The final topic: Introduction to Machine Learning – overview

Min Zhang

z-m@tsinghua.edu.cn

Overview

- Basic concepts
- Machine learning approaches
- Selected advanced topics
- Theoretical analyses of learning
- Summary

I. Basic Concepts

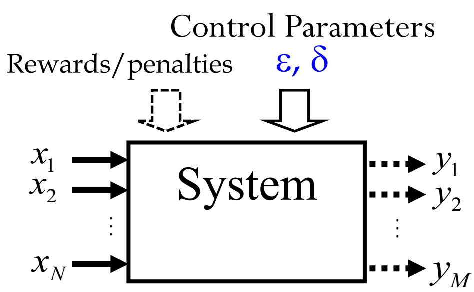
What’s machine learning？

What's machine learning?

- Learning

  = improving with experience at some task
- T （Task）
- P （Performance）
- E （Experience）

A generic system



Input Variables: $\mathbf{x} = (x_1, x_2, \ldots, x_N)$

Hidden Variables: $\mathbf{h} = (h_1, h_2, \ldots, h_K)$

Output Variables: $\mathbf{y} = (y_1, y_2, \ldots, y_K)$

II. Machine learning approaches

The branches of machine learning

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

The branches of machine learning

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

Supervised, Unsupervised and Reinforcement

	Supervised	Unsupervised	Reinforcement
Instances for learning	(X, Y) pair, usually with human involvement	X only, usually without human involvement	X *and rewards / penalties* human involved either or not

Supervised, Unsupervised and Reinforcement

	Supervised	Unsupervised	Reinforcement
Instances for learning	*(X, Y)* pair, usually with human involvement	*X* only, usually without human involvement	*X and rewards / penalties* human involved either or not
Goal of learning	Learning *relation* between *X* and *Y*	Learning *structure* of *X*	*Learning good actions to get as much reward as possible*

Supervised, Unsupervised and Reinforcement

	Supervised	Unsupervised	Reinforcement
Instances for learning	*(X, Y)* pair, usually with human involvement	*X* only, usually without human involvement	*X and rewards / penalties* human involved either or not
Goal of learning	Learning *relation* between *X* and *Y*	Learning *structure* of *X*	*Learning good actions to get as much reward as possible*
Measure of success	Loss function	No	Rewards

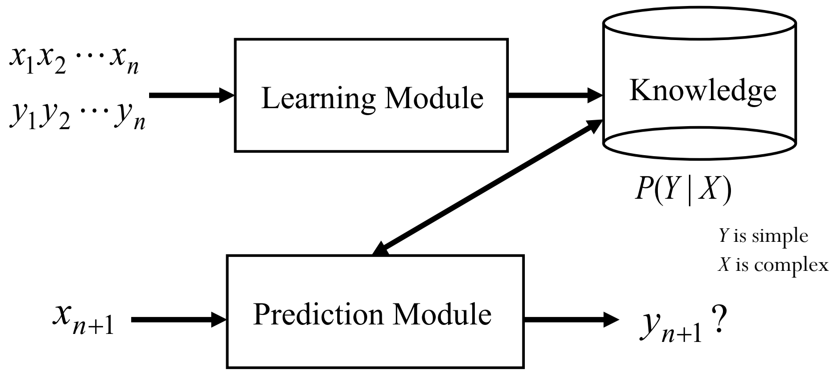
Supervised, Unsupervised and Reinforcement

	Supervised	Unsupervised	Reinforcement
Instances for learning	(X, Y) pair, usually with human involvement	X only, usually without human involvement	X *and rewards / penalties* human involved either or not
Goal of learning	Learning *relation* between X and Y	Learning *structure* of X	*Learning good actions to get as much reward as possible*
Measure of success	Loss function	No	Rewards
Application	Prediction: X=input, Y=output	Analysis: X=input	Decision-making: X=input, Y=rewards/penalties

II. Machine learning approaches (part I)

—— Supervised Learning

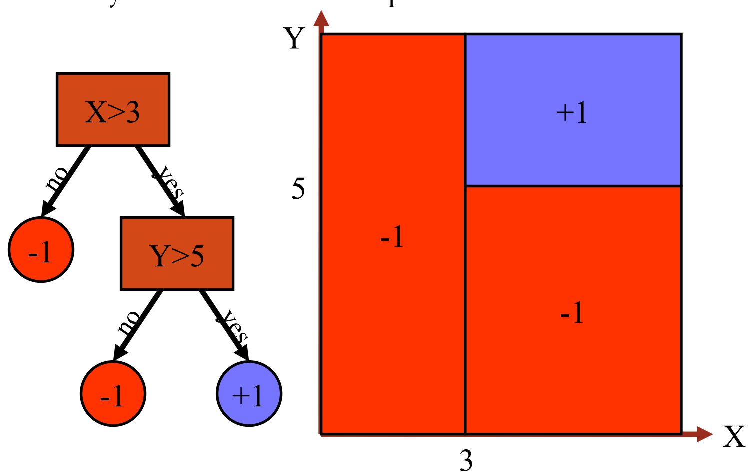
Supervised learning



15 introduction to machine learning: overview

A1. Decision tree

- Data effectively modeled via decision splits on attributes



16 introduction to machine learning: overview

- Decision tree:
  use concepts/rules to represent hypothesis
- Intuitional, easy to get the explanation of the data by the learned hypothesis
- But what if we cannot find the obvious rules for the observed data?

↓

Using statistical approaches

A2. Bayesian Learning

- Condition → Result
  - e.g. pneumonia → lung cancer?
  - Hard to tell directly
- Reversed thinking

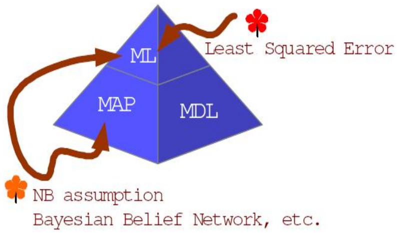
  (Result → Causal)
  - e.g. How many lung cancer patients have suffered from pneumonia?

Bayesian Learning

- Bayes theorem
  - Use prior probability to inference posterior probability

$$P(h\,|\,D)=\frac{P(D\,|\,h)P(h)}{P(D)}$$

- Max A Posterior, MAP, h_{MAP}， 极大后验假设
  - Generally we want the most probable hypothesis given the training data
- Maximum Likelihood, ML, h_{ML}, 极大似然假设
  - The smart man always learns the most from experiences if he knows $p(h)$
  - ML vs. LSE (Least Square Error)
- Naïve Bayes, NB, 朴素贝叶斯
  - Independent assumption
  - NB vs. MAP
- Minimum description length, MDL,最小描述长度
  - Tradeoff: hypothesis complexity vs. errors by h
  - MDL vs. MAP



A3. HMM

What if the observed data is indirect evidence? – hidden state exists

Property 1： Markov assumption

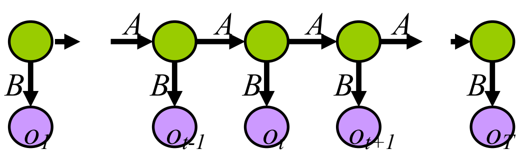
$p(X_i\,|\,X_{i-1}\ldots X_1) = p(X_i\,|\,X_{i-1})$

Property 2： time invariance assumption

$p(X_{i+1}\,|\,X_i) = p(X_{j+1}\,|\,X_j)$， for any i,j

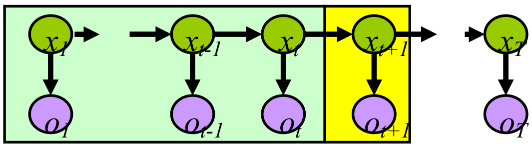
Property 3： independent observation assumption

$p(O_1,\ldots,O_T \mid X_1,\ldots,X_T) = \prod p(O_t \mid X_t)$



HMM basic Problem1 – Estimation

- Estimation problem: Compute the probability of a given observation sequence $p(\sigma \mid \mu)$
  - Define forward variable, backward variable
  - Dynamic programming – forward algorithm, $O(N^2T)$



$$\alpha_{t+1}(j) = \sum_{i=1\ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

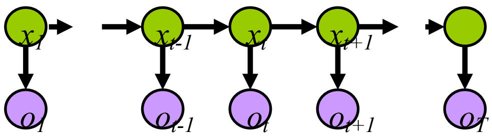
$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$

21

introduction to machine learning: overview

HMM basic Problem2 – Decoding

- Given an observation sequence, compute the most likely hidden state sequence
- Dynamic programming – Viterbi algorithm, $O(N^2T)$



$$\delta_t(j) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

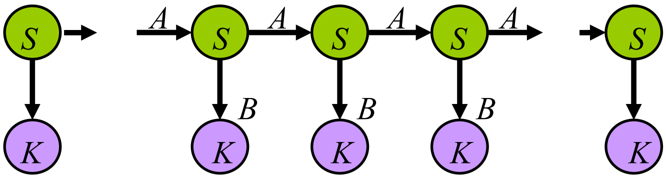
$$\boxed{\begin{aligned} \delta_{t+1}(j) &= \max_i (\delta_t(i) a_{ij} b_{jo_{t+1}}) \\ \psi_{t+1}(j) &= \arg\max_i (\delta_t(i) a_{ij} b_{jo_{t+1}}) \end{aligned}}$$

22

introduction to machine learning: overview

What we've learned from HMM

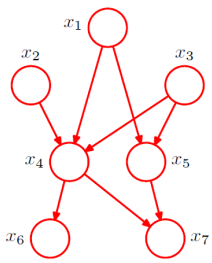
HMM



We use the special structure of this model to do a lot of neat math and solve problems that are otherwise not solvable.

A4. Probabilistic graphical models

- Bayesian networks
  - Generative model
  - Conditional independence and D-separation
- Markov random fields
  - Conditional independence and graph separation
  - Joint distribution factorization

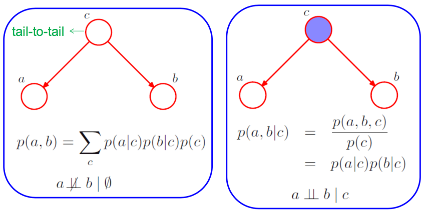


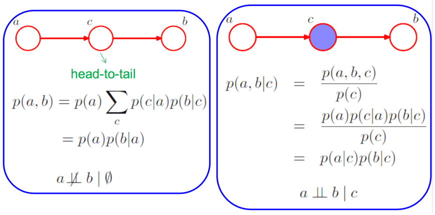
$$p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5)$$

Three basic graphs:

I. Tail-to-tail →

- If c is observed, the path is blocked.





← II. Head-to-tail

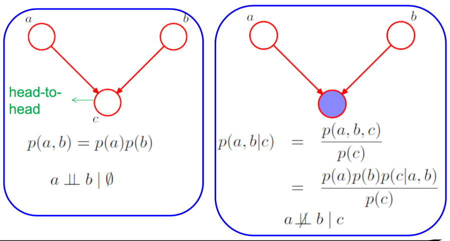
- If c is observed, the path is blocked.

$p(a,b,c) = p(a)p(b)p(c|a,b)$

If *c* is observed, the path is unblocked!

III. Head-to-head →

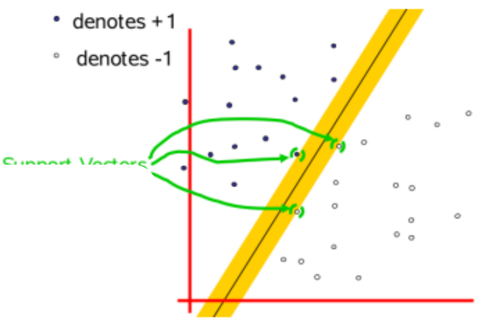
- If c is observed, the path is unblocked.
- "Explaining away"

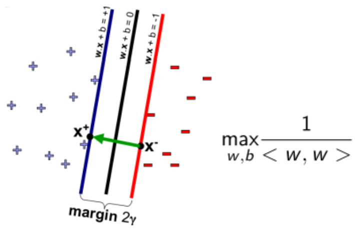


A5. Kernel methods and Non-linear SVM

Maximize the margin

- Define: the margin of a linear classifier as the width that the boundary could be increased by before hitting a data point
- Maximize the margin





$$\max_{w,b} \frac{1}{< w, w >}$$

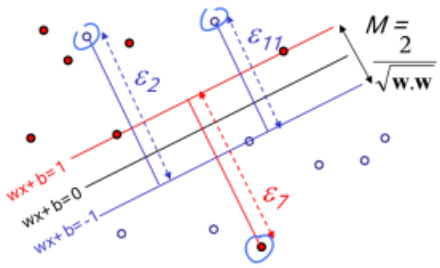
Separable case

$$\min_{w,b} \frac{1}{2} < w, w >$$

$$s.t. y_i (< w, x_i > + b) \geq 1$$

Non-separable Case

- Minimizing training error



Non-separable Case

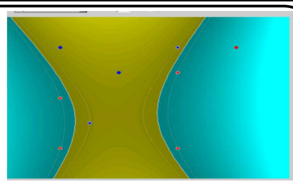
$$\min_{w,b} \frac{1}{2} < w, w > + C \sum_i \varepsilon_i$$

$$s.t. \ (< w, x_i > + b)\, y_i \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

27 introduction to machine learning: overview

Non-linear SVM



- Input space → feature space $\Phi(x) : R^n \mapsto F$
- Non-linear in low dim. → linear Hyperplane in higher dim.
- common kernels
  - Polynomials of degree d $K(x, y) = (< x, y >)^d$
  - Polynomials of degree up to d $K(x, y) = (< x, y > + 1)^d$
  - Gauss Kernel $K(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$
  - Sigmod Kernel $K(x, y) = \tanh(\eta < x, y > + v)$
- Software
  - LIBSVM http://www.csie.ntu.edu.tw/~cjlin/libsvm
  - SVMlight http://svmlight.joachims.org

28 introduction to machine learning: overview

- Previous learning approaches
  - Estimate the characteristics (e.g. distribution)
  - Make an assumption to the model
  - Find the optimal parameter

↓

But sometimes, we know nothing before learning

↓

Is there a learning approach that is NOT

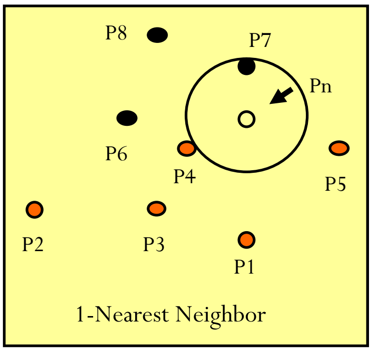
"model assumption + parameter estimation" ?

29 introduction to machine learning: overview

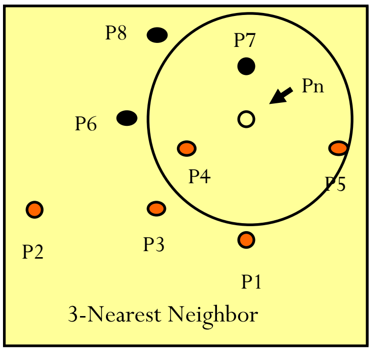
A6. k-Nearest Neighbor (KNN)

- Thinking is reminding, making analogies
- One takes the behavior of one's company

"近朱者赤， 近墨者黑"



1-Nearest Neighbor

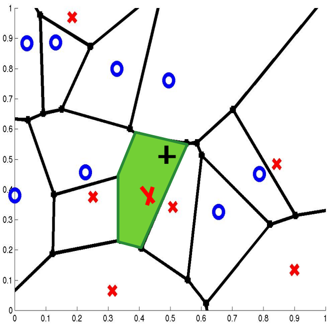


3-Nearest Neighbor

30 introduction to machine learning: overview

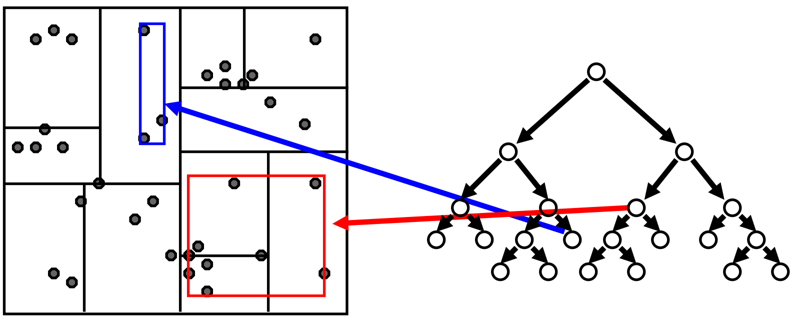
KNN

- Main assumption
  - An effective distance metric exists
- Nonparametric
- Conceptually simple, yet can model any function



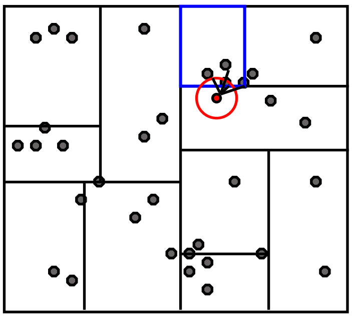
- Memory cost
- CPU cost
- Feature selection problem
  - Irrelevant features have negative impact on the distance metric
- Sensitive to representation

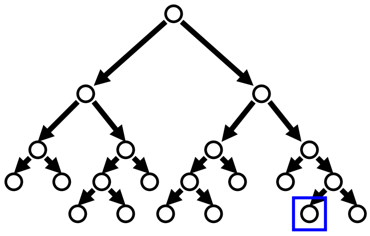
More on efficiency – KD-Tree (Construction)



We will keep around one additional piece of information at each node: **The (tight) bounds of the points at or below this node**.

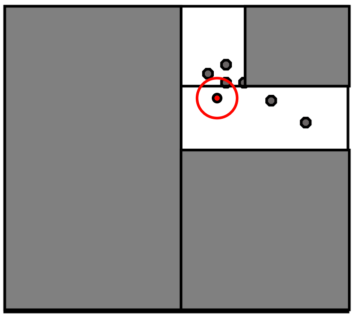
More on efficiency – KD-Tree (Query)

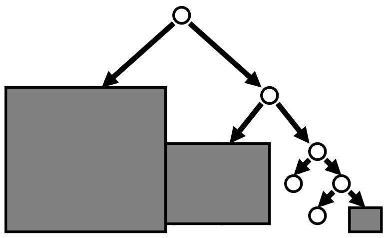




Each time a new closest node is found, we can update the distance bounds.

More on efficiency – KD-Tree (Query)

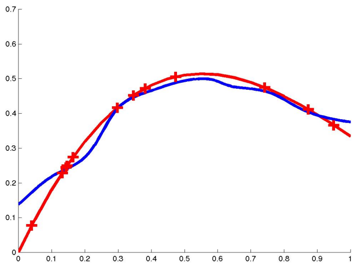




Using the distance bounds and the bounds of the data below each node, we can prune parts of the tree that could NOT include the nearest neighbor.

A memory based learner: 4 factors

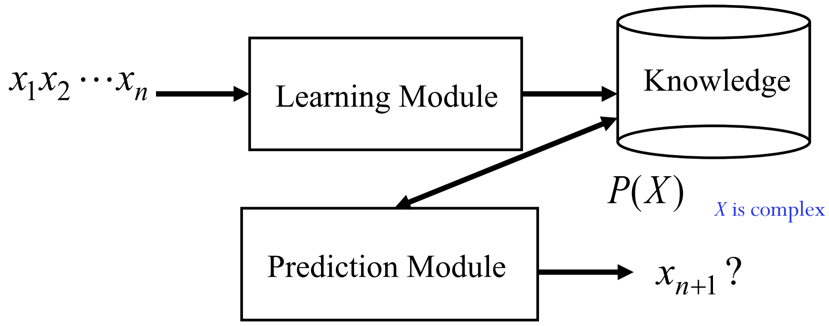
1. A distance metric
   Euclidian / Scaled Euclidian / ……
2. How many nearby neighbors to look at?
   1, k or all
3. A weighting function (optional)
   $w_i = exp(-D(x_i, query)^2 / K_w^2)$
4. How to fit with the local points?
   Nearest neighbor, or
   Voting among K neighbors, or
   The weighted average of the outputs
   $\mathbf{predict} = \Sigma w_i y_i / \Sigma w_i$



II. Machine learning approaches (part II)

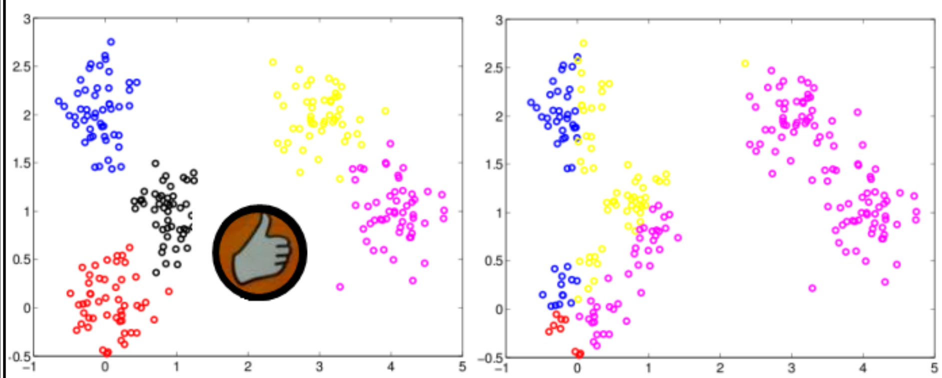
—— Unsupervised Learning

Unsupervised learning



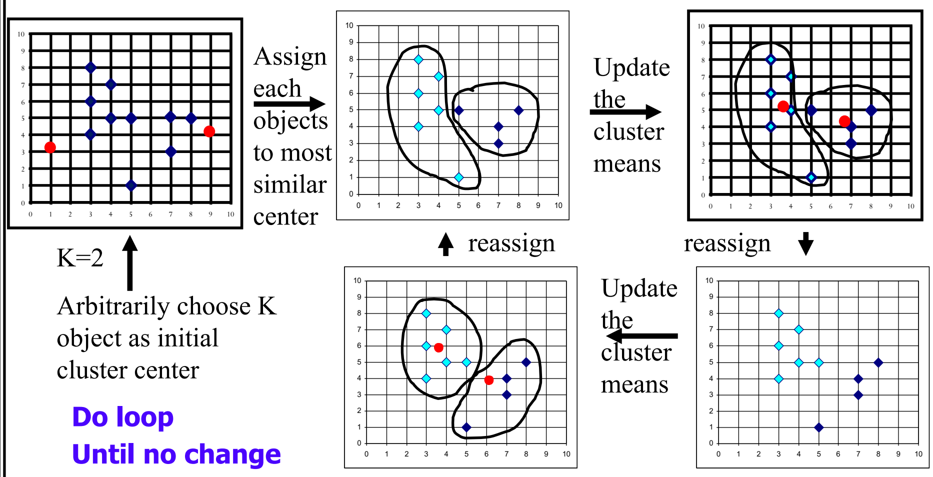
- Build a model or find useful representations of the input that can be used for decision making, predicting future inputs, efficiently communicating the inputs to another machine, etc
- Find patterns (discover the structure) in the data above and beyond what would be considered pure unstructured noise.

What are good clusters?



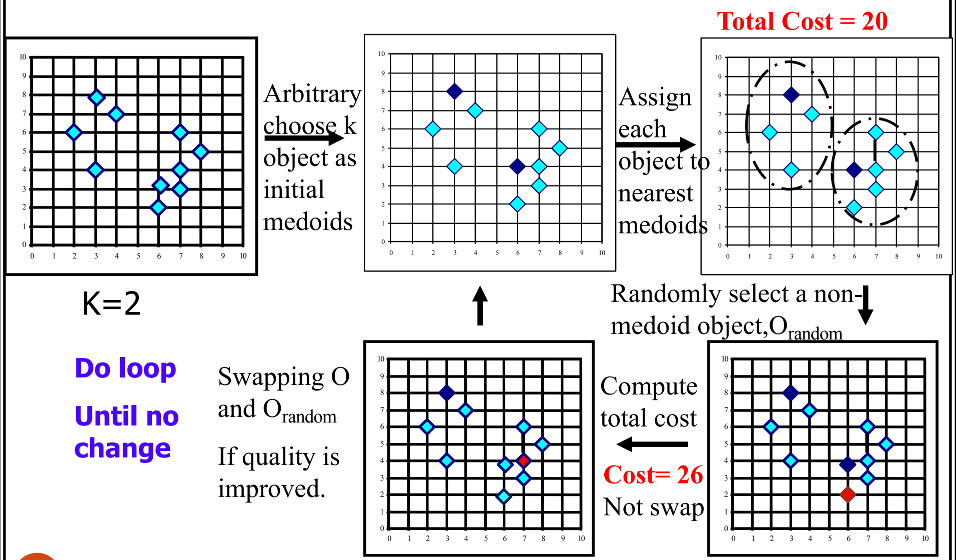
- The intra-cluster distance is small
- The inter-cluster distance is large

A7. K-Means



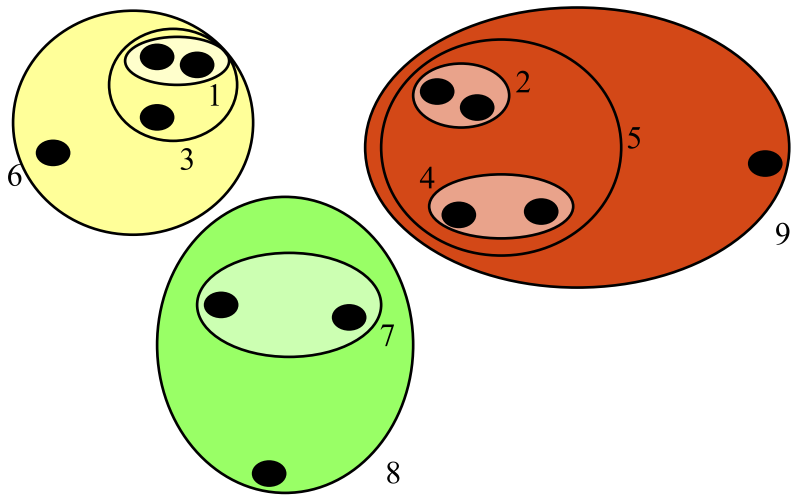
39 introduction to machine learning: overview

A8. K-Medoids



40 introduction to machine learning: overview

A9. Hierarchical Clustering (Agglomerative)

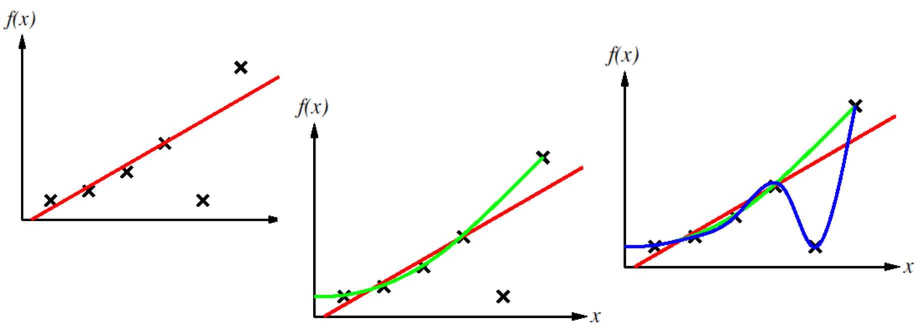


III. Selected advanced topics

—— 1. Overfitting problem

E1. Overfitting problem

- Hypothesis Space H
  - Set of hypotheses being considered



Overfitting problem

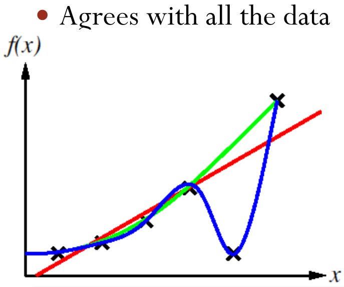
- Consistent hypothesis
  - Agrees with all the data

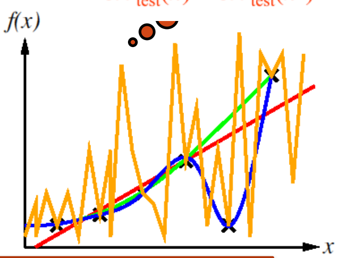
- $h \in H$ overfits training data if there's an alternative $h' \in H$ such that:

$err_{train}(h) < err_{train}(h')$

AND

$err_{test}(h) > err_{test}(h')$





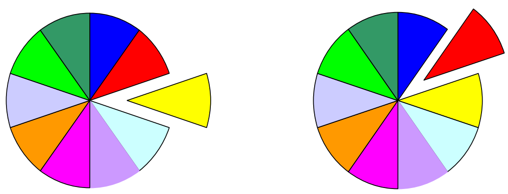
Two tips:
- Generalization ability
- Compare two algorithms on the same dataset!
  (If you use different datasets: diff. data → diff. performance)

III. Selected advanced topics

—— 2. Limited data

Learning on limited data (1) ：E2. Cross Validation

- When data is limited
  - what is the best way to use this data to both learn a hypothesis and estimate its accuracy?
- k- fold cross validation 交叉验证
  - Use average error to estimate error



46

introduction to machine learning: overview

Learning on limited data (2) : E3. Boostrap sampling

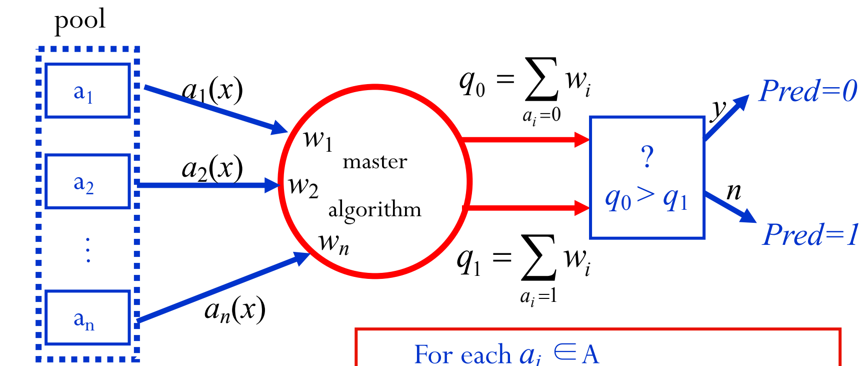
- Bootstrap sampling
  - Given a set D containing m training examples
  - Create D_i by drawn m examples uniformly at random with replacement from D (drawn with replacement, 取出放回)

III. Selected advanced topics

—— 3. Ensemble learning

"Two heads are better than one."
"三个臭皮匠，顶一个诸葛亮"

A10. Weighted majority algorithm

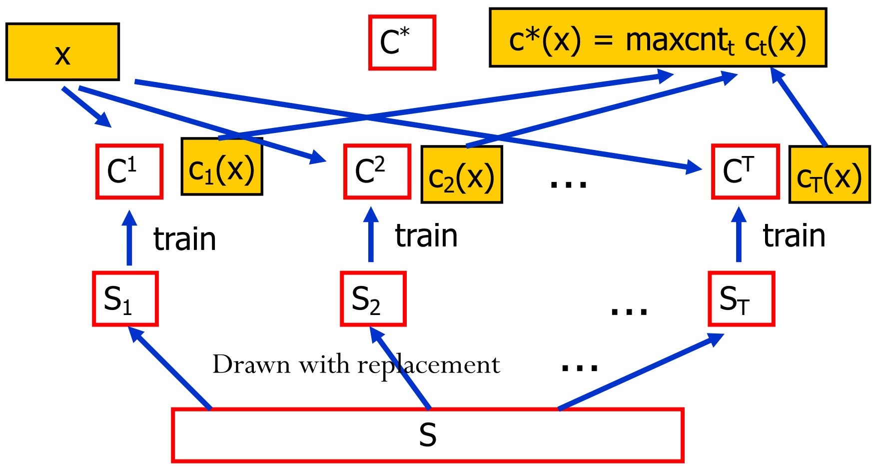


Assume: binary output $\{0,1\}$

For each $a_i \in A$
If $a_i(x) \neq c(x)$, then $w_i \leftarrow \beta w_i$
$\beta \in [0,1)$

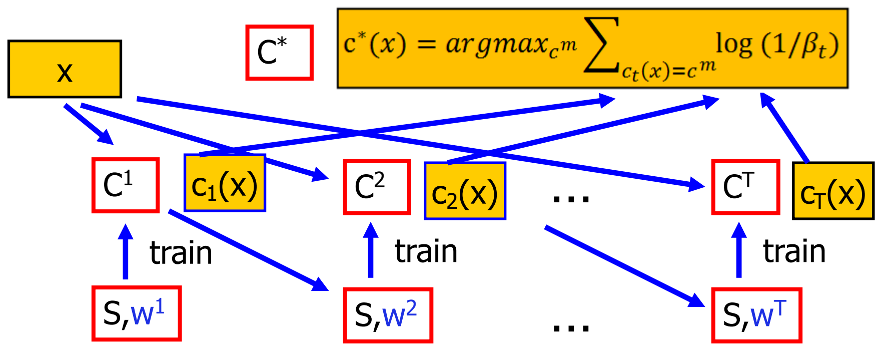
49 introduction to machine learning: overview

A11. Bagging (boostrap aggregating)



50 introduction to machine learning: overview

A12.Boosting (learn from failures)



What is a good weak learner?

The set of weak rules (features) should be:

- Unstable: small change in training set cause large change in hypothesis produced
- Simple: allow efficient search for a rule with non-trivial weighted training error. Calculation of prediction from observations should be very fast
- Small: to avoid over-fitting.

Reweighting vs. Resampling

III. Selected advanced topics

—— 4. Deep learning

A13. Deep learning: When does it help?

- With massive amounts of computational power
- With sufficient data is available
  - Generally has complicate structure
- When you don't have ideas on how to select good features
- Deep = Deep nets (the network has many layers)
- Currently little in-depth knowledge has been leveraged in learning procedure in DL approaches

What we have briefly introduced

- Multi-layer perceptron
- Convolutional neural nets

Supervised learning

- Restricted Boltzmann machine
- Deep belief network

Unsupervised learning

- Applications
- Remarks
  - Going deeper and deeper
  - Large models seems to be critical -- Parallel computing
  - Need more theoretical foundations
  - Unsupervised learning deserves further investigation

55

introduction to machine learning: overview

IV. Machine learning approaches (part III)

—— Reinforcement Learning

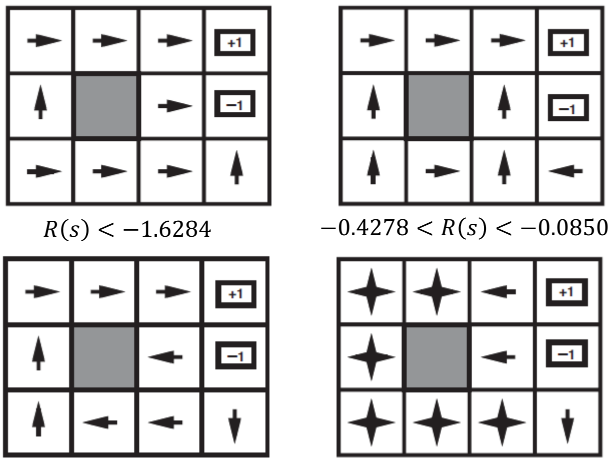
Markov Decision Process

- Components:
  - States s, beginning with initial state s_0
  - Actions a
    - Each state s has actions $A(s)$ available from it
  - Transition model $P(s'|s,a)$
    - Markov assumption: the probability of going to s' from s depends only on s and a and not on any other past actions or states
  - Reward function $\rho(s)$
- The "solution" to an MDP
  - Policy $\pi(s)$: the action that an agent takes in any given state

58

Example: Grid world

- **The careful balancing of risk and reward is a characteristic of MDPs**



$R(s) < -1.6284$

$-0.4278 < R(s) < -0.0850$

$-0.0221 < R(s) < 0$

$R(s) > 0$

59

MDP vs RL

Regular MDP

- Given:
  - Transition model $P(s'|s,a)$
  - Reward function $R(s)$
- Find:
  - Policy $\pi(s)$

Reinforcement learning

- Transition model and reward function initially unknown
- Find
  - Policy $\pi(s)$
- "Learn by doing"

Imagine playing a new game whose rules you don't know; after a hundred or so moves, your opponent announces, "You lose."This is reinforcement learning.

60

Model-based learning vs. model-free learning

- Model-based
  - Learn the model of the MDP (transition probabilities $P(s'|s,a)$ and rewards $\rho(s)$) and try to solve the MDP concurrently
- Model-free
  - Learn how to act without explicitly learning the transition probabilities $P(s'|s,a)$ and rewards $\rho(s)$
  - Q-learning ⟵ Value iteration in MDP
  - Actor-critic learning ⟵ Policy iteration in MDP

61

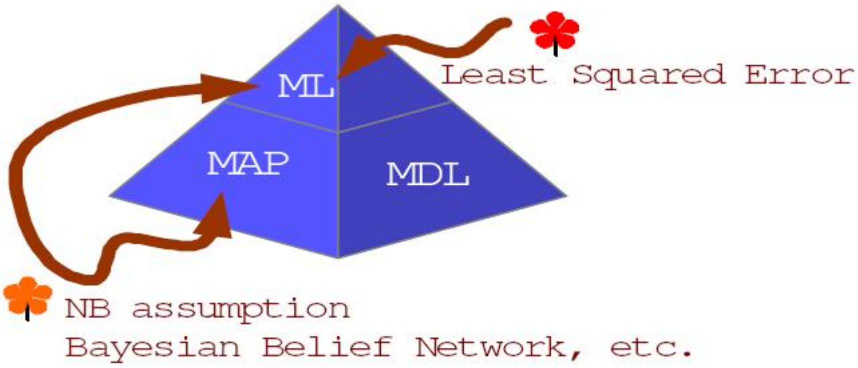
Reinforcement learning key points

- Markov decision process
  - Value iteration
  - Policy iteration
- Reinforcement learning
  - Model-based vs. model-free
  - Q-learning
  - Actor-critic learning

62

V. Theoretical analyses of learning

T1. Bayesian statistics



T2. Minimum Description Length (MDL)

$$h_{\mathrm{MDL}} = \underset{h \in \mathcal{H}}{argmin}\{L_{C_1}(h) + L_{C_2}(D|h)\}$$

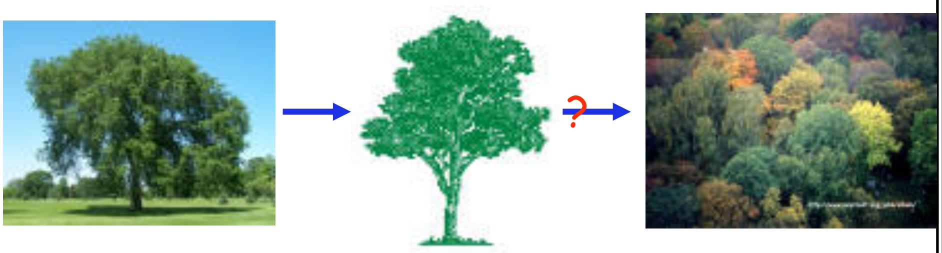
- Tradeoff: complexity of hypothesis vs. the number of errors committed by the hypothesis
  - Prefer a shorter hypothesis that makes a few errors
  - Not a longer hypothesis that perfectly classifies the training data

dealing with ***overfitting*** problem

$$\begin{aligned} h_{\mathrm{MAP}} &= \underset{h \in \mathcal{H}}{argmax}\, \mathrm{P}(D|h)\mathrm{P}(h) \\ &= \underset{h \in \mathcal{H}}{argmax}\{\log_2 \mathrm{P}(D|h) + \log_2 \mathrm{P}(h)\} \\ &= \underset{h \in \mathcal{H}}{argmin}\{\underbrace{-\log_2 \mathrm{P}(D|h)}_{L_{C2}(D|h)} \underbrace{-\log_2 \mathrm{P}(h)}_{L_{C1}(h)}\} \\ &= h_{\mathrm{MDL}} \end{aligned}$$

T3. Hypotheses evaluation

- Performance estimation
  - Given the observed accuracy of a hypothesis over a limited sample of data
  - how well does this estimate its accuracy over additional data?



Hypotheses evaluation

1. Estimating hypothesis accuracy, confidence

   Binomial Dist. → Normal Dist., Confidence interval

2. h_1 outperforms h_2 over some samples
   - In general, h_1 is better than h_2 ?

     Difference of hypotheses → to find one-sided c.i.

3. How to use limited data to learn and estimate?

   Paired t-test, k-fold cross validation, c.i. with $t_{N,k-1}$

Important theoretical background: Central Limit Theorem

Distribution of sample mean Y_{mean} is known although distribution of Y_i is not

T4. PAC Learning Framework: PAC learnable

- For all

  $c \in C$,

  distributions $\mathcal{D}$ over X (instance length: n),

  ε such that $0 < \varepsilon < ½$

  δ such that $0 < \delta < ½$

- L will output a hypothesis $h \in H$ with

  [1] probability $\geq (1 - \delta)$ (probably)

  $\text{error}_{\mathcal{D}}(h) \leq \varepsilon$ (approximately)

  [2] in time that is polynomial in $1/\varepsilon$, $1/\delta$, n, and $size(c)$.

→ C is PAC-learnable (PAC可学习的) by L using H

68 introduction to machine learning: overview

T5. Sample complexity

- **How many training examples are sufficient to learn the target concept?**

(Randomly generated instances, labeled by teacher, instance x generated randomly, teacher provides $c(x)$).

69 introduction to machine learning: computational learning theory

(1) Sample complexity: Finite hypothesis space

- How many training examples are sufficient to successfully learn the target function?
- Consistent learner (一致学习器)

$$|H|e^{-\varepsilon m} \le \delta \quad \Rightarrow \quad m \ge \frac{1}{\varepsilon}(\ln|H| + \ln\frac{1}{\delta})$$

- Agnostic learner (不可知学习器)

$$m \ge \frac{1}{2\varepsilon^2}(\ln|H| + \ln(1/\delta))$$

(2) Sample complexity: Infinite hypothesis space

$$m \ge \frac{1}{\varepsilon}\left(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon)\right)$$

- The Vapnik-Chervonenkis Dimension VC(H) of hypothesis space H defined over instance space X
  - is the size of the largest finite subset of X shattered by H.
- A set of instances S is shattered by hypothesis space H
  - If and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy
- if arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$

If we find **ONE** set of instances of size d that can be shattered, then VC(H) ≥ d.

To show that VC(H) < d, we must show that **NO** set of size d can be shattered.

T6. Mistake bounds

- How many mistakes will the learner make before succeeding?
- E.g.
  - Halving: $\lfloor \log_2 |H| \rfloor$
  - Weighted majority: $\dfrac{k \log_2 \frac{1}{\beta} + \log_2 n}{\log_2 \frac{2}{1+\beta}}$

72

introduction to machine learning: overview

V. Summary

Summary (1)

- Basic concepts
  - What's machine learning
  - Typical machine learning tasks
  - Inductive learning assumption
  - Inductive bias

Summary (2)

- Machine learning approaches
  - Supervised learning
    - Decision tree
    - Bayes learning (MAP, ML, Naïve Bayes)
    - HMM (forward, backward, viterbi) and graphical models (concepts, algorithms)
    - Kernel methods (max margin, SVM, kernel)
    - Instance based learning (KNN, implementation:KD-Tree)
  - Unsupervised learning
    - Clustering (Hierarchical agglomerative, K-means, K-mediods)
  - Reinforcement learning
    - MDP, Q-learning, Actor-critic learning

Summary (2) – final exam req.

- Machine learning approaches
  - Supervised learning
    - Decision tree
    - Bayes learning (MAP, ML, Naïve Bayes)
    - HMM (forward, backward, viterbi) and graphical models (concepts, algorithms)
    - Kernel methods (max margin, SVM, kernel)
    - Instance based learning (KNN, implementation:KD-Tree)
  - Unsupervised learning
    - Clustering (Hierarchical agglomerative, K-means, K-mediods)
  - Reinforcement learning
    - Basic concepts, algorithms

Summary (3)

- Selected advanced topics
  - Overfitting problem
  - Learning with limited data
    - K-fold cross validation
    - Boostrapping sampling
  - Ensemble learning
    - Weighted majority algorithm
    - Bagging
    - Boosting
  - Deep Learning

Summary (3) – final exam req.

- Selected advanced topics
  - Overfitting problem
  - Learning with limited data
    - K-fold cross validation
    - Boostrapping sampling
  - Ensemble learning
    - Weighted majority algorithm
    - Bagging
    - Boosting
  - Deep Learning

Summary (4)

- Theoretical analyses of learning
  - What algorithms are good？
    - Bayes statistics， MDL
  - Confidence degree of learning algorithm
    - Hypothesis evaluation
  - PAC learning framework
    - Possibility of learning (Sample complexity)
      - Finite hypo. space: Consistent learner, Agnostic learner
      - Infinite hypo. space: sample complexity, VC dimension
    - Efficiency of learning (Computational complexity)
  - Effectiveness of learning
    - Mistake bounds
- Experiments, design and analysis

Summary (4) – final exam req.

- Theoretical analyses of learning
  - What algorithms are good?
    - Bayes statistics， MDL
  - Confidence degree of learning algorithm
    - Hypothesis evaluation
  - PAC learning framework
    - Possibility of learning (Sample complexity)
      - Finite hypo. space: Consistent learner, Agnostic learner
      - Infinite hypo. space: sample complexity, VC dimension
    - Efficiency of learning (Computational complexity)
  - Effectiveness of learning
    - Mistake bounds
- **Experiments, design and analysis**

80

introduction to machine learning: overview

考试： 6月9日 同上课时间、地点

闭卷

可以带计算器

答疑：6月7日（周三） 14:00-17:00 FIT 1-507