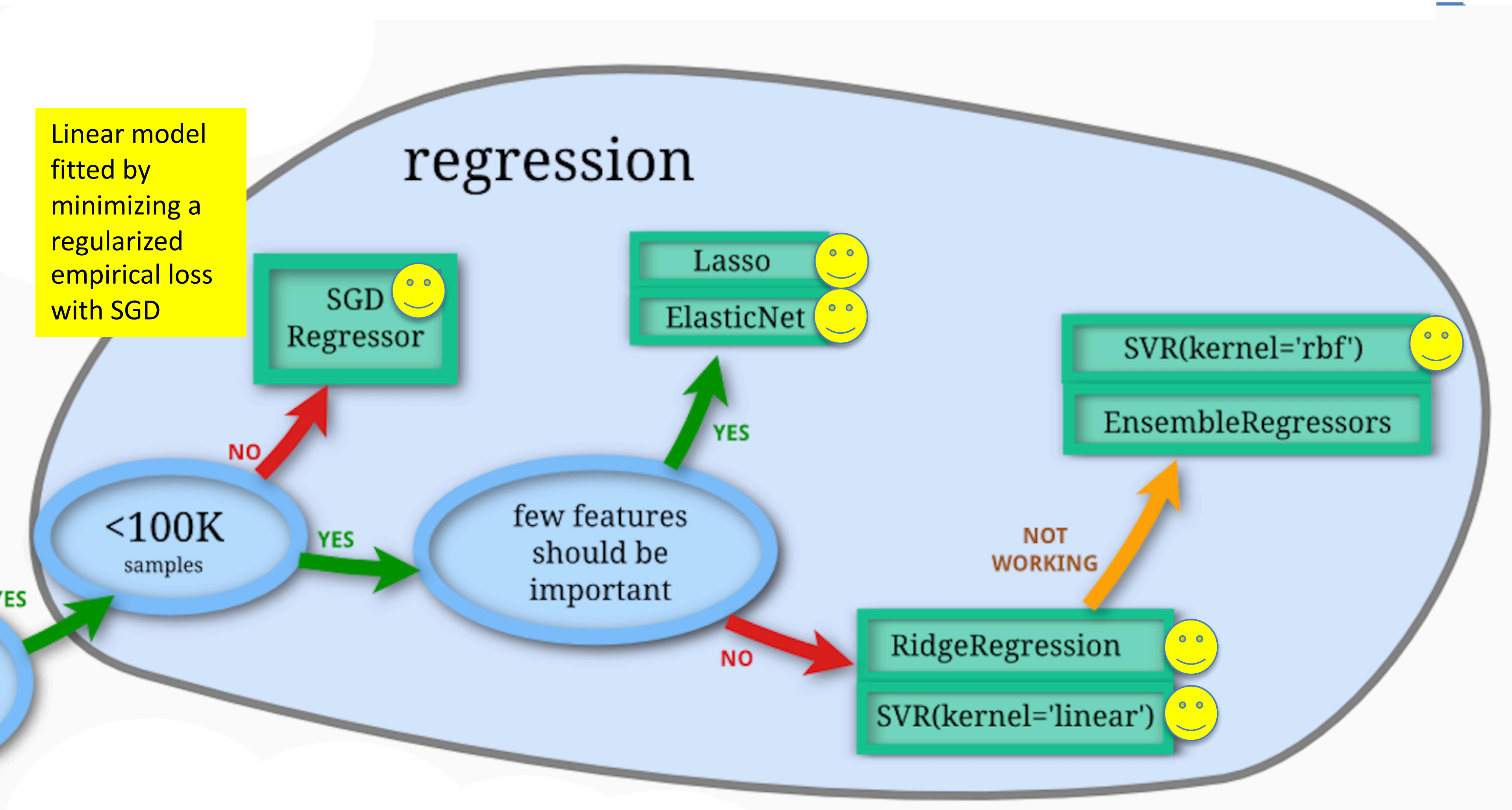
- Five Tribes of Machine Learning
- Review of ML methods covered so far
  - Regression (supervised)
  - Classification (supervised)
  - Unsupervised models
  - Learning theory
  - Review of Six Assignments
- Four books to recommend

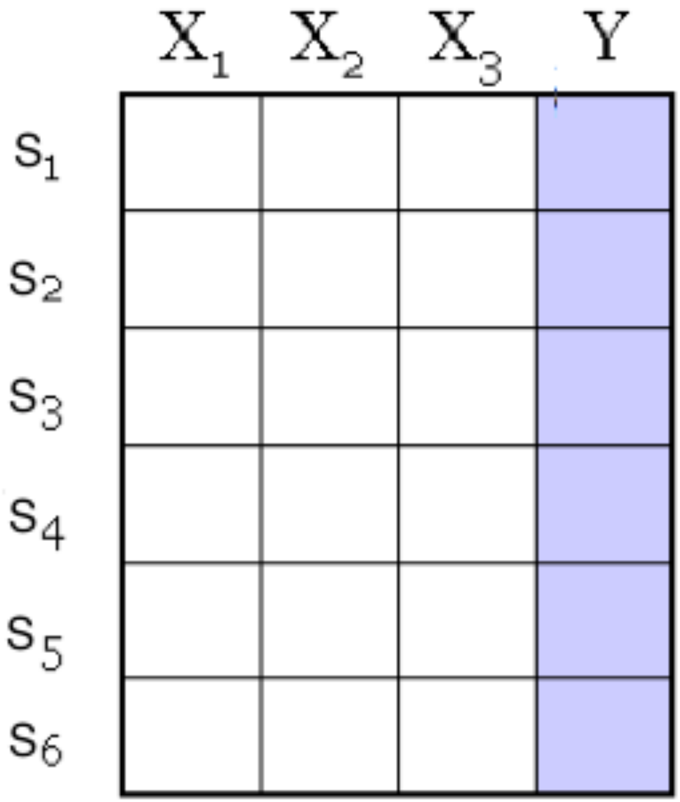
What we have covered (I)

- Supervised Regression models
  - Linear regression (LR)
  - LR with non-linear basis functions
  - LR with Regularizations
  - Feature selection *

Scikit-learn : Regression

Linear model fitted by minimizing a regularized empirical loss with SGD





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]

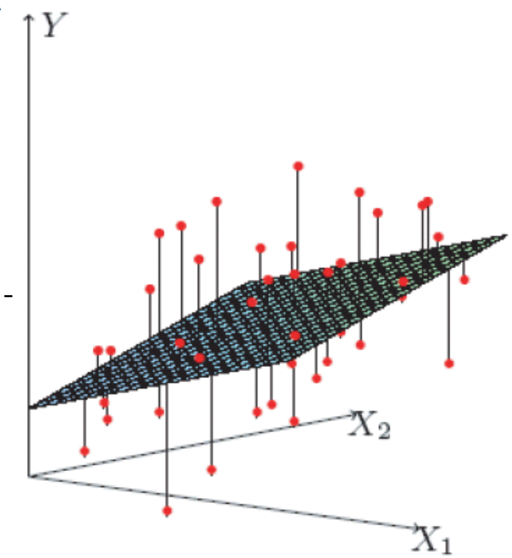
(1) Multivariate Linear Regression

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Regression → Y = Weighted linear sum of X' s → Least-squares → Linear algebra → Regression coefficients

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

$$\rightarrow \theta$$



(2) Multivariate Linear Regression with basis Expansion

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Regression → Y = Weighted linear sum of (X basis expansion) → Least-squares → Linear algebra → Regression coefficients

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

(3) Regularized multivariate linear regression

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Regression → Y = Weighted linear sum of X's → Least-squares+ Regularization -norm → Normal Eq / SGD / GD → Regularized Regression coefficients

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

(4) Feature Selection

Task

Representation

Score Function

Search/Optimization

Models, Parameters

Dimension Reduction

n/a

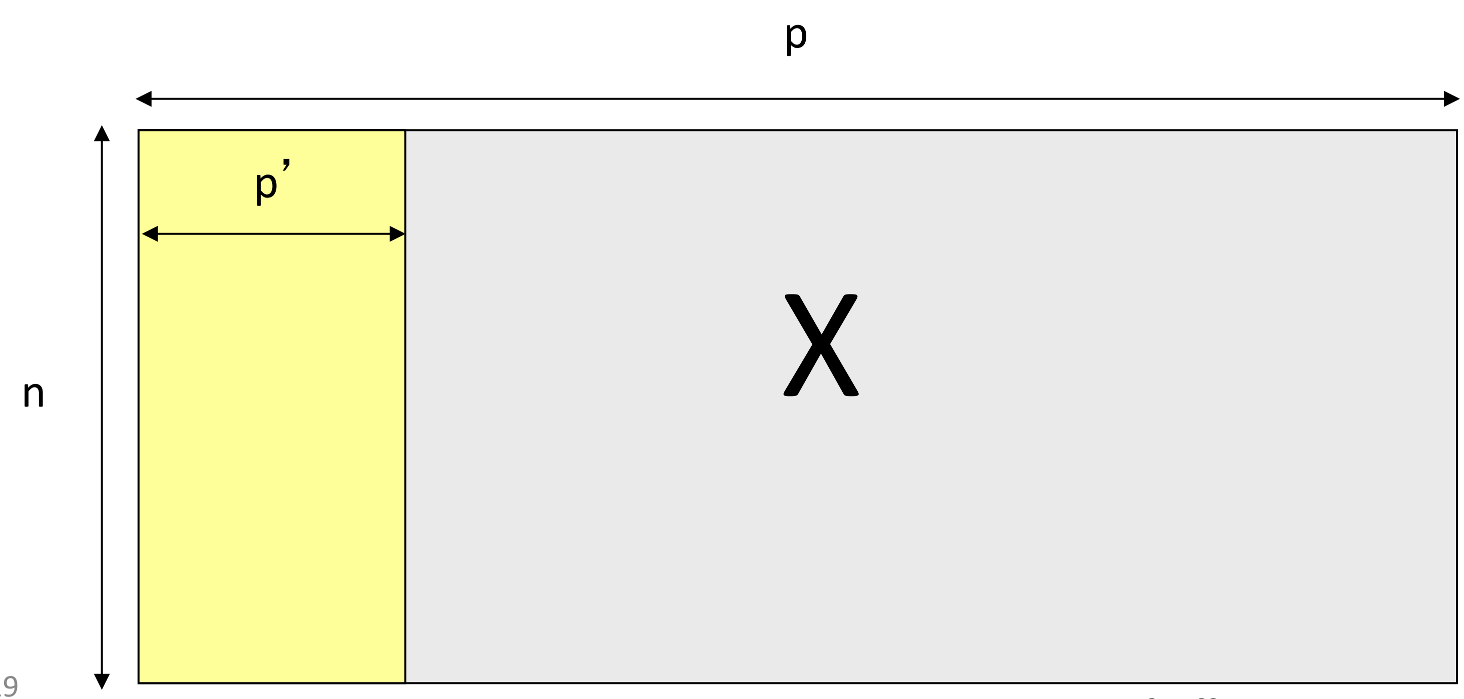
Many possible options

Greedy (mostly)

Reduced subset of features

(4) Feature Selection

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

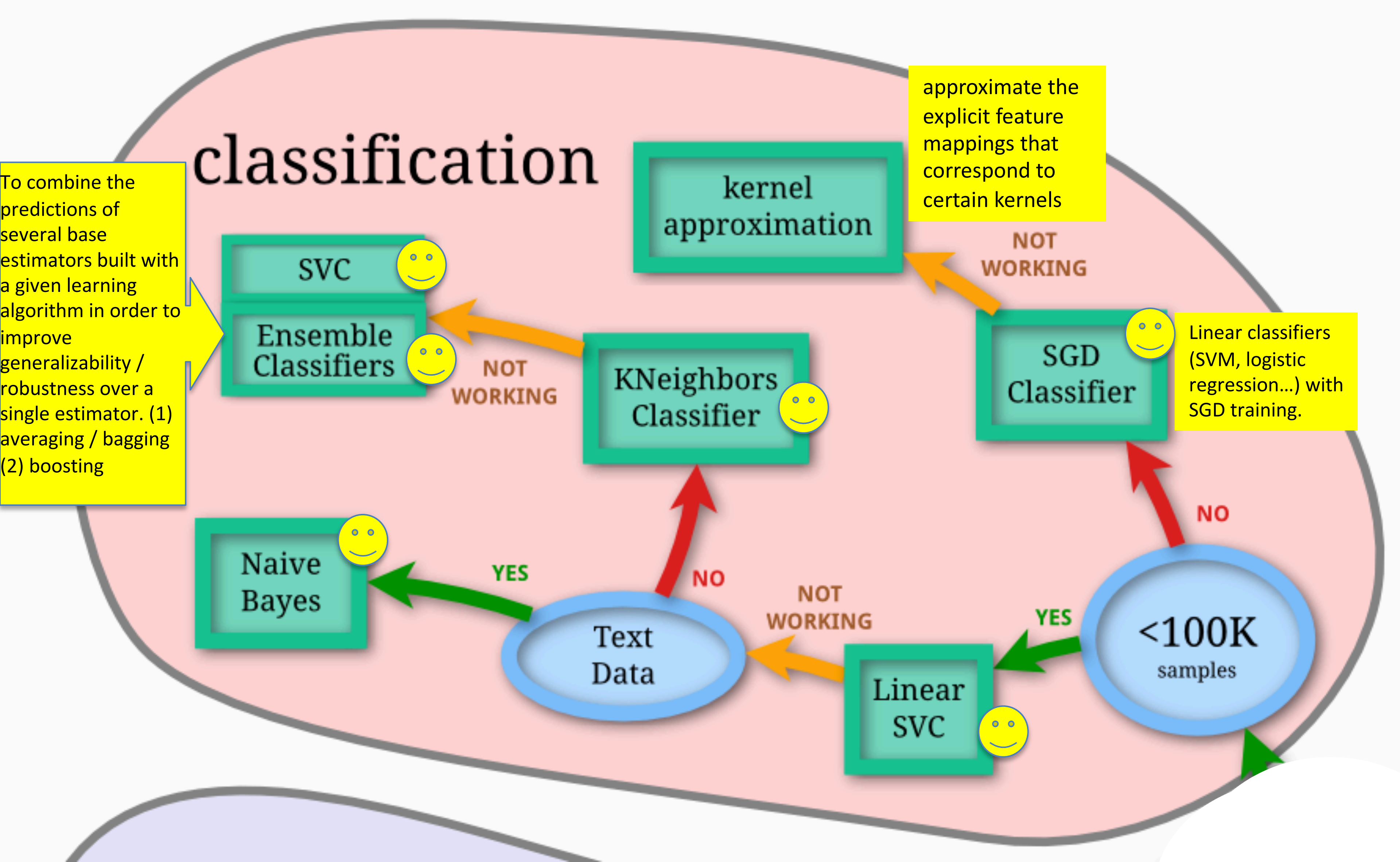


From Dr. **Isabelle Guyon**

Final Review

- ❑ Five Tribes of Machine Learning
- ❑ Review of ML methods covered so far
  - ❑ Regression (supervised)
  - ❑ Classification (supervised)
  - ❑ Unsupervised models
  - ❑ Learning theory
  - ❑ Review of Six Assignments
- ❑ Four books to recommend

Scikit-learn : Classification



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

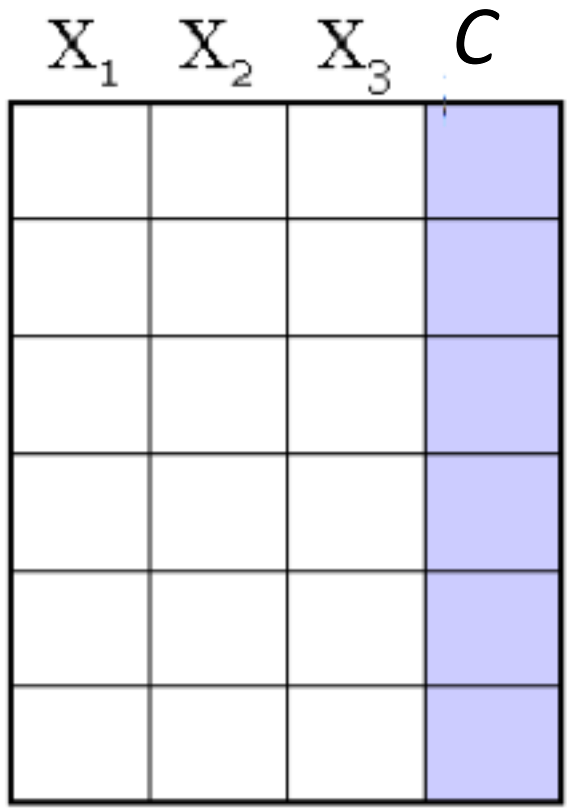
Three major sections for classification

- 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, neural networks
   - e.g., support vector machine, decisionTrees

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

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

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

- **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

Task	classification
Representation	Local Smoothness
Score Function	EPE with L2 loss ➔ conditional mean (Extra)
Search/Optimization	NA
Models, Parameters	Training Samples

(2) Support Vector Machine

Task

↓

Representation

↓

Score Function

↓

Search/Optimization

↓

Models, Parameters

classification

↓

$$K(\mathbf{x}, z) := \Phi(\mathbf{x})^T \Phi(z)$$

Kernel Func K(xi, xj)

↓

Margin + Hinge Loss

↓

QP with Dual form

↓

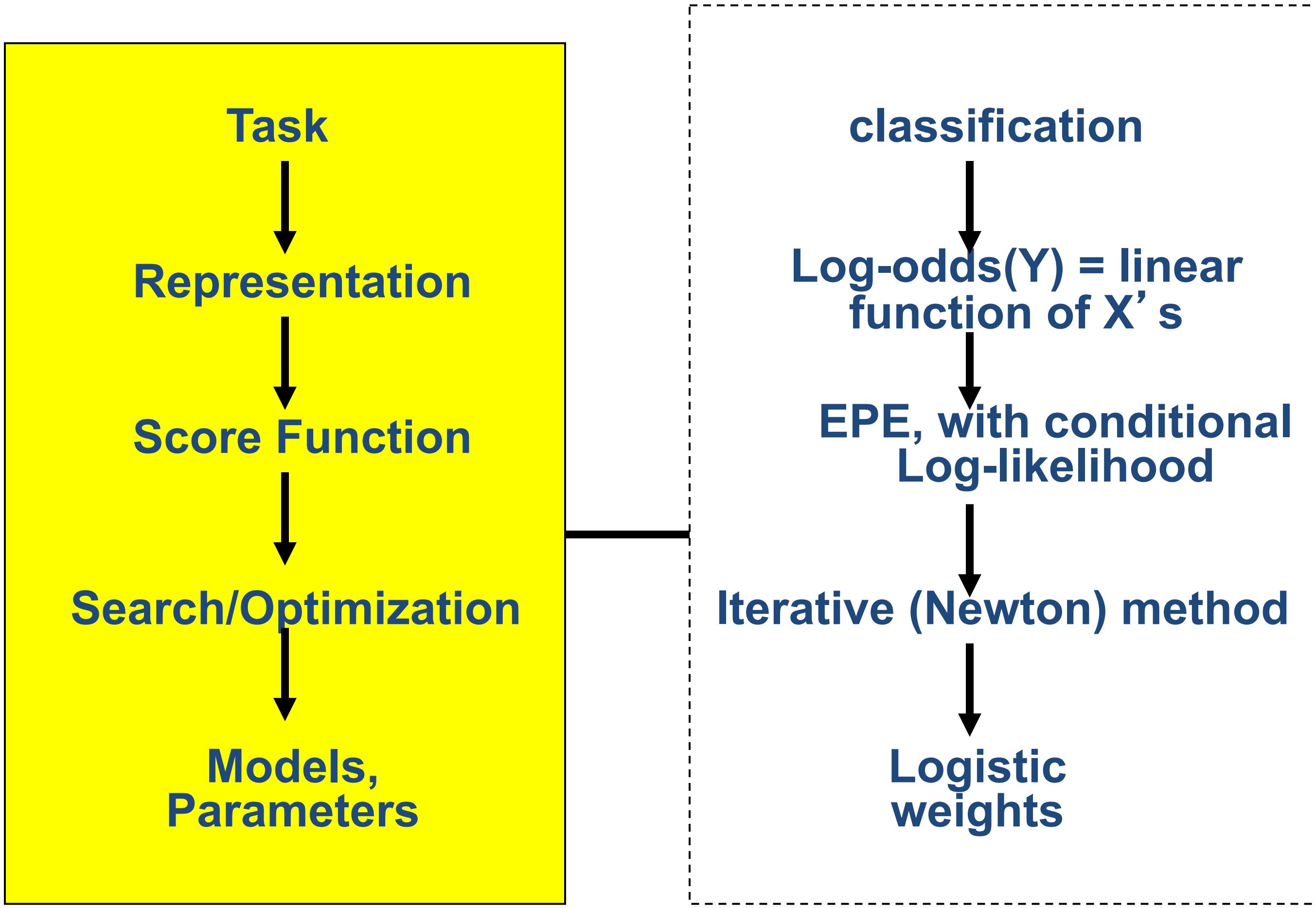
Dual Weights

$$w = \sum_i \alpha_i x_i y_i$$

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

$$\text{subject to } \forall \mathbf{x}_i \in Dtrain : y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) \geq 1 - \varepsilon_{\mathrm{i}}$$

(3) Logistic Regression



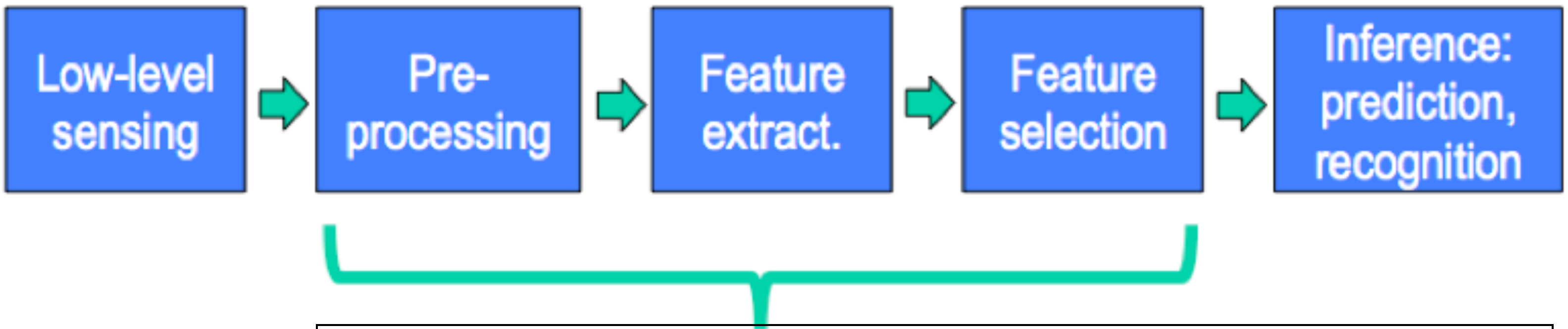
$$P(c=1|x)=\frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

(4) Neural Network

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Classification / Regression → Multilayer Network topology → conditional Log-likelihood , Cross-Entropy / MSE → SGD / Backprop → NN network Weights

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

Some Recent DNN Trends

- 1. CNN / Residual / Dynamic parameter
- 2. RNN / Attention / Seq2Seq / BERT …
- 3. Neural Architecture with explicit Memory
- 4. Learning to optimize / Learning DNN architectures
- 5. Autoencoder / layer-wise training
- 6. Learning to learn / meta-learning/ few-shots
- 7. DNN on graphs / trees / sets
- 8. NTM 4program induction / sequential decisions
- 9. Generative Adversarial Networks (GAN)
- 10. Deep Generative models, e.g., autoregressive
- 11. Deep reinforcement learning
- 12. Validate / Evade / Test / Understand / Verify DNNs

New Network topology

New Losses

New Inputs

New Tasks

New Model Properties

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

(5) Generative Bayes Classifier

Task → Representation → Score Function → Search/Optimization → Models, Parameters

classification → Prob. models p(X|C) → EPE with 0-1 loss, with Log likelihood(optional) → MLE → Prob. Models' Parameter

$$P(X_1, \cdots, X_p \mid C)$$

Gaussian Naive

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

Bernoulli Naïve

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

Multinomial

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

(6) Decision Tree / Random Forest

Task	Classification
Representation	Partition feature space into set of rectangles, local smoothness
Score Function	Split Purity measure / e.g. IG / cross-entropy / Gini /
Search/Optimization	Greedy to find partitions
Models, Parameters	Tree Model (s), i.e. space partition

Random Forest

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Classification / Regression
↓
multiple (almost) full decision trees / bootstrap samples / sample features
↓
Split with Purity measure / e.g. IG / cross-entropy / Gini /
↓
Greedy like Decision Tree (e.g. GINI)
↓
Multiple Tree Model (s), i.e. space partition

Boosting

Task

Representation

Score Function

Search/Optimization

Models, Parameters

Classification / Regression

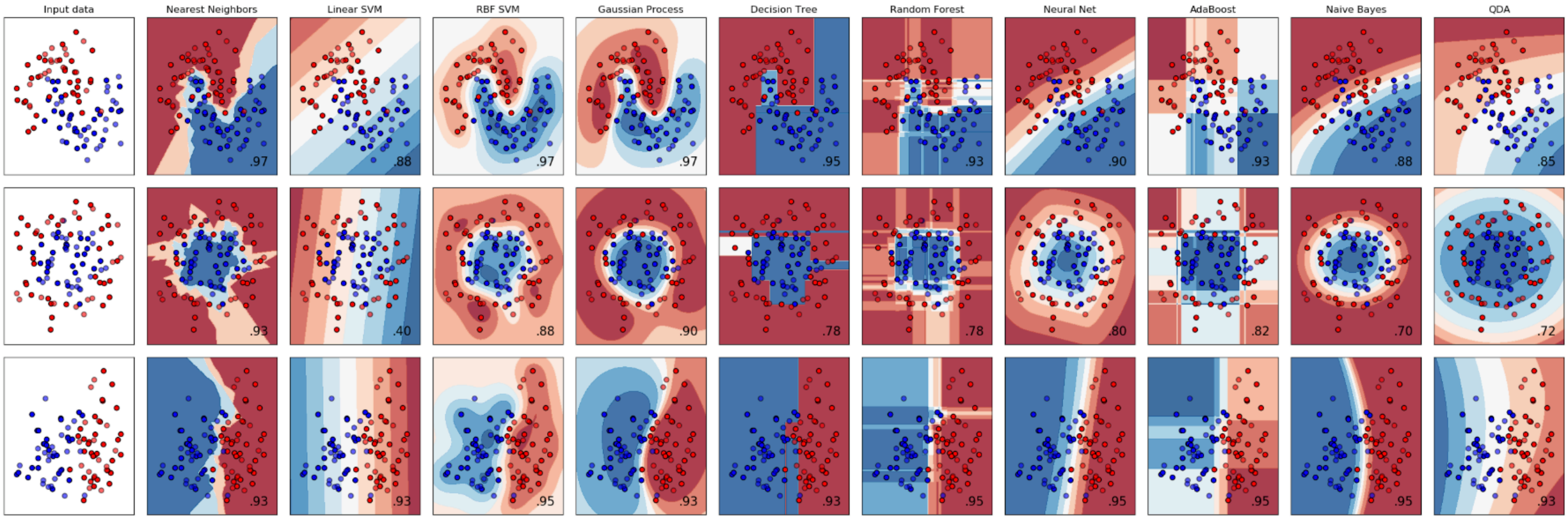
Weighted sum of a series of shallow decision trees

Exponential Loss

Gradient based search of weights / DT split

Multiple Trees + Tree Weights

https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html



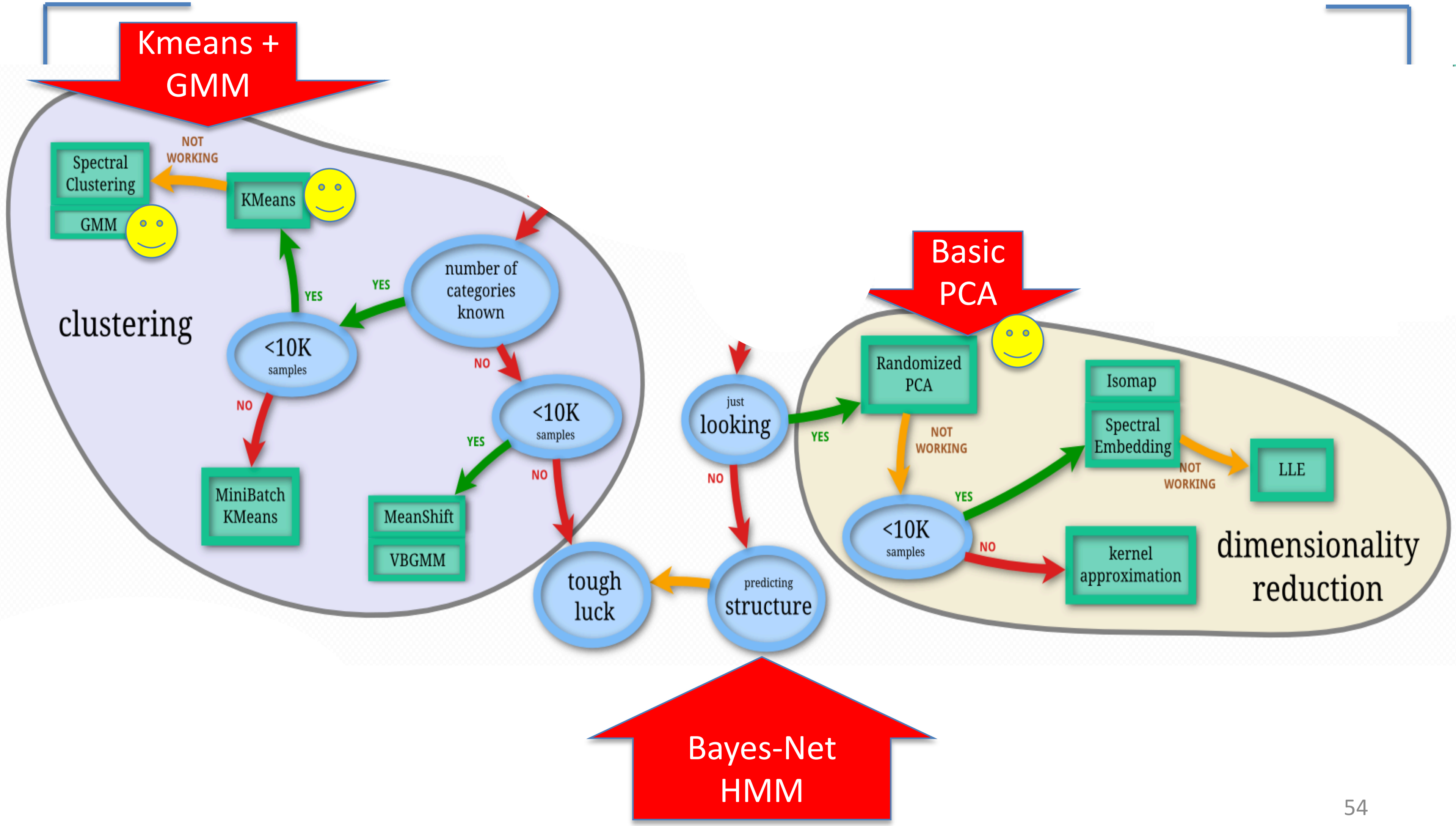
- ✓ different assumptions on data
- ✓ different scalability profiles at training time
- ✓ different latencies at prediction (test) time
- ✓ different model sizes (embedability in mobile devices)
- ✓ different level of model interpretability

Adapt from Olivier Grisel's talk

Final Review

- Five Tribes of Machine Learning
- Review of ML methods covered so far
  - Regression (supervised)
  - Classification (supervised)
  - Unsupervised models
  - Learning theory
  - Review of Six Assignments
- Four books to recommend

Then after classification



What we have covered (III)

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

An unlabeled Dataset X

	X_1	X_2	X_3
s_1			
s_2			
s_3			
s_4			
s_5			
s_6			

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

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)

Task

↓

Representation

↓

Score Function

↓

Search/Optimization

↓

Models, Parameters

Dimension Reduction

↓

Gaussian assumption (Implicit)

↓

Direction of maximum variance

↓

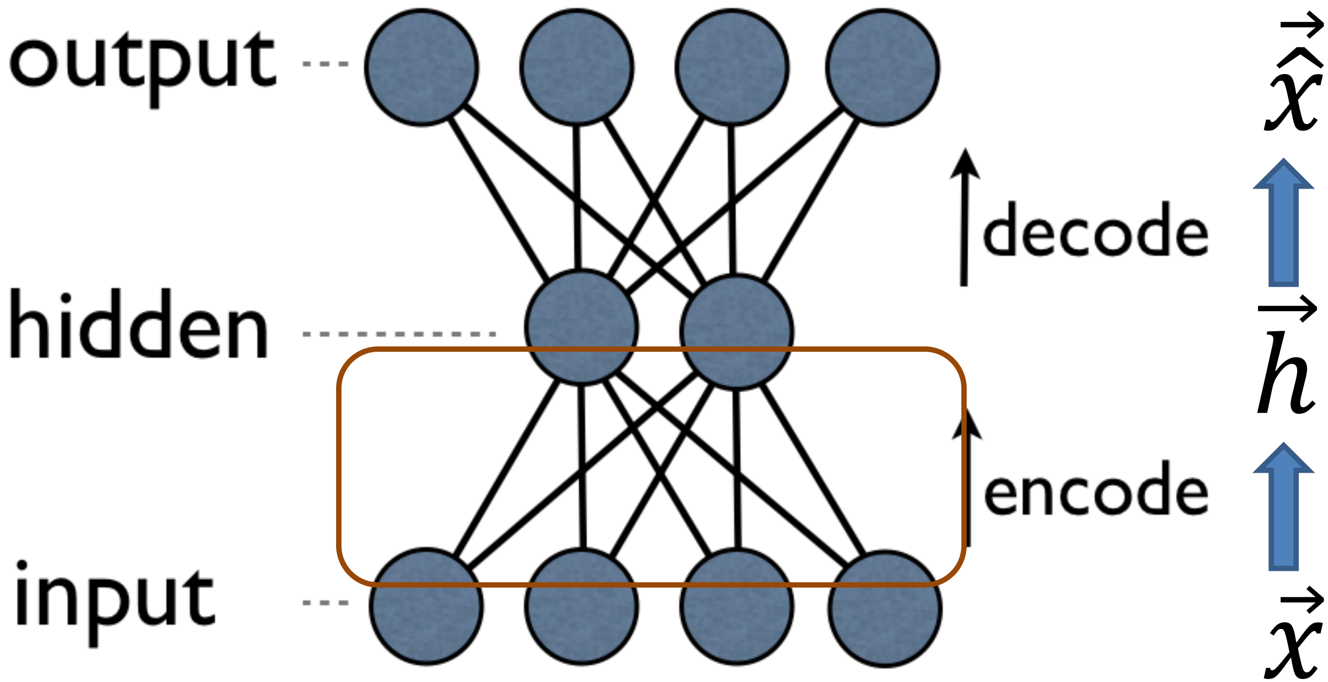
Eigen-decomp

↓

Principal components

Reconstruction Loss :

auto-encoder-decoder is trained to reproduce the input



Minimize diff

$|\vec{\hat{x}} - \vec{x}|$

Reconstruction Loss: force the 'hidden layer' units to become good / reliable feature detectors

(1) Hierarchical Clustering

Task	**Clustering**
Representation	**Distance measure**
Score Function	**No clearly defined loss**
Search/Optimization	**greedy bottom-up (or top-down)**
Models, Parameters	**Dendrogram** (tree)

(2) K-means Clustering

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Clustering → Spherical clusters → Sum-of-square distance to centroid → K-means (special case of EM) algorithm → Cluster membership & centroid

Gaussian with spherical

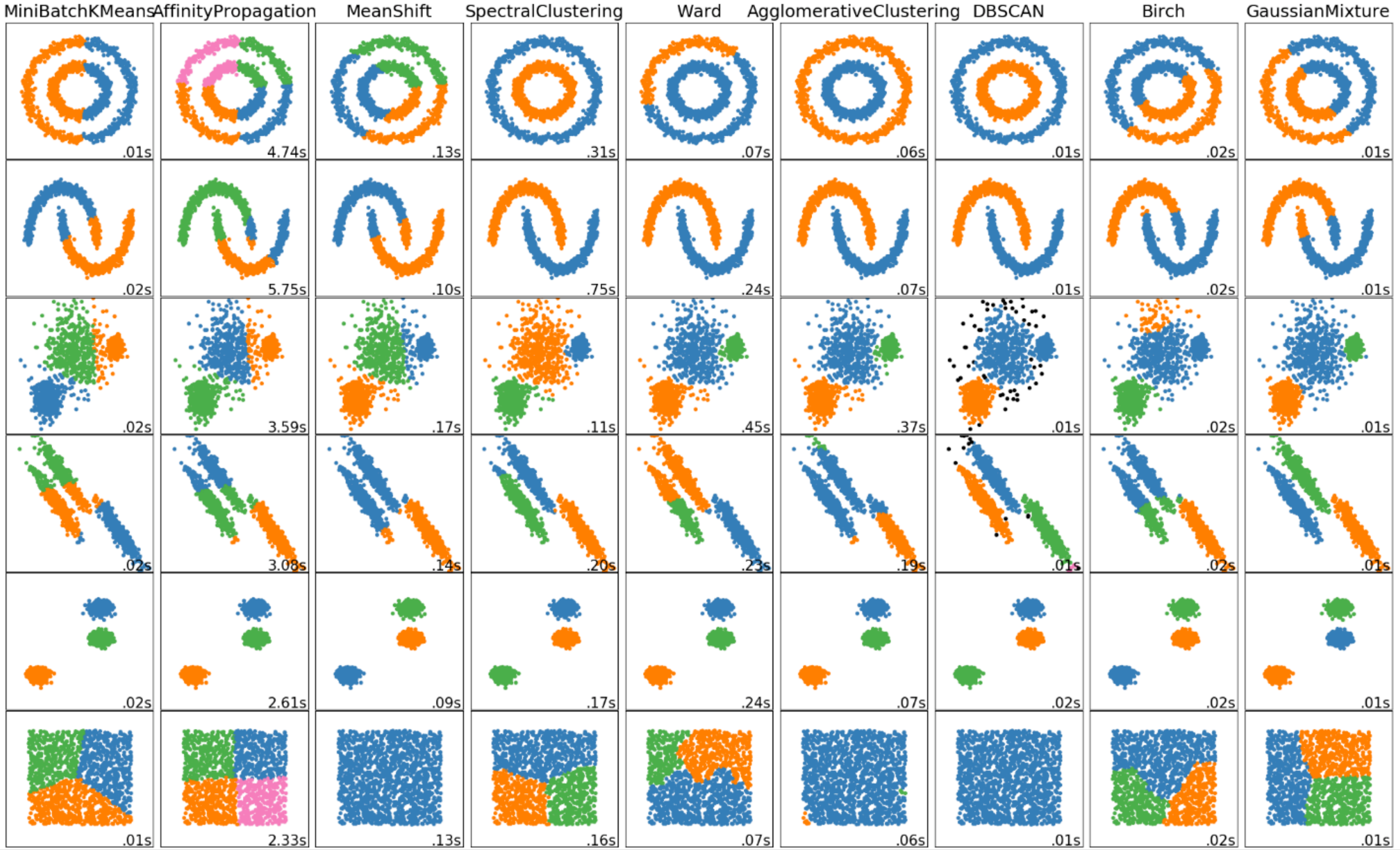
(3) GMM Clustering

Task ↓ **Representation** ↓ **Score Function** ↓ **Search/Optimization** ↓ **Models, Parameters**

Clustering ↓ **Mixture of Gaussians** ↓ **Likelihood** ↓ **EM algorithm** ↓ **Each point's soft membership & mean / covariance per cluster**

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

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html



✓different assumptions on data

✓different scalability profiles

✓different model sizes (embedability in mobile devices)

Final Review

- Five Tribes of Machine Learning

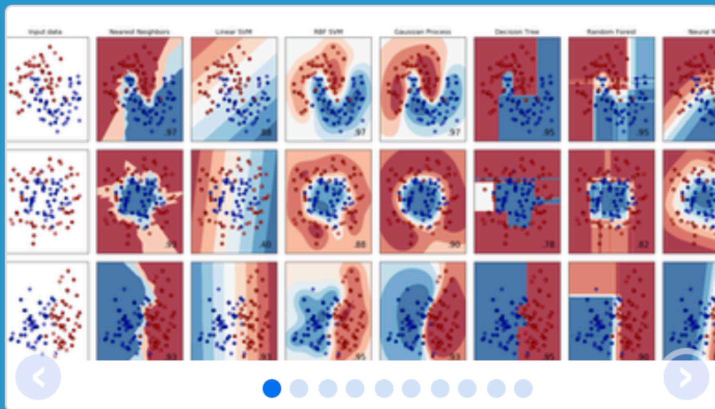
- Review of ML methods covered so far
  - Regression (supervised)
  - Classification (supervised)
  - Unsupervised models
  - Learning theory
  - Review of Six Assignments

- Four books to recommend

What we have covered (IV)

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

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

CV-based Model Selection

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 ....

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

Expected prediction error (EPE):

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

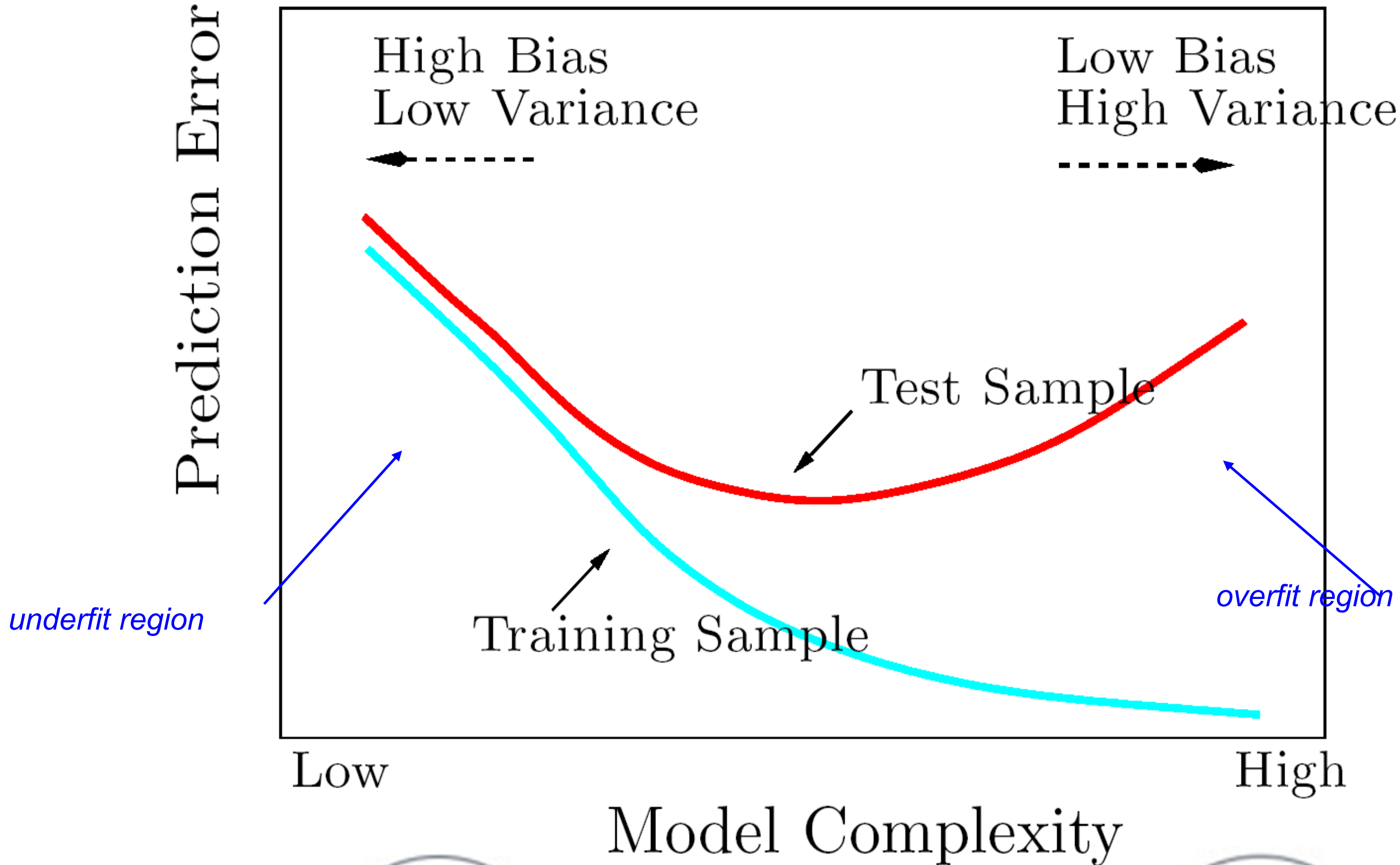
Unavoidable error

Error due to incorrect assumptions

Error due to variance of training samples

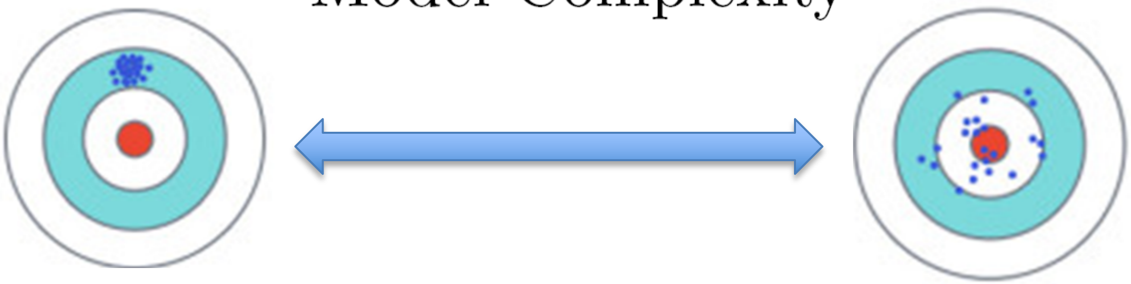
Slide credit: D. Hoiem

Recap: Bias-Variance Tradeoff / Model Selection



Large Bias

Small Variance



Small Bias

Large Variance

Logistic Regression vs. Naïve /LDA

Discriminative classifier (Logistic Regression)

- Smaller asymptotic error
- Slow convergence ~ O(p)

Generative classifier (Naive Bayes)

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

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

What we have covered for each component

Task	Regression, classification, clustering, dimen-reduction
Representation	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,
Score Function	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
Search/ Optimization	Normal equation, gradient descent, stochastic GD, Newton, Linear programming, Quadratic programming (quadratic objective with linear constraints), greedy, EM, asyn-SGD, eigenDecomp, backprop
Models, Parameters	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), …

Final Review

- ❑ Five Tribes of Machine Learning

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

- ❑ Four books to recommend

HW1

Task

↓

Representation

↓

Score Function

↓

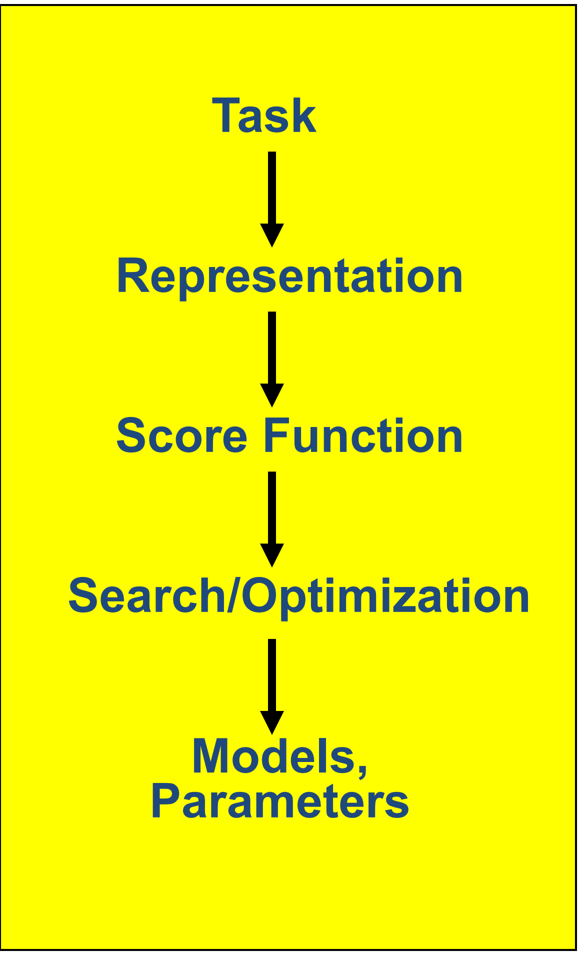
Search/Optimization

↓

Models, Parameters

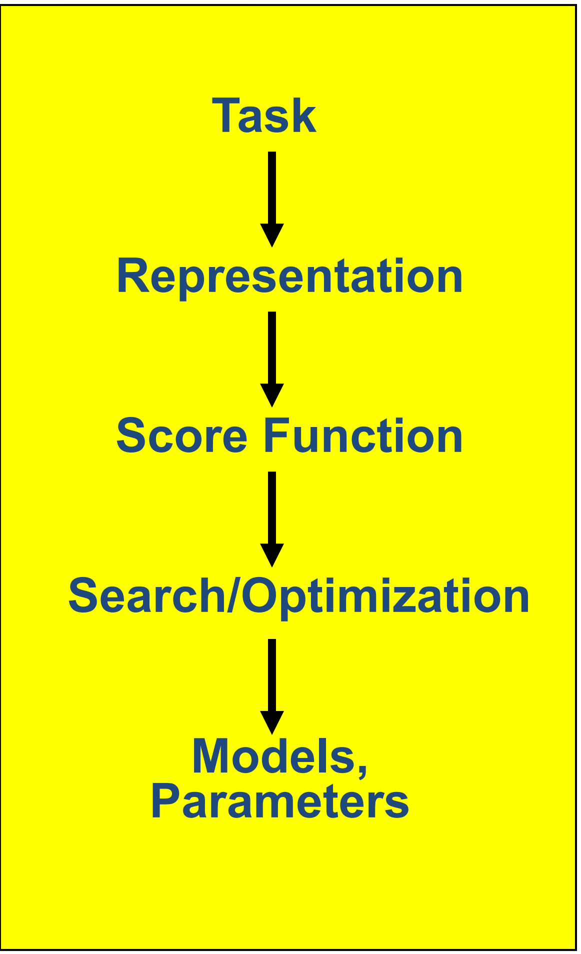
- Q1: Linear algebra review
- Q2: Linear regression model fitting
  - Data loading
  - Basic linear regression
  - Ways to train : Normal equation / SGD / (mini)-Batch GD
- Sample exam QA:
  - regression model fitting

HW2



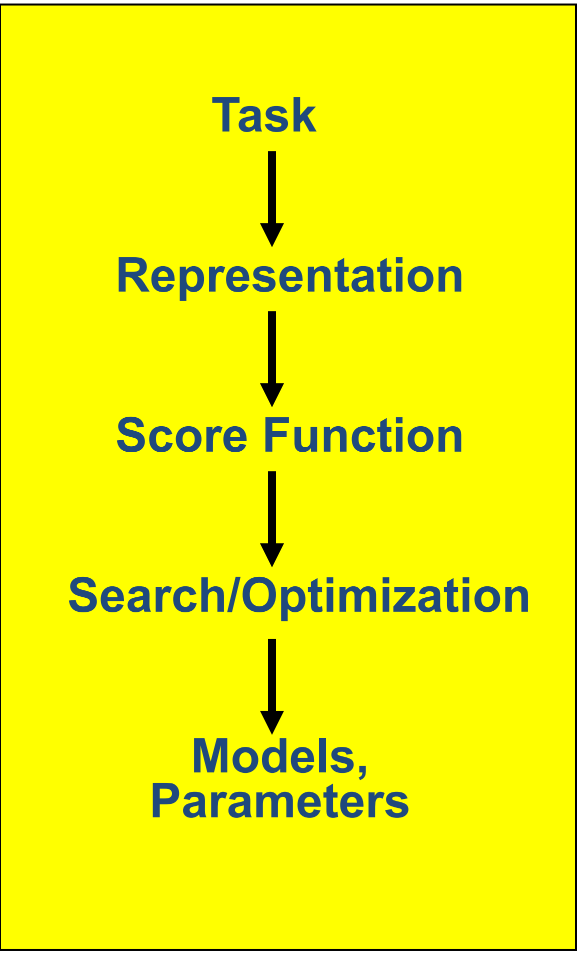
- Q1: Polynomial regression
  - Model fitting
  - Model selection of degree
  - Many sanity figures to plot
- Q2: Ridge regression
  - Math derivation of ridge
  - Understand why/how Ridge
  - Model selection of Ridge with kCV
- Sample QA:
  - Regularization

HW3



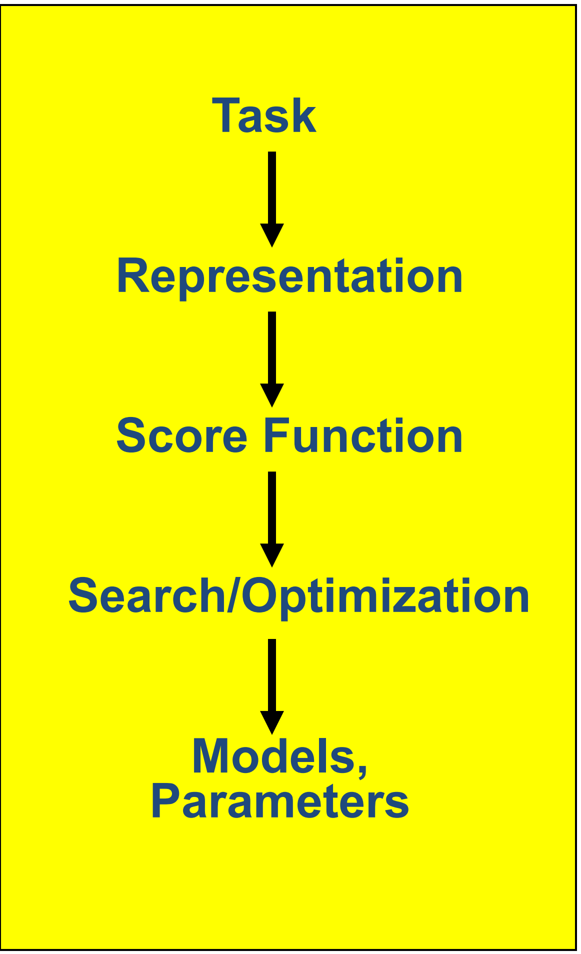
- Q1: KNN to implement and model selection of K
- Q2: Support Vector Machines with Scikit-Learn
  - Data preprocessing
  - How to use SVM package
  - Model selection for SVM
  - Model selection pipeline with train-vali, or train-CV; then test
- Sample QAs:
  - SVM

HW4



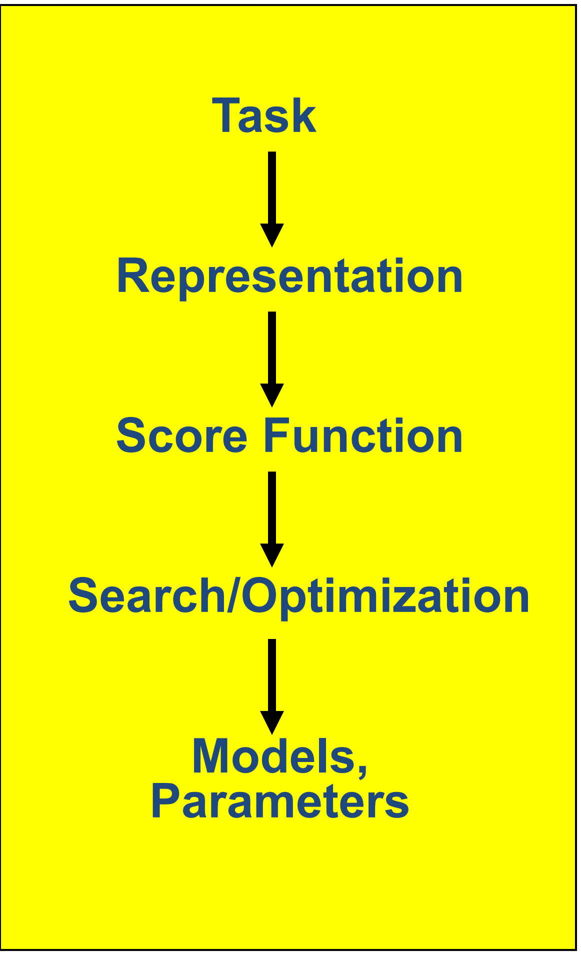
- Q1: Neural Network Tensorflow Playground
  - Interactive learning of MLP
  - Feature engineering vs.
  - Feature learning
- Q2: Image Classification /Keras
  - DNN Tool: Keras using
  - Extra: PCA for image classification
- Sample QAs:
  - Neural Nets

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 QAs:
  - Bayes Classifier

HW6



- Q1: Unsupervised Clustering of audio data and consensus data
  - Data loading
  - K-mean clustering
  - GMM clustering (several variations)
  - How to find K: knee-finding plot
  - How to measure clustering: purity
- Sample QAs:
  - Kmeans and GMM
  - Decision trees

Final Review

- Five Tribes of Machine Learning

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

- Four books to recommend

Highly Recommend One Book:
0. By 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

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
  - This book provides a fascinating exploration of how computer algorithms can be applied to our everyday lives.

Highly Recommend
Four Extra-curriculum books

- 2. 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\
  - 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

- 3. Book: Homo Deus- A Brief History of Tomorrow
  - https://www.goodreads.com/book/show/31138556-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.""

References

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