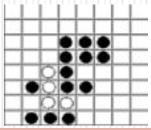
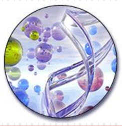




Welcome to

Introduction to Machine Learning!



Coffee Time

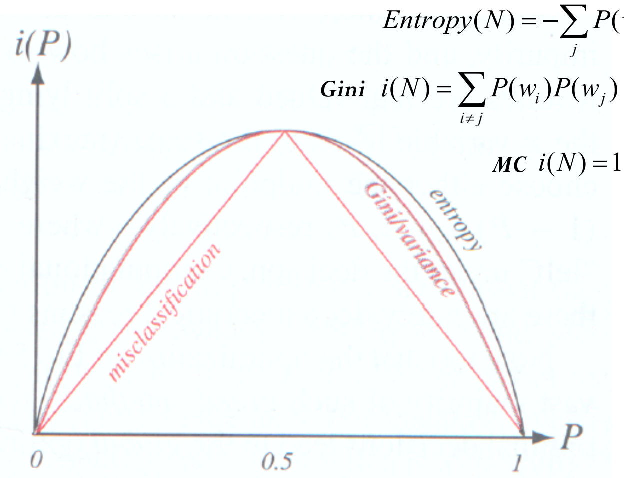
Debug: Update on Impurity Issue

Last class: Impurity

$$Entropy(N) = -\sum_j P(w_j)\log_2 P(w_j)$$

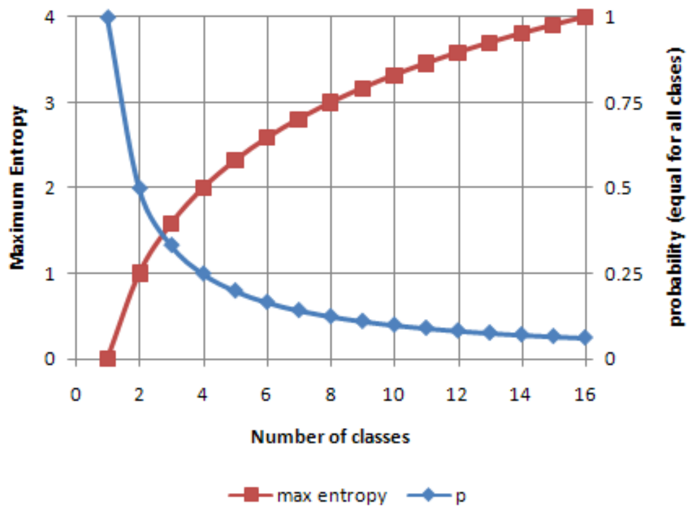
$$\textbf{Gini} \quad i(N) = \sum_{i \neq j} P(w_i)P(w_j) = 1 - \sum_j P^2(w_j)$$

$$\textbf{MC} \quad i(N) = 1 - \max_j P(w_j)$$



Impurity (Entropy)

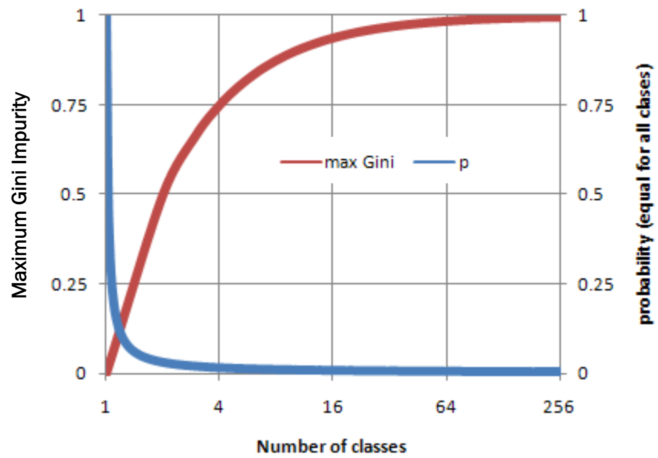
$$Entropy(N) = -\sum_j P(w_j)\log_2 P(w_j)$$



Impurity (Gini)

$$i(N)=\sum_{i\neq j}P(w_i)P(w_j)=1-\sum_j P^2(w_j)$$

Maximum Gini impurity happens at 1-n*(1/n)^2=1-1/n



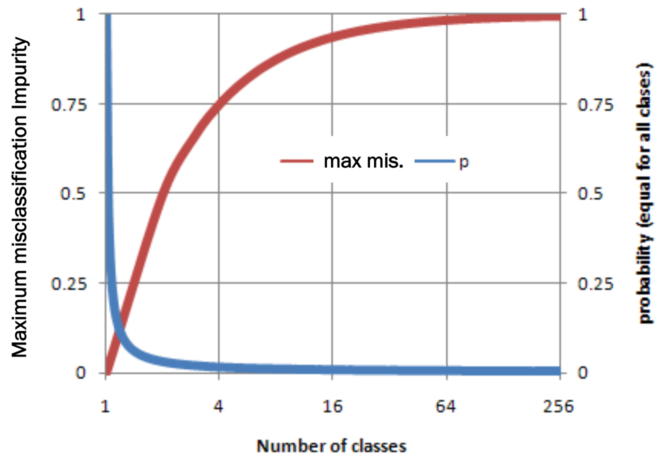
Impurity (Misclassification)

$$i(N)=1-\max_j P(w_j)$$

Maximum Gini impurity happens at 1-n*(1/n)^2=1-1/n

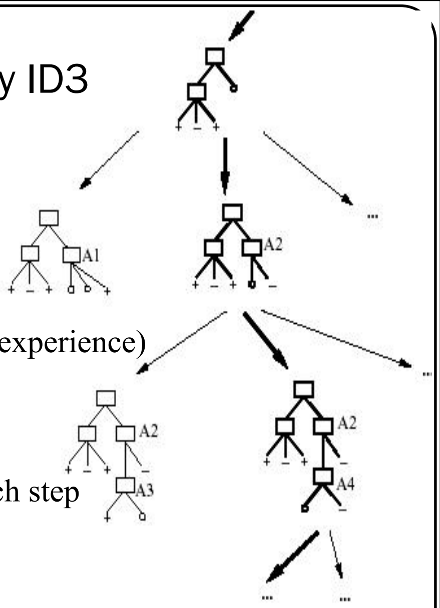
For n classes, Maximum Misclassification impurity = Maximum Gini impurity

x{1/n}=1-1/n



Topic 2. Decision Tree (II)

Hypothesis space search by ID3



- Hypothesis space is complete
  - Target function surely in there
- Output a single hypothesis
  - Can't play over 20 questions (by experience)
- No back tracking
  - Local minima…
- Use all the data in the subset for each step
  - Statistically-based search choices
  - Robust to noisy data

Inductive bias in ID3

- Note H is the power set of instances X
  - No restriction on the hypothesis space
- Preference for trees with high IG attributes near the root
  - Attempt to find the shortest tree
  - Bias is a ***preference*** for some hypotheses (search bias), rather than a ***restriction*** of hypothesis space H (language bias).
  - Occam's razor: prefer the shortest hypothesis that fits the data

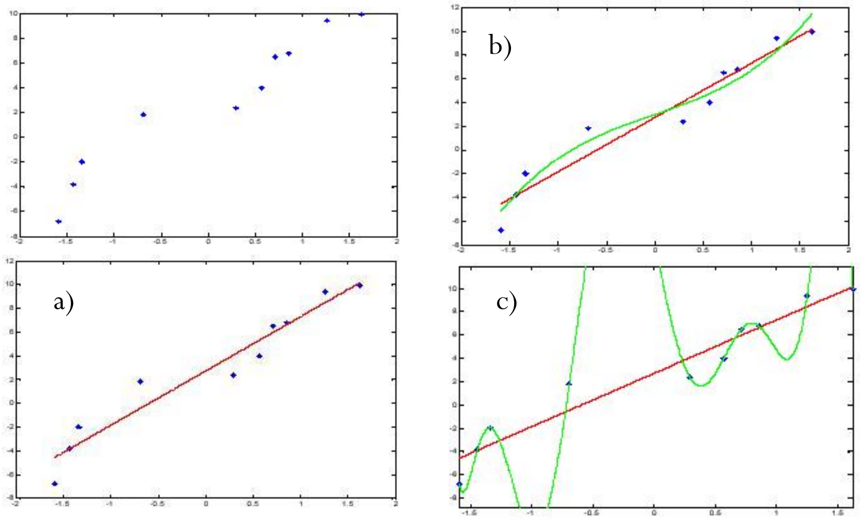
Occam's razor

- Just gives an idea here, no detail discussion
- For more information:
  - Domingos, The role of Occam's Razor in knowledge discovery. Journal of Data Mining and Knowledge Discovery, 3(4), 1999.

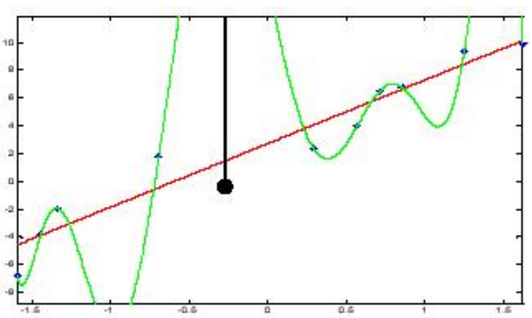
Decision Tree

- Introduction -- basic concepts
- ID3 algorithm as an example
  - Algorithm description
  - Feature selection
  - Stop conditions
  - Inductive bias for ID3
- Over-fitting and Pruning

What's over-fitting ?



What's over-fitting？



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

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

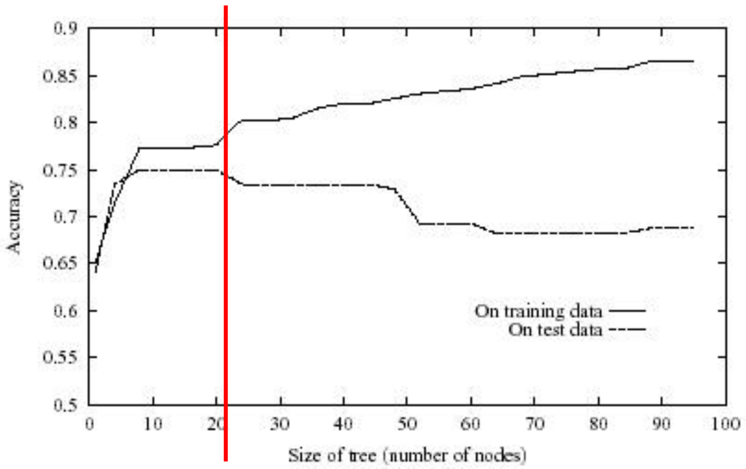
  AND

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

An example of over-fitting in DTree

- Each leaf corresponds to a single training point and the full tree is merely a convenient implementation of a lookup table

Over-fitting in Decision Tree Learning



Avoid over-fitting

- Two ways of avoid over-fitting for DTree
  I. Stop growing when data split not statistically significant (pre-pruning)
  II. Grow full tree, then post-pruning

For Option I:

Hard to estimate the size of the tree

Pre-Pruning: When to stop splitting (I) Number of instances

- Frequently, a node is not split further if
  - The number of training instances reaching a node is smaller than a certain percentage of the training set
    - (e.g. 5%)
  - Regardless the impurity or error.
  - Any decision based on too few instances causes variance and thus generalization error.

Pre-Pruning: When to stop splitting (2) Threshold of information gain value

- Set a small threshold value, splitting is stopped if $\Delta i(s) \leq \beta$
- Benefits: Use all the training data. Leaf nodes can lie in different levels of the tree.
- Drawback: Difficult to set a good threshold

Avoid over-fitting

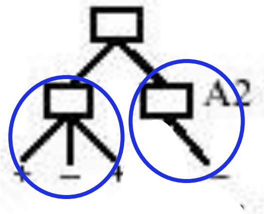
- Two ways of avoid over-fitting for D-Tree
  I. Stop growing when data split not statistically significant (pre-pruning)
  II. Grow full tree, then <u>post-pruning</u>

For option II:

- How to select "best" tree?
  - Measure performance over training data (statistical pruning)
    - Confidence level (will be introduced later)
  - Measure performance over separate validation data set
- MDL (Minimize Description Length 最小描述长度):
  minimize (*size*(tree) + *size*(misclassifications(tree)))

Post-pruning (1). Reduced-Error pruning

- Split data into training set and validation set
  - Validation set:
    - Known label
    - Test performance
    - No model updates during this test!
- Do until further pruning is harmful:
  - Evaluate impact on validation set of pruning each possible node (plus the subtree it roots)
  - Greedily remove the one that most improves validation set accuracy

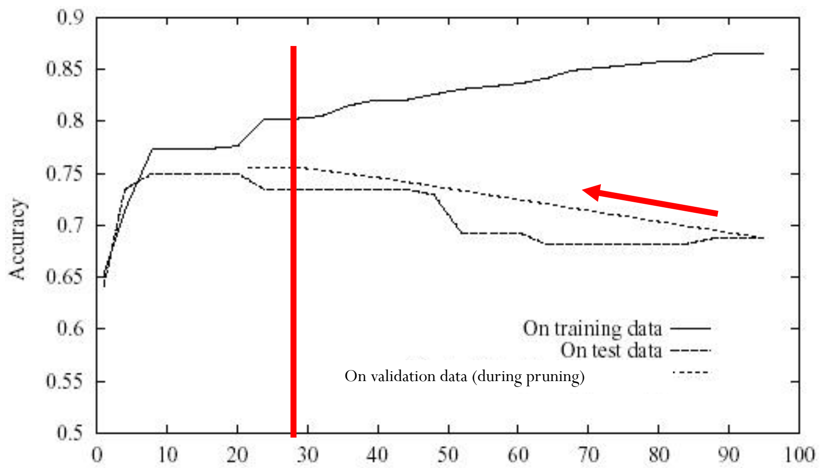


How to assign the label of the new leaf node?

Supplement: strategies of the new leaf node label after pruning

- Assign the most common class.
- Give the node multiple-class labels
  - Each class has a support degree (based on the number of the training data with each label)
  - On test: select one class with probability, or select multiple classes
- If it is the regression tree (numeric labels), can be averaged, or weighted average.
- ……

Effect of Reduced-Error pruning



Post-pruning (2). Rule Post-pruning

1, Convert tree to equivalent set of rules

- e.g. if (outlook=sunny)$\wedge$(humidity=high) then playTennis = no

2, Prune each rule by removing any preconditions that result in improving its estimated accuracy

- i.e. (outlook=sunny), (humidity=high)

3, Sort rules into desired sequence (by their estimated accuracy) .

4, Use the final rules in the same sequence when classifying instances.

(after the rules are pruned, it may not be possible to write them back as a tree anymore.)

One of the most frequently used methods, e.g. in C4.5.

Why convert the decision tree to rule before pruning?

- Independent to contexts.
  - Otherwise, if the tree were pruned, two choices:
    - Remove the node completely, or
    - Retain it there.
- No difference between root node and leaf nodes.
- Improve readability

Brief overview of Decision Tree Learning (Part 1)

- Introduction -- basic concepts
- ID3 algorithm as an example
  - Algorithm description
  - Feature selection
  - Stop conditions
  - Inductive bias for ID3
- Over-fitting and Pruning
  - Pre-pruning
  - Post-pruning: Reduced-Error pruning, Rule post-pruning
  - In practice, pre-pruning is faster, post-pruning generally leads to more accurate trees

Brief overview of Decision Tree Learning (Part 1)

- The basic idea come from human's decision procedure
- Simple, easy to understand: If…Then…
- Robust to noise data
- Widely used in research and application
  - Medical Diagnosis (Clinical symptoms → disease)
  - Credit analysis (personal information → valuable custom?)
  - Schedule
  - ……
- A decision tree is generally tested as the benchmark before more complicated algorithms are employed.

Part 2: Advanced Topics in Decision Tree

Problems & improvements

1. Continuous attribute value

$$x_l < x_s < x_u$$

Temperature	40	48	60	72	80	90
decision	No	No	Yes	Yes	Yes	No

- Create a set of discrete attribute value
- Options:
  - I. Get the medium of the adjacent values with different decisions

$$x_s = (x_l + x_u)/2$$

(Fayyad proved that thresholds lead to max IG satisfies the condition in 1991)

  - II. Take into account the probability $x_s = (1-P)x_l + Px_u$

2. Attributes with many values

Problem:

- Bias: If attribute has many values, IG will select it
  - e.g. Date as an attribute
- One possible solution: use *GainRatio* instead

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

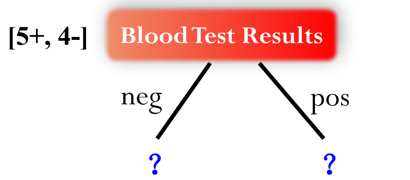
$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

Punish factor， entropy of S on A

3. Unknown attribute values

BTR	Temp	…	label
neg	normal	…	-
neg	normal	…	-
neg	normal	…	-
neg	normal	…	-
neg	high	…	+
pos	normal	…	+
pos	high	…	+
pos	high	…	+
?	normal	…	+

With missing data



Most common training: neg

[2+, 4-] **[3+, 0-]**

Most common according to the label: pos

[1+, 4-] **[4+, 0-]**

Assign probability: neg 5/8, pos 3/8

[(1+5/8)+, 4-] **[(3+3/8)+, 0-]**

4. Attributes with costs

- Tan & Schlimmer (1990)

$$\frac{Gain^2(S,A)}{Cost(A)}$$

- Nunez (1988)

$$\frac{2^{Gain(S,A)}-1}{(Cost(A)+1)^w}$$

  - w:[0,1] importance of cost

What's more …

- Perhaps the simplest and the most frequently used algorithm
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Small computation costs
- Decision Forest:
  - Many decision trees by C4.5
- For More information about C4.5 (C5.0):
  - http://www.rulequest.com/see5-info.html
  - Ross Quinlan's homepage:

    http://www.rulequest.com/Personal/

Inductive learning hypothesis

Min Zhang
z-m@tsinghua.edu.cn

Inductive learning hypothesis

- Much of the learning involves acquiring general concept from specific training examples.

- Inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data.
  - **Notice: over-fitting problem**

Inductive learning hypothesis

- The *Inductive Learning Hypothesis*:

  Any hypothesis found to approximate the target function well over **a sufficiently large set** of training examples will also approximate the target function well over unobserved examples.

  (任一假设若在足够大的训练样例集中很好地逼近目标函数，它也能在未见实例中很好地逼近目标函数)

Topic 3. Bayesian Learning

Min Zhang
z-m@tsinghua.edu.cn

Background of Bayesian Learning

- Discover relationship between two events

(causal analysis, the precondition & the conclusion)

- A → B
  - e.g. pneumonia → lung cancer?
  - Hard to tell directly
- Reversed thinking
  - e.g. How many lung cancer patients have suffered from pneumonia?
- In our daily life, disease diagnose by a doctor is a Bayesian learning process.

Bayes Theorem

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

Thomas Bayes (1702~1761)

An example: Lab test result: +， has a particular form of cancer?

- $P(h \mid D)$ = the posterior probability of h

  $P(h \mid D)$: prob. of -- test result='+' then has cancer

- $P(h)$ = the prior probability of h

  $P(h)$: prob. of -- has cancer

- $P(D)$ = the prior probability of D

  $P(D)$: prob. of -- test result = '+'

- $P(D \mid h)$ = the probability of D given h

  $P(D \mid h)$: prob. of -- has cancer then test result ='+'

Bayes Theorem

- $P(h)$
  - hypotheses: mutually exclusive
  - H space: totally exhaustive
  - $\sum P(h_i) = 1$
- $P(D)$
  - D is taken as the sample of all possible data
  - Independent with h
  - Can be ignored in comparison among different hypotheses
- $P(D \mid h)$
  - log likelihood $\log(P(D \mid h))$

An example

- Lab test result: +， has a particular form of cancer?
  - Correct positive: 98% (has cancer, then test result +)
  - Correct negative: 97% (not cancer, then test result -)
  - Over entire population of people, only 0.008 have cancer

$$\boldsymbol{P(cancer \mid +) = ?}$$

$$\boldsymbol{P(cancer \mid +) = P(+\mid cancer)\, P(cancer) \,/\, P(+) = 0.21}$$

$$P(cancer) = 0.008 \qquad P(\neg cancer) = 0.992$$

Choosing hypotheses — MAP

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- Generally we want the most probable hypothesis given the training data
- Maximum A Posteriori (MAP): (最大后验假设) h_{MAP}

$$\begin{aligned} h_{\text{MAP}} &= \underset{h \in \mathcal{H}}{argmax}\, \mathrm{P}(h|D) \\ &= \underset{h \in \mathcal{H}}{argmax} \frac{\mathrm{P}(D|h)\mathrm{P}(h)}{\mathrm{P}(D)} \\ &= \underset{h \in \mathcal{H}}{argmax}\, \mathrm{P}(D|h)\mathrm{P}(h) \end{aligned}$$

An example – MAP

- Lab test result: +， has a particular form of cancer?
  - Correct positive: 98% (cancer, result +)
  - Correct negative: 97% (not cancer, result -)
  - Over entire population of people, only 0.008 have cancer

$$\boxed{\underset{h \in \mathcal{H}}{argmax}\, \mathrm{P}(D|h)\mathrm{P}(h)}$$

P(+| cancer) P(cancer) = 0.0078, P(+| ¬cancer) P(¬cancer) = 0.0298

$$h_{MAP} = \neg cancer$$

$$\begin{aligned} P(cancer) &= 0.008 & P(\neg cancer) &= 0.992 \\ P(+|cancer) &= 0.98 & P(-|cancer) &= 0.02 \\ P(+|\neg cancer) &= 0.03 & P(-|\neg cancer) &= 0.97 \end{aligned}$$

Brief Overview

- Bayes theorem
  - Use prior probability to inference posterior probability
- Max A Posterior, MAP, h_{MAP}， 极大后验假设

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

Choosing hypotheses — ML

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

The smart man always learns the most from experiences if he knows P(h).

- If we know nothing about hypotheses, or if we know all hypotheses have same probabilities, then MAP is Maximum Likelihood (h_{ML}极大似然假设)

$$h_{ML} = \arg\max_{h_i \in H} P(D|h_i)$$

A Note to Likelihood

- Likelihood is the hypothetical probability that an event which has already occurred would yield as a specific outcome. The concept differs from that of a probability in that a probability refers to the occurrence of future events, while a likelihood refers to past events with known outcomes.

-- from Concise Encyclopedia of Mathematics

Founders of MLE:

Maximum Likelihood Estimation

Gauss, Karl Friedrich (1777-1855)

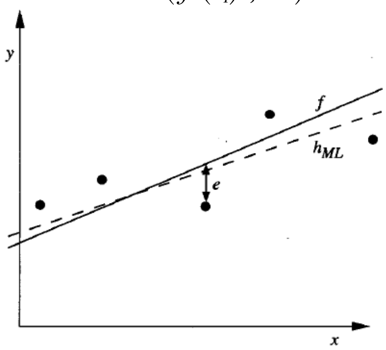
Fisher, Ronald Aylmer (1890-1962)

Maximum Likelihood & Least Square Error

- Training data: <x_i, d_i>
- $d_i = f(x_i) + e_i$,

  d_i : independent samples. $f(x_i)$: noise-free value of target function
- e_i : noise, independent random variables, normal distribution $N(0, \sigma^2)$

  → d_i : normal distribution $N(f(x_i), \sigma^2)$



Maximum Likelihood & Least Square Error

- Training data: <x_i, d_i>
- $d_i = f(x_i) + e_i$,

  d_i : independent samples. $f(x_i)$: noise-free value of target function
- e_i : noise, independent random variables, normal distribution $N(0, \sigma^2)$

  → d_i : normal distribution $N(f(x_i), \sigma^2)$

Normal distribution

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\, p(D|h)$$

Independent samples

$$= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} p(d_i|h)$$

$$= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}$$

Log function : monotonic

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2$$

Maximum Likelihood & Least Square Error

constant

$$
\begin{aligned}
h_{ML} &= \underset{h\in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \underset{h\in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \underset{h\in H}{\operatorname{argmax}} \sum_{i=1}^{m} -(d_i - h(x_i))^2 \\
&= \underset{h\in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2
\end{aligned}
$$

- independent random variables, normal distribution noise $N(0, \sigma^2)$, $\boldsymbol{h}_{\mathrm{ML}} = \boldsymbol{h}_{\mathrm{LSE}}$
- Please reading section 6.4 of the book *machine learning* (p164 in En. version).

Brief Overview

- Bayes theorem
  - Use prior probability to inference posterior probability
- Max A Posterior, MAP, h_{MAP}， 极大后验假设
- Maximum Likelihood, ML, h_{ML}, 极大似然假设
  - ML vs. LSE (Least Square Error)

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

Naïve Bayesian Classifier (朴素贝叶斯分类器)

- *Assume target function* $f : X \rightarrow V$, where each instance $x = (a_1, a_2, \ldots, a_n)$. Than most probable value of $f(x)$ is:

$$v_{\text{MAP}} = \underset{v_j \in V}{argmax}\, \mathrm{P}(x|v_j)\mathrm{P}(v_j)$$

- Naïve Bayes assumption:

Independent attributes

$$\mathrm{P}(x|v_j) = \mathrm{P}(a_1, a_2 \cdots a_n|v_j) = \prod_i \mathrm{P}(a_i|v_j)$$

- Naïve Bayes classifier:

$$\begin{aligned} v_{\text{NB}} &= \underset{v_j \in V}{argmax}\, \mathrm{P}(v_j) \prod_i \mathrm{P}(a_i|v_j) \\ &= \underset{v_j \in V}{argmax}\{\log \mathrm{P}(v_j) + \sum_i \log \mathrm{P}(a_i|v_j)\} \end{aligned}$$

If independent attribute condition is satisfied, then $v_{MAP} = v_{NB}$

Example: Word Sense Disambiguation(词义消歧)

- e.g. fly =? bank = ?
- To the word *w*, using context *c* to disambiguation
  - e.g. A fly flies into the kitchen while he fry the chicken.
  - Context *c:* a groups of words w_i around *w* (-- features / attributes)
  - s_i: the i^{th} sense of the word *w* (-- output label)
- Naïve Bayes assumption: $\mathrm{P}(c|s_k) = \prod_{w_i \in c} \mathrm{P}(w_i|s_k)$
- Bayes decision:

$$s = \underset{s_k}{argmax}\{\log \mathrm{P}(s_k) + \sum_{w_i \in c} \log \mathrm{P}(w_i|s_k)\}$$

where: $\mathrm{P}(w_i|s_k) = \dfrac{C(w_i, s_k)}{C(s_k)}$ $\mathrm{P}(s_k) = \dfrac{C(s_k)}{C(w)}$

Brief Overview

- Bayes theorem
  - Use prior probability to inference posterior probability
- Max A Posterior, MAP, h_{MAP}， 极大后验假设
- Maximum Likelihood, ML, h_{ML}, 极大似然假设
  - ML vs. LSE (Least Square Error)
- Naïve Bayes, NB, 朴素贝叶斯
  - Independent attribute / feature assumption
  - NB vs. MAP

$$P(h \mid D) = \frac{P(D \mid h) P(h)}{P(D)}$$

MDL (Minimum Description Length)

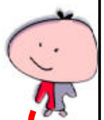
- Occam's razor:
  - prefer the shortest hypothesis
- MDL:
  - prefer the hypothesis h that minimizes:

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

where $L_C(x)$ is the description length of x under encoding C

Explanation to MDL（information theory）

- Code design for randomly send messages
  - the probability to message i is p_i
- What's the optimal (shortest expected coding length) code?
  - Assign shorter codes to messages that are more probable
  - The optimal code for message i is -$\log_2 p$ bits [Shannon & Weaver 1949]
- -$\log_2 p\ (h)$: length of h under optimal code C
- -$\log_2 p\ (D\,|\,h)$: length of D given h under optimal code C



MDL and MAP

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

Another Explanation to MDL

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

- length of h, and the cost of encoding data given h
  - Suppose the sequence of instances is already known to both transmitter and receiver
  - No misclassification: no need to transmit any information given h
  - Some are misclassified by h: need transmit
  - 1. which example is wrong?

    -- at most $\log_2 m$ (m: # of instances)
  - 2. the correct classification?

    -- at most $\log_2 k$ (k: # of classes)

Explanation to MDL

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

- 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

Overview

- Bayes theorem
  - Use prior probability to inference posterior probability
- Max A Posterior, MAP, h_{MAP} (极大后验假设)
- Maximum Likelihood, ML, h_{ML} (极大似然假设)
  - ML vs. LSE (Least Square Error)
- Naïve Bayes, NB, 朴素贝叶斯
  - Independent assumption
  - NB vs. MAP
- Maximum description length, MDL (最小描述长度)
  - Tradeoff: hypothesis complexity vs. errors by h
  - MDL vs. MAP

$$P(h \mid D) = \frac{P(D \mid h) P(h)}{P(D)}$$

Overview: MAP_MDL_ML_NB