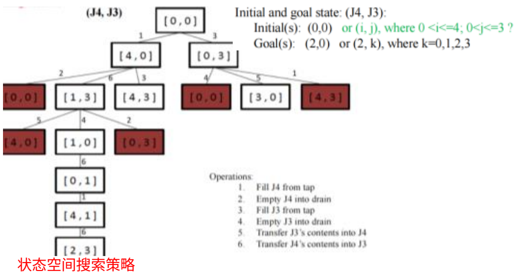
1-2结构和状态空间搜索

欧拉:一个图包含0/2个奇数次节点;当所有顶点的度数都是偶数时,可回原点

状态空间法求解问题

N Nodes: 部分问题解决状态. A Arcs: 解决问题过程中的步骤

S Start state GD Goal state: 访问状态的可测量属性，发展路径的属性



状态空间搜索策略

数据驱动搜索: facts & rules→ new facts → goal Forward chaining

目标驱动搜索: goal → sub-goals → facts Backward chaining

相似: search the same state space

不同: the order and actual number of states searched can differ;

problem properties determine preferred strategy

回溯法 DE→NSL→CS→SL

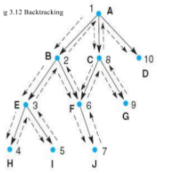
CS=Current State（目前正在考虑的状态）

SL= State List（当前路径上的状态列表）

NSL=New State List（新状态列表包含等待计算的节点）（未处理状态）

DE=Dead Ends（状态列表和目标节点失败）

Suppose *G* is the Goal State



A trace of backtrack on the graph of Figure 3.12

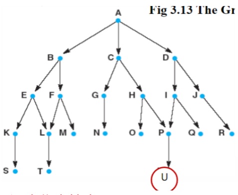
Initialize: SL = [A]; NSL = [A]; DE = [ ]; CS = A;

AFTER ITERATION	CS	SL	NSL (Unprocessed)	DE
0	A	[A]	[A]	[ ]
1	B	[B A]	[B C D A]	[ ]
2	E	[E B A]	[E F B C D A]	[ ]
3	H	[H E B A]	[H I E F B C D A]	[ ]
4	I	[I E B A]	[I E F B C D A]	[H]
5	F	[F B A]	[F B C D A]	[E I H]
6	J	[J F B A]	[J F B C D A]	[E I H]
7	C	[C A]	[C D A]	[B F J E I H]
8	G	[G C A]	[G C D A]	[B F J E I H]

广度优先搜索 (BFS): First-in-first-out (FIFO)以逐层的方式探索搜索空间

Open: 已生成的状态，但其子状态尚未被检查

Closed: 已经检查过的状态



open = [A]; closed = [ ]
open = [B,C,D]; closed = [A]
open = [C,D,E,F]; closed = [B,A]
open = [D,E,F,G,H]; closed = [C,B,A]
open = [E,F,G,H,I,J]; closed = [D,C,B,A]
open = [F,G,H,I,J,K,L]; closed = [E,D,C,B,A]
open = [G,H,I,J,K,L,M] (as L is already on open); closed = [F,E,D,C,B,A]
open = [H,I,J,K,L,M,N]; closed = [G,F,E,D,C,B,A]

深度优先搜索 (DFS): Last-in-first-out (LIFO)

当对一个状态进行检查时，其所有子状态及其后代状态都会在该状态的兄弟状态之前被检查。

Open: 已生成的状态，但其子状态尚未被检查

Closed: 已经检查过的状态

1. open = [A]; closed = [ ]
2. open = [B,C,D]; closed = [A]
3. open = [E,F,C,D]; closed = [B,A]
4. open = [K,L,F,C,D]; closed = [E,B,A]
5. open = [S,L,F,C,D]; closed = [K,E,B,A]
6. open = [L,F,C,D]; closed = [S,K,E,B,A]
7. open = [T,F,C,D]; closed = [L,S,K,E,B,A]
8. open = [F,C,D]; closed = [T,L,S,K,E,B,A]
9. open = [M,C,D], L is already on closed; closed = [F,T,L,S,K,E,B,A]
10. open = [C,D]; closed = [M,F,T,L,S,K,E,B,A]

and so on until either U is discovered or open = [ ]

优缺点:

BFS: shortest path;

never get trapped

DFS: less memory,

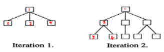
space only B*n. BFS need B^n

Prone to getting lost

(infinite depth trap

+missing the optimal path)

Depth-First Search with Iterative Deepening



Shortest path，Space: B*n

b: maximum branching factor of the search tree搜索树的最大分支因子

d: depth of the least-cost solution最低成本解的深度

m: maximum depth of the state space状态空间的最大深度

Uninformed	BFS	DFS	DFS ID
Completeness	Yes	No	Yes
Optimality	Yes	No	Yes
Time	b^d	b^m	b^d
Space	b^d	bm	bd
Informed	Greedy	A*	Minimax
Completeness	No	Yes	Yes
Optimality	No	Yes	Yes
Time	b^m	b^m	b^m
Space	b^m	b^m	bm

1-3 启发式搜索和游戏

无信息搜索策略: 只使用问题定义中可用的信息,系统性产生新状态,Inefficient

信息搜索策略: 使用特定问题的知识来指导;使用和评估函数来估计节点的可取性;使用更有效的启发式应用搜索

启发式搜索: 算法使用启发式度量来搜索状态空间;个人发现和经验

劣势: fallible, limited information, seldom predict, sub-optimal, can't be eliminated

爬山法: 选出表现最佳的孩子继续进行培育;其兄弟姐妹以及父母均不再保留。当搜索达到比其任何子女表现都好的状态时便停止。

最佳优先搜法:利用状态的启发式成本选择最佳节点

设计启发式

Tiles out of places:不符合点的数量

Sum of distance out of place: 所有移回去的距离和

最佳优先搜索: f(n) = g(n)+h(n)

g(n)路径成本函数 = 从初始状态到当前状态的成本

h(n)启发式函数 = 到目标的最短路径的估计成本,不大于实际成本，可接受f :

Greedy Best-First(n)=h(n), g(n)=0，只看子节点到目标距离

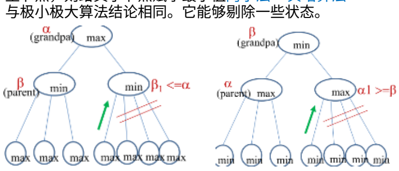
A*: 扩展具有最低g+h值的节点，其中h是可接受的目标距离+来时路

极小极大算法：

每个起始节点都被赋予值 1 或 0

如果父节点是 MAX 类型节点，则给其子节点赋予最大值如果父节点是 MIN 类型节点，则给其子节点赋予最小值阿尔法 - 贝塔算法：

与极小极大算法结论相同。它能够剔除一些状态。



1-4 数据挖掘与关联分析

关联规则挖掘：发现表明强相关性/关系的模式。

项集：零个或多个项目的集合

关联规则: X→Y, X∈I, Y∈I, X∩Y= ∅

Support: specify how often a rule is applicable in a given dataset

Confidence: specify how frequently items in Y appear in transactions that contain X

$$\text{Support}(X) = \sigma(X)/|T| = P(X) \quad \text{Support of itemset } X\text{: the Probability of } X$$

$$\text{Support}(X \to Y) = \frac{\sigma(X \cup Y)}{|T|} = P(X \cup Y)$$

$3^d-2^{d+1}+1$ possible rules

2^k-1 possible items

$$\text{Confidence}(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} = \frac{P(X \cup Y)}{P(X)} = P(Y \mid X)$$

两个任务①Frequent Itemset Generation：Find all itemsets that satisfy the minimum support threshold (minsup) . These itemsets are called frequent itemsets . ②Rule Generation：Extract all high-confidence rules from the frequent itemsets. These rules are called strong rules .

Aprior principle 生成候选项集 候选项集修剪

Ex.4.2 2) Additional: Apriori :

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

1-itemset	Count - σ
A	8
B	7
C	6
D	5
E	3

C1= 1-itemset : { A, B, C, D, E } = L1

C2' =C2= { AB: 5, AC:4, AD:4,AE:2, BC:5,BD:3, ~~BE:1~~, CD:3,CE:2, DE:2 }

L2= { AB: 5, AC:4, AD:4,AE:2, BC:5,BD:3, CD:3,CE:2, DE:2 }

C3' = { ABC, ABD, ~~ABE~~, ACD, ACE, ADE, BCD, CDE, ~~BCE~~, ... }

C3= { ABC: 3, ABD:2, ACD: 2, ~~ACE:1~~, ADE:2, BCD:2, ~~CDE :1~~}

L3 = {ABC: 3, ABD:2, ACD: 2, ADE:2, BCD:2 }

C4' = {ABCD, ~~ABDE, ACDE,~~ ... }

C4 = {~~ABCD:1~~}

L4= { }

Apriori finds Frequent Patterns: 5+9+5 =19

L3: ABC: 3, ABD:2, ACD: 2, ADE:2, BCD:2,

L2: AB: 5, AC:4, AD:4,AE:2, BC:5,BD:3, CD:3,CE:2, DE:2

L1: A:8, B:7, C:6, D:5, E:3

FP-Growth finds FP: 19

E=3: AE=2; CE=2; DE=2; **ADE=2;**

D=5: **BCD=2**; BD=3; CD=3; AD=4; **ABD=2; ACD=2;**

C=6: **ABC=3**; AC=4; BC=5;

B= 7: AB=5;

A=8

The FP-Growth Algorithm

support→Head Table→Sorted F-list→FP-tree → conditional pattern base→conditional 1 FP→conditional f-list conditional FP-tree

FP-tree: a novel data structure

FP-Growth: an efficient mining method

项集是否为频繁项集？ 找到 'A' 的条件模式基。

统计 'C' 在该条件模式基中的总支持度计数。

将这个计数与最小支持度计数进行比较。

如果 'C' 的总支持度计数大于等于最小支持度计数，则 {A, C} 是频繁的。

如果 'C' 的总支持度计数小于最小支持度计数，则 {A, C} 不是频繁的。

Lift: Association Rules Evaluation

If |Y|=k→ 2^k-2 candidate rules

Lift: a simple correlation measure between X and Y

$$\text{Lift}(X,Y) = \frac{\text{Confidence}(X \to Y)}{\text{Support}(Y)} = \frac{P(X \cup Y)}{P(X)P(Y)} = \frac{P(Y \mid X)}{P(Y)}$$

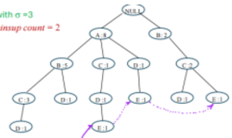
如何看待这个指标:

$$\text{Lift}(X,Y)\begin{cases} =1, & \text{if } X \text{ and } Y \text{ are independent; 彼此独立} \\ >1, & \text{if } X \text{ and } Y \text{ are positively correlated; 正相关} \\ <1, & \text{if } X \text{ and } Y \text{ are negatively correlated. 负相关} \end{cases}$$

ABCD:3 denotes 4-itemList: {A,B, C, D} with σ =3

Exercise 4.2 : FP-Growth discussions *minsup count = 2*

TID	Items (Sorted FP)
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}



f-list : A –B –C –D - E

Head-Table

1-itemset	Count - σ	HP
A	8	
B	7	
C	6	
D	5	
E	3	

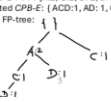
Conditional-Pattern-Base of *E*:

{ ACD:1, AD:1, BC:1 }

CPB-E 1-FP: {A:2, C:2, D: 2, ~~B:1~~}

Sorted *CPB-E*: { ACD:1, AD: 1, C:1}

The FP-tree:



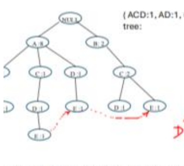
CPB-E定义

前缀路径集合

（删除不满足

min sup的单项

之后的）

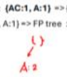


(ACD:1, AD:1, C:1) => FP tree:

Look for FP in CPB – *D|E*

CPB- D|E: (AC:1, A:1) => (A:2; ~~C:1~~)

FP-: (A:1, A:1) => FP tree :



Look for FP in CPB – *C|E*

CPB- C|E: (A:1)=> (~~A:1~~)

FP-: ( ) => no FP tree

Look for FP in CPB – *A|E*

CPB- *A|E : { }*

FP- : { } => *no* FP tree

Look for FP in conditional pattern base - *E*

CPB-E: (ACD:1, AD:1, BC:1) => (A:2; ~~B:1~~; C:2; D:2)

FP-: (ACD:1, AD:1, C:1) => FP tree

|*DE*: { AC:1, A:1} => (A:2,~~C:1~~) => f-list: (A:1,A:1)

|*CE*: {A:1 } => (~~A:1~~) => ()

|*AE*: ( )

FP |E : AE=2; CE=2; DE=2; ADE=2

Item	Conditional Pattern Base	Conditional 1 FP=> C. f-list
E	{ ACD:1, AD:1, BC:1 }	{ A:2, C:2, D:2} => {ACD:1, AD:1, C:1} |E
D	{ABC:1, AB:1, AC:1, A:1, BC:1 }	{ A:4, B:3, C:3} => {ABC:1, AB:1, AC:1, A:1, BC:1} |D
C	{ AB:3, A:1, B:2}	{A:4, B:5} => {AB:3, A:1, B:2} |C
B	{ A:5 }	{A:5} |B
A	∅	∅

FP 树仅有一条分支可以枚举频繁模式

1.A key observation is that if a conditional FP - tree has only a single branch (i.e., it is just a path rather than a branching tree), every subset of the items along that path must appear in exactly the same transactions that support the entire path.

2.In other words, there is no further branching to explore, and all combinations of the items in that single path are guaranteed to be frequent.

2-2分类与决策树

Attribute Type	Description	Examples
Nominal	different names to distinguish one object from another. (=, ≠) 不同的名称来区分不同的对象。(=, ≠)	employee ID numbers, eye color, gender, blood type, social group
Ordinal	provide information to order objects. (<, >) 提供对对象排序的信息。(< >)	rating {poor, average, good, excellent}, grades, street numbers
Interval	Differences between values are meaningful, i.e., a unit of measurement exists. (+, -) 数值之间的差异是有意义的，即存在一个测量单位。（+，-）	calendar dates, temperature in Celsius or Fahrenheit
Ratio	Both differences and ratios of values are meaningful. (x , /) 数值的差值和比率都有意义。（加、除）	seconds, days, counts, mass, length, electrical current

决策树定义: 决策树具有类似流程图的树形结构；

每个节点表示对一个属性的测试；每个分支代表测试的结果；

每个叶节点持有类别标签；最顶层的节点是根节点

决策树优点：1.Decisions trees are computationally inexpensive to build. 2.Once a decision tree is built, classifying a test sample is relatively fast, with a worst-case complexity of O(w), where w is the maximum depth (levels) of the tree. 3.Easy to interpret. 4.Quite robust to noises and redundant attributes. 5.Can avoid overfitting problems. 6. Do not require specific types of probability distributions for the attributes. 7. Can achieve classification accuracy

PID	Fever	Cough	Sore Throat	Tiredness	Flu
1	no	yes	no	yes	−
2	no	yes	no	no	−
3	mild	yes	no	yes	+
4	yes	mild	no	yes	+
5	yes	no	yes	yes	+
6	yes	no	yes	no	−
7	mild	no	yes	no	+
8	no	mild	no	yes	−
9	no	no	yes	yes	+
10	yes	mild	yes	yes	+
11	no	mild	yes	no	+
12	mild	mild	no	no	+
13	mild	yes	yes	yes	+
14	yes	mild	no	no	−

Disclaimer: Synthetic data

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$$

$$Info_{Fever}(D) = \frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.694 \text{ bits}$$

no中找-和+

$$Gain(Fever) = Info(D) - Info_{Fever}(D) = 0.940 - 0.694 = 0.246 \text{ bits}$$

Information gain: use ID3 algorithm, minimizes the information (entropy/uncertainty) in resulting partitions to achieve the least randomness or impurity. $Gain(A) = Info(D) - Info_A(D)$

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Gain ratio: information gain favor tests with many classe, C4.5^ Normalizes information gain with a split information

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|}\log_2\left(\frac{|D_j|}{|D|}\right) \qquad GainRatio_A(D) = \frac{Gain(A)}{SplitInfo_A(D)}$$

Gini index: the impurity of D $\quad Gini(D) = 1 - \sum_{i=1}^{2} p_i^2$

$$Gini_A(D) = \frac{|D_1|}{|D|}Gini(D_1) + \frac{|D_2|}{|D|}Gini(D_2) \qquad \Delta Gini(A) = Gini(D) - Gini_A(D)$$

树剪枝定义：In practice, branches due to noises or outliers in the training dataset could be formed.

过拟合Tree pruning can address this over-fitting problem by using an attribute selection measures to assess the goodness of the split and remove unreliable branches.

前剪枝 pre-pruning approach, a tree is pruned by ending its construction early using a leaf node labeled with the most frequent class among the subset samples.

后剪枝 post-pruning approach, subtrees are removed from a " fully grown " trees. Each removed subtree is replaced with a leaf node and labeled with the most frequent class among the subtree.。

随机森林定义: an ensemble classifier that leverages the power of multiple decision trees for making decisions

评估分类器性能

$$accuracy = \frac{TP+TN}{P+N} \qquad F = \frac{2\times precision \times recall}{precision + recall}$$

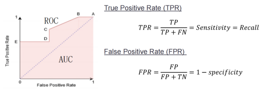
actual class \ predicted class	Yes	No	Total
Yes	TP	FN	P
No	FP	TN	N
Total	P'	N'	P+N

$$error\ rate = \frac{FP+FN}{P+N} = 1 - accuracy$$

$$precision = \frac{TP}{TP+FP} = \frac{TP}{P'} \qquad recall = \frac{TP}{TP+FN} = \frac{TP}{P}$$

$$sensitivity = \frac{TP}{P} \qquad specificity = \frac{TN}{N}$$

Receiver Operating Characteristic (ROC)visualizes the efficiency of a classification model in the detection of true positives while avoiding the occurrence of false positives.The Area Under the Curve (AUC), i.e. the shaded part in the above figure,is the measure of the ability of a classifier to distinguish between classes.



$$TPR = \frac{TP}{TP+FN} = Sensitivity = Recall \qquad FPR = \frac{FP}{FP+TN} = 1 - specificity$$

评估分类器指标: Speed: the computational costs involved in generating and applying the classifier.Robustness: the ability of the classifier to make correct prediction from noisy data or with missing values.Scalability: the ability to construct the classifier efficiently given large amounts of data. Interpretability: the level of understanding and insight that can be provided by the classifier.

分类方法: The Handout Method: The data are randomly partitioned into two independent sets, a training set and a test set. Typically, two-thirds of the data form the training set, and the remaining one-third forms the test set. The training set is used to learn the classifier, whose accuracy is then evaluated by the test set.Cross-Validation: In k-fold cross-validation, the data are randomly partitioned into k mutually exclusive subsets called "folds", D_1, D_2 ,⋯, D, each of approximately equal size. Training and testing are performed k times . In each iteration i, partition D is reserved as the test set, and the remaining partitions are collectively used to learn the classifier.

Leave-one-out is a special case of k-fold cross-validation, where only one sample is left out each time for testing.

2-3最近邻分类器和支持向量机

Minkowski distance: $\quad d(\mathbf{x},\mathbf{y}) = \left(\sum_{k=1}^{n}|x_k - y_k|^r\right)^{1/r}$

r=1: city block, Manhattan, taxicab, L1 norm

$$d(\mathbf{x},\mathbf{y}) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

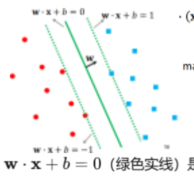
r=2: Euclidean distance

r=∞: supremum, Lmax, L∞ norm distance

$$d_{\text{Supremum}}(x_1, x_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

Cosine similarity: $\cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

SVM: $y_i = \text{sign}(\mathbf{w}\cdot\mathbf{x}_i + b)$



$$\cdot(\mathbf{x}_a - \mathbf{x}_c) = 2$$

$$\|\mathbf{w}\|\times d = 2 \Longrightarrow d = \frac{2}{\|\mathbf{w}\|} \qquad y = \begin{cases} 1, & \text{if } \mathbf{w}\cdot\mathbf{z} + b > 0 \\ -1, & \text{if } \mathbf{w}\cdot\mathbf{z} + b < 0 \end{cases}$$

$$\max d = \frac{2}{\|\mathbf{w}\|} \Longrightarrow \min \frac{\|\mathbf{w}\|^2}{2}$$

subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, N$

No-separable: $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i$

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} + C\left(\sum_{i=0}^{n}\xi_i\right)^k$$

$\mathbf{w}\cdot\mathbf{x} + b = 0$（绿色实线）是决策边界。

$\mathbf{w}\cdot\mathbf{x} + b = 1$ 和 $\mathbf{w}\cdot\mathbf{x} + b = -1$（绿色虚线）分别表示**支持向量的边界。**

【高频概念】什么是支持向量，为什么使得他们是支持向量？

定义The maximum margin linear decision boundary maximizes the distance to the support vectors, which are the closest training samples from each class. For this dataset, the support vectors are {支持向量，例如(2,3) and (4,5)}.

决策边界定义The identified boundary, {决策边界方程，例如：x+y−7=0}, achieves the maximum margin by being equidistant from these support vectors, specifically positioning itself in the middle of the narrowest separation region they define.

原因Given that this is a linearly separable case, the margin boundaries parallel to the decision boundary pass through these two nearest sample points. Meanwhile, based on the decision boundary determined by these two support vectors, the margin can be maximized, satisfying the binary classification condition

$$\lambda_i \geq 0, \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

数学层面，支持向量需要满足： $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$

支持向量选择原因 (Support Vectors) (Vertical Lines垂线)

指的是距离决策边界(Decision Boundary) 最近的两侧的两个点

① Support vectors are Class 1 x3(5,5) and Class 2 x4(9,5)② They are the closest points to the hyper plane (distant = 2) and therefore define it③ Moving or removing them would move the boundary.

PS:① They are the training points closest to the maximum margin hyperplane and lie on the margin boundaries, defining the position of the optimal separation boundary.

2-4 神经网络

激活函数: linear: x, sigmoid: $\sigma(x)=\frac{1}{1+e^{-ax}}$ $\tanh x=\frac{\sinh x}{\cosh x}=\frac{e^x-e^{-x}}{e^x+e^{-x}}$ $\tanh'(x)=1-\tanh^2(x)$

tanh: $\sigma(x)=\tanh(\gamma x) =\frac{e^{2\gamma x}-1}{e^{2\gamma x}+1}$ ReLU: $\sigma(x)=\begin{cases}x, & x\geq 0\\ 0, & x<0\end{cases}=\max(0,x)$

sign: $\sigma(x)=\mathrm{sign}(x)=\begin{cases}+1, & \text{if } x\geq 0\\ -1, & \text{if } x<0\end{cases}$ Leaky ReLU: $\sigma(x)=\begin{cases}x, & x\geq 0\\ ax, & x<0\end{cases}=\max(ax,x)$ where $a\ll 1$

梯度下降：一种迭代优化算法 $w'=w-\eta\frac{\partial J(w)}{\partial w}$

J(w)：一个误差/成本/损失函数→最小化

w：梯度下降法中的参数/权重→通过迭代更新

η：在每次迭代中确定更新幅度

反向传播算法

Minimize the squared errors (a loss function) sum

$$E=\frac{1}{2}\sum_{k=1}^{c}(t_k-o_k)^2=\frac{1}{2}\|t-o\|^2 \quad \Delta w_{jk}=-\eta_w\frac{\partial E}{\partial w_{jk}} \quad \Delta b_j=-\eta_b\frac{\partial E}{\partial b_j}$$

netj:input, oj:output $net_j=\sum_i w_{ji}o_i+b_j$ $o_j=\sigma(net_j)$

Error: k for output units, j for hidden units If $\sigma(x)=1/(1+e^{-x})$ is the

$\delta_k=\sigma'(net_k)(t_k-o_k)$ 导数 $\delta_k=o_k(1-o_k)(t_k-o_k)$

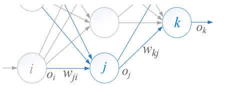
$\delta_j=\sigma'(net_j)\sum_k\delta_k w_{kj}$ $\delta_j=o_j(1-o_j)\sum_k\delta_k w_{kj}$

η: learning rate, t:target, o:output, wkj:j→k

$w_{kj}=w_{kj}+\Delta w_{kj}$ where $\Delta w_{kj}=\eta_w\delta_k o_j$

$w_{ji}=w_{ji}+\Delta w_{ji}$ where $\Delta w_{ji}=\eta_w\delta_j o_i$

$b_j=b_j+\Delta b_j$ where $\Delta b_j=\eta_b\delta_j$



步骤：1.算所有 Unit 的 Input,netj，记得加 bias，只加一次，注意+-

2. 算经过 Attention Func σ 后的 output oj
3. 算 output unit 的输入 Input，同 1
4. Output unit 的 output oj=σ(netj)
5. Error:先算 Output unit. δk. 再算 Hidden unit. δj.
6. Update 值，w&b，看是否要Δ后多一步

梯度消失radients shrink exponentially while back propagating through many layers because σ ' (x)≤0.25 for a sigmoid. Products of many σ ' (x) terms drive δ toward zero, so earlier layers learn extremely slowly or stop learning altogether.

措施 ReLU / variants (LeakyReLU, GELU) keep σ ' (x)≈1 for x>0 ②Proper weight initialization (He/Kaiming, Xavier) maintains variance of activations and gradients. ③Batch / Layer Normalization rescales activations, keeping them in regions with healthy derivatives. ④Residual connections (ResNets, Highway nets) provide short gradient paths that bypass many non linearities.⑤Gradient clipping or adaptive optimizers (Adam, RMSProp) prevent tiny updates after many layers.

Convolutional Neural Networks (CNN)

神经网络，层数大于 3 → 深度学习网网络

池化：下采样操作，最大值/平均值/L2-normpooling

L2 范数规整：防止对训练数据集过度拟合

1. R(w)：权重惩罚 $J'(w)=J(w)+\alpha R(w)$

$R(w)=\|w\|_1=\sum_k|w_k|$, L1 Regularization (LASSO)

$R(w)=\|w\|_2^2=\sum_k(w_k)^2$, L2 Regularization (Ridge)

$R(w)=\|w\|_1+\beta\|w\|_2^2$, Elastic Net Regularization

2. Early stopping: when the validation errors off/begins to increase

卷积层输出维度计算及参数对应关系

$$W_2=\frac{(W_1-F+2P)}{S}+1 \quad H_2=\frac{(H_1-F+2P)}{S}+1 \quad D_2=K$$

CNN优势1. Local receptive fields & weight sharing – convolution kernels reuse parameters, slashing memory and overfitting risk. 2.Translation invariance – achieved through convolutions + pooling, so an object is recognised wherever it appears. 3. Automatic hierarchical feature learning – early layers learn edges/colours, deeper layers learn shapes and objects, avoiding handcrafted descriptors

Recurrent Neural Network (RNN) Handle sequential data

$h_t=\sigma(W_t h_{t-1}+W_x x_t)$ $y_t=F(h_t)$

Generative Adversarial Networks (GANs)

Generate high-quality samples by optimizing the interplay between generator network G and discriminator network D

Transformers: Have encoder and decoder process.

The Multi-Head Attention: use a self-attention mechanism

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

3-1 回归聚类

回归目的：建立模型来描述变量之间的关系。它可以预测某些变量的值，并将这些值赋给其他变量。

有监督学习（Supervised）：类别标签在训练阶段是预定义并提供的，常见任务包括分类和回归。

无监督学习（Unsupervised）：没有预先定义的标签，仅提供数据。常见方法包括主成分分析（PCA）和聚类（Clustering）用于发现数据的结构或表示形式。

半监督学习（Semi-supervised）：只有一小部分训练数据有标签。

弱监督学习（Weakly supervised）标签存在噪声或者标签与目标任务不完全一致

聚类（Clustering）：给定一组数据样本，目标是将数据分组或组织，使得同一组内的数据彼此更加相似，而与其他组的数据差异更大。

簇（Cluster）：一组彼此相似的数据。

聚类目标：将数据点划分为 K 个簇。

划分算法：non-overlapping & exactly or

分层算法：嵌套簇nested clusters

$$\min_{\{\mu_k\}}\min_{\{r_{ik}\}}\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{1}{2}r_{ik}\|x_i-\mu_k\|_2^2$$

K-Means

Initialize: Pick k random points as the cluster center

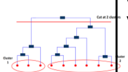
Iterate:1. Assign every data point x to its closest cluster center, according to the given distance metric, i.e., find the μk such that d(x, μk) is minimized. 2.Update the cluster center to be the average of its assigned data points

Stopping criterion: when no points' assignments change

√Simple and cheap

×Initialization 会影响结果 stuck at poor local minimum

×Number of clusters needs to be pre-defined



Hierarchical Agglomerative Algorithm (HAC)

Initialize: Each point as an individual cluster

Iterate:1. Merge the two clusters with the minimum centroid or average linkage

2.Update the centroid of the new cluster (after merging)

Stopping Criterion: only K clusters or 1 cluster left

损失函数：within-cluster sum of squares, WCSS

注意！如果 Merge A1 with (A4,A6), new centroid 要（2C+A1）除以三

Dendrogram: A tree-like diagram that records sequence of merges

优劣 √Initialization: result is independent of it

√Do not assume/pre-define the number of clusters

×More memory- and computationally-intensive

MIN/ Single Linkage:the minimum distance between any pair of two data samples from each cluster

MAX/Complete Linkage:the maximum distance between any pair of two data samples from each cluster

Average Linkage:the average distance between all pairs of two data samples from each cluster

Centroid Distance:the distance between the means of data samples (i.e., centroids) from each cluster

回归: Predict a continuous quantity $f_{w,b}(x)=w^Tx+b$

Accuracy=root mean squares error

Linear Regression

Training:find the best w based on training data Minimize the l2 loss

$$\min_{w,b}\hat{L}(f_{w,b})=\min_{w,b}\frac{1}{N}\sum_{i=1}^{N}(w^Tx_i+b-y_i)^2$$

Loss function: mean squared error between wᵀx+b&yi Minimizing vertical offset

$$\hat{L}(f_w)=\frac{1}{N}\sum_{i=1}^{N}(w^Tx_i-y_i)^2=\frac{1}{N}\|Xw-y\|_2^2$$

Gradient w $\nabla_w\|Xw-y\|_2^2=\nabla_w[(Xw-y)^T(Xw-y)]$

将梯度设为零以获得极小值点： $=\nabla_w[w^TX^TXw-2w^TX^Ty]$

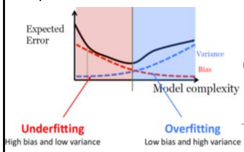
$X^TXw=X^Ty$ $=2X^TXw-2X^Ty$

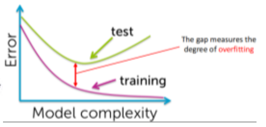
最小二乘估计量： $w=(X^TX)^{-1}X^Ty$

3-2正则化和优化

Challenges: exact model & distribution of data

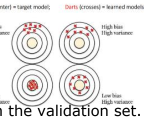
Expected error: the likelihood that misclassified by fθ





偏差-方差权衡

机器学习算法在训练集之外无法泛化的基本困境在于要在两种错误来源之间进行权衡。偏差是由于学习算法中的错误假设而产生的误差。高偏差可能导致算法遗漏特征与目标输出之间的相关关系（例如，模型过于简单 -> 欠拟合）。方差是由于对训练集中的小波动敏感而产生的误差。高方差可能导致算法对训练数据中的随机噪声进行建模，而非预期的输出（例如，模型过于复杂 -> 过拟合）。



Expected error ≈ bias+variance(+noise)

Overfitting≈ValidationError-TrainingError

High complexity→Large Gap→Overfitting

Low Complexity→Small Gap→Underfitting

最好epoch x achieves the best generalization on the validation set.

过拟合the training loss is very low but the validation loss has risen sharply. In other words, overfitting has occurred, so the parameters at epoch x will likely generalize poorly.

容量Because the training loss can continue going down to near - zero while the validation loss increases, the model clearly has sufficient capacity to overfit the training data.

训练集增加1.Training loss typically goes up slightly, because with much more data (and presumably more diversity in the training samples) the network cannot " memorize " them all as easily. 2. Validation loss often goes down, because having many more training examples improves generalization.

正则化以防止过拟合1. Limit the model complexity by reducing the model expressiveness Dropout: ignore layer outputs

Early Stopping: sample the model every few iterations of training, check validation error, when it reach minimum, stop

Weight Sharing: force parameters. Ex.RNN

2. Increase the training data complexity/size,to reduce the variance Add more real training data

Data Augmentation:Geometric: flipping, rotation,shearing,multiple crops,Photometric: color transformations,Other: scaling,add noise,compression artifacts,lens distortions

3-3 降维、PCA和贝叶斯

降维目标find the best-fitting low-dimensional subspace to represent $(x_i)_{i=1}^N$ 1.Give data into d dimensions. 2.Convert them to r<d dimensions. 3.With minimal loss.

高维数问题1. high dimensions=lots of features. 2. Complex system to process: Inefficient algorithm, overfitting to noise/other data corruptions

降维原因 1.Remove Feature Redundancy + Noise 2.Prevent Overfitting: Reduced Model Complexity , Simplified Data distribution 3.Better Visualization

PCA定义 Principal Component Analysis (PCA) is a simple and widely used unsupervised learning method designed to identify the " best" low-dimensional subspace for projecting high-dimensional data.

最佳定义The term " best" can be interpreted in two equivalent ways: (1) minimizing the mean square error (MSE) between the original data and its projections, and (2) maximizing the variance of the projected data along each principal component.

目标These objectives are achieved simultaneously through PCA, making it an effective technique for dimensionality reduction while preserving the most significant patterns in the data.

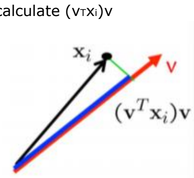
PC1:PC1 is the eigenvector with the largest eigenvalue; it defines the direction along which the data show maximum variance. All subsequent PCs are orthogonal to PC1 and to each other, each explaining the greatest remaining variance.

PC2 与 PC1 正交，次大特征值所对应特征向量

Calculate distance between each point to the mean di

步骤Steps: Apply SVD to the data matrix. Select the unit weight vector Vₖfrom the V matrix, and calculate (vᵀxᵢ)v

- x_i (black): the original data.
- v (red): PCA subspace.
- $(v^Tx_i)v$ (blue): projected data.
- Minimize MSE (green) for all $\{x_i\}$ i.e., the perpendicular offset.



PCA VS线性回归1. Learning target: PCA is unsupervised; linear regression is supervised. 2.Optimisation criterion: PCA maximises variance of projected predictors; regression minimises residual variance of the response.

Bayesian Inference

Conditional probability P(A|B)

Joint probability P(A,B): $P(A,B)=P(A|B)*P(B)$

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

Bayes' Theorem:

$$P(B)=\sum_i P(B|A_i)P(A_i)$$

Prior probability P(A) $P(A|B)=\frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|\neg A)P(\neg A)}$

Posterior P(A|B)

Likelihood P(B|A)

Maximum A Posteriori(MAP): h_{MAP} = argmaxP(A|B)

If P(A) is constant, h_{MAP} = argmaxP(B|A)

$$P(a_1,\ldots,a_d\mid v_j)=\prod_{i=1}^{d}P(a_i\mid v_j)$$

Naive Bayes:

Compare P(yes|<S,C,H,S>)and P(no|<S,C,H,S>) can use to compare P(yes,<S,C,H,S>) and P(no,<S,C,H,S>)

P(yes, <S,C,H,S>)=P(yes)P(<S,C,H,S>|yes)

P(<S,C,H,S>|yes)=P(S|yes)P(C|yes)P(H|yes)P(S|yes)

【高频题】为什么该决策边界能够在两个类别之间形成最大间隔

In the case of linear separability:

The objective function of SVM is to find, among all hyperplanes that can correctly separate the data, the one that maximizes the minimum distance from the class boundary to the hyperplane;

$$\max d=\frac{2}{\|w\|}\Rightarrow\min\frac{\|w\|^2}{2}$$

In the case of a hard margin (linearly separable), SVM solves the following constrained optimization problem:

$$\min_{w,b}\frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w\cdot x_i+b)\geq 1,\quad i=1,\ldots,N,$$

Minimizing $\frac{1}{2}\|w\|^2$ is equivalent to maximizing the margin $d=\frac{2}{\|w\|}$. At the same time, the constraint $y_i(w\cdot x_i+b)\geq 1$ ensures correct classification and fixes the minimum distance from the support vectors to the decision boundary at $\frac{1}{\|w\|}$.

Therefore, the decision hyperplane obtained by applying SVM is guaranteed to form the largest margin between the two classes.

SVM-2 用0训练 margin

由于 SVM-2 的约束条件仅仅是 wx+b⩾0 或 wx+b<0，因此可以随意对 (w,b) 进行缩放（例如乘以 2、10 或 0.1），而仍然满足相同的不等式。这意味着 SVM-2 的间隔并非固定在某个有限值上——解决方案并非唯一，间隔可以通过适当的缩放变得任意大。相比之下，SVM-1 将支持向量固定为 ±1，因此其间隔固定为 2/∥w∥因此，与 SVM-1 相比，SVM-2 的边界实际上是“无界的”（或者说“未明确界定的”），因为不存在对边界进行强制归一化的操作，使得边界无法被限定到某个特定的值。

SVM-1's constraints require a functional margin of at least 1 (wx+b⩾1 and wx+b⩽?1), leading to a positive geometric margin 2/||w|| which is maximized. SVM-2's constraints (wx+b⩾0 and wx+b⩽0) only require the data to be on the correct side of the w?x+b=0 hyperplane, effectively allowing a functional margin of zero. This means SVM-2 acts like a simple perceptron finding any separating hyperplane, without optimizing for or guaranteeing a positive margin.

判断Would SVM-3 make any classification errors on the training data?" “是否会产生分类错误”时，我们核心关注的是决策边界x+y=0 Decision Boundary以及样本点相对于它的位置，而不是训练时的约束条件（Training Constraints) x+y=+-1,这个边界是在训练 SVM 模型时，用来找到最优参数 w 和 b 的和找maximum margin

SVM-3 uses parameters w ' =w/2 and b ' =b/2. Its decision boundary is defined by the equation (w/2)x+(b/2)=0. This equation is equivalent to wx+b=0, which is the exact same decision boundary as SVM-1. Since SVM-1 is trained on linearly separable data and makes no classification errors (as implied by its constraints wx+b⩾1 for class 1 and wx+b⩽1 for class 0), SVM-3, sharing the identical decision boundary, will also correctly classify all training samples. The classification of a point depends only on which side of the decision boundary it falls, and this remains unchanged.

KNN与SVM对比

Nearest Neighbor Classifier assigns the class label of a test sample to be the same as that of its nearest training sample. It requires a similarity metric to compare two samples. This method requires almost no training but needs to store all training samples and their labels, and involves significant computation during classification.

Support Vector Machine (SVM) learns a binary classification decision boundary by maximizing the margin between two hyperplanes. SVM adopts a soft-margin approach to learn a decision boundary that can tolerate small training errors. For both separable and non-separable cases, the SVM learning problem can be solved by constructing a Lagrangian dual optimization problem. (Introducing slack variables)

SVM多分类：SVM can be extended to multi-class classification.

One-vs-All: All k classifiers are used to evaluate the test sample, and the class corresponding to the SVM classifier with the highest decision function value is selected as the final classification result.

决策树vsKNN 1.Model type:Decision tree is eager (model built once); k-NN is lazy (all computation deferred to query time). 2.Prediction cost: Tree: O(log N) after training; k-NN: O(N · d) per query (distance to every point).3.Interpretability & feature weighting Tree gives explicit IF - THEN rules and can ignore irrelevant attributes; k-NN is opaque and sensitive to irrelevant or differently-scaled features.

FP - Growth VS Apriori:FP - Growth uses a highly compressed representation of the dataset called an FP - tree (a prefix tree), which avoids storing many duplicates. It avoids the repeated, full - database scans required by Apriori (Apriori typically scans once per candidate - itemset size), because it recursively mines " conditional " FP - trees instead.

模型容量Model capacity refers to the ability of the model to learn complex patterns. The training loss continues to decrease while the validation loss starts to increase. This divergence is a classic sign of overfitting, which indicates the model has excessive capacity, allowing it to memorize noise in the training data rather than learning generalizable patterns.

模型复杂度引发的误差和方差的变化解释

1.Bias will tend to increase (the smaller network may underfit more). Lower-capacity models (e.g., fewer hidden nodes) have fewer parameters, so they cannot contort themselves to fit every tiny fluctuation in the training set. They will produce more " stable " predictions under small changes to the training data—this is lower variance.

2.Variance will tend to decrease (the smaller network is less likely to overfit). we mean that the model becomes less sensitive to small fluctuations in the training data. High-capacity models (e.g., large networks) have many parameters and can fit training data very flexibly. Small changes or noise in the training set can lead to large changes in the learned parameters, and hence large changes in predictions—this is high variance.

k-最近邻（k-Nearest Neighbor，简称KNN）算法是监督学习。一种基础且广泛应用的机器学习算法，主要用于分类和回归问题。它是一种非参数学习算法，也就是说，它不对数据的分布做任何假设。“近朱者赤，近墨者黑”：对于一个新的未知数据点，KNN会计算它与训练数据集中所有已知数据点的距离。找到最近的k个邻居：然后，算法会选出与这个新数据点距离最近的k个数据点（即“邻居”）。这里的“k”是一个由用户定义的整数，通常是一个较小的奇数。投票决定/平均值:分类问题：在这k个最近邻居中，算法会统计每个类别的数量。新数据点会被归类到数量最多的那个类别。回归问题：算法会计算这k个最近邻居的属性值的平均值（或加权平均值），并将这个平均值作为新数据点的预测值。

PCA 和线性回归的区别？Learning Type and Goal:

PCA: An unsupervised method used for dimensionality reduction or feature extraction. It finds patterns in the input data itself, without using a target variable.

Linear Regression: A supervised method used for prediction. It models the linear relationship between input features and a continuous target variable, requiring labeled data (inputs and corresponding targets).

Role of Variables:

PCA: Treats all features equally to find principal components that capture maximum variance. It does not distinguish between independent and dependent variables.

Linear Regression: Clearly separates variables into independent features and a dependent target, aiming to predict the target based on the features.