CHEAT SHEET INF283 *Machine Learning – Tommy Odland – Last edit: November 30, 2016*

▷ Introduction

Nomenclature

Symbol	Meaning
h	Hypothesis (function)
H	Hypothesis space (function space)
x	Instance (data)
X	Instance space (data space)
c	Target concept (function)
$\wedge$	Conjunction
$\vee$	Disjunction

A motivating example

Consider the following dataset:

#	AGE	HAIR	HEIGHT	SEX
1	27	long	162	F
2	32	short	181	M
3	15	short	175	M
⋮	⋮	⋮	⋮	⋮

- Attributes – AGE , HAIR and HEIGHT .
- Attributes have possible values that are categorical, discrete or real. Here AGE is discrete, HAIR is categorical and HEIGHT is real.
- The target attribute is SEX .
  - If real/discrete target concept → Regression problem
  - If categorical target concept → Classification problem

▷ Machine learning

- Inductive and deductive reasoning
  - Inductive: Specific → general
  - Deductive: General → specific
- Supervised learning – a learner is presented with samples and the target value.
- Unsupervised learning – a learner is presented with samples, but no target value.
- Inductive bias: the set of assumptions that the learner uses to predict outputs given inputs that it has not previously encountered.
- Inductive bias = restriction bias + preference bias. Restriction bias is related to restrictions in the hypothesis space, preference bias is related to preferences in the hypothesis space.
- Good error functions are (1) differentiable (2) increase monotonically on both sides. A good error function is $E(x_i) = \frac{1}{2}\sum_i (h(x_i) - c(x_i))^2$.
- Well posed problem consists of a task T, a performance measure P and a training experience E.
- Data preprocessing is very important in practice. For algorithms such as neural networks and support vector machines, normalizing is crucial.

▷ Concept learning

- Concept learning: Inferring boolean valued target concept c.
- The FIND-S algorithm starts with most specific hypothesis $\langle \emptyset, \emptyset, ..., \emptyset \rangle$ and makes it more general as data is encountered. Finds a maximally specific hypothesis.
- The CANDIDATE ELIMINATION algorithm finds all hypotheses consistent with the data by incrementing two boundaries G_i and S_i as samples are encountered.
- Both of the above algorithms perform poorly on noisy data.

▷ Decision trees

Information theory

- Information entropy is defined as

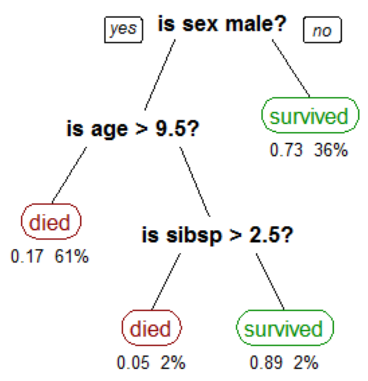
$$\mathrm{E}(S) := -\sum_{c \in V} p_c \log_2 p_c$$

  where $c \in V$ are the categorical variables.
- Information entropy measures impurity, or uncertainty. A coin toss has high entropy.
- The definition of information gain for A with respect to S is:

$$\mathrm{Gain}(S, A) := \mathrm{E}(S) - \sum_{v \in \mathrm{values}(A)} \frac{|S_v|}{|S|} \mathrm{E}(S_v)$$

- Information gain is of little use when an attribute has many distinct values. This problem is overcome by normalizing $\mathrm{Gain}(S, A)$ by diving by $\mathrm{E}(A)$. This normalizes with respect to the entropy of the attribute A.
- Entropy may be interpreted as the expected number of bits to optimally encode elements uniformly draw from S for transmission.

Decision trees



- Attributes must be categorical. Explicitly models conditionality between variables.
- Less confidence as we go down, because decision is made based on fewer samples.
- Appropriate classification problems have discrete attribute values. Decision trees are robust against noisy data. It may overfit, the overfitting problem is mitigated by either (1) early stopping or (2) post-pruning. Post-pruning is more successful in practice.
- Reduced error pruning – iteratively remove nodes in the tree if removal increases accuracy over validation set. Stop when no nodes can be removed without decreasing the accuracy.
- The inductive bias is a preference for short trees, and trees which place high information gain at the top of the tree. The ID3 algorithm is greedy, and in general not optimal.

▷ Performance evaluation

- A confusion matrix is given below:

		Predicted value	
		T	F
True value	T	TP	FN
	F	FP	TN

- The accuracy is $\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$
- The sensitivity is $\frac{\text{TP}}{\text{TP} + \text{FN}}$ (true positive)
- The specificity is $\frac{\text{TN}}{\text{TN} + \text{FP}}$ (true negative)
- The precision is $\frac{\text{TP}}{\text{TP} + \text{FP}}$ and the recall is $\frac{\text{TP}}{\text{TP} + \text{FN}}$
  - The F-measure combines precision and recall.
- k-FOLD CROSS VALIDATION – Split the dataset into k disjoint, equal sized sets, train and test on each partition and average the results.
- $\text{error}_S(h)$ (sample error) can be used to set up a confidence interval for $\text{error}_{\mathcal{D}}(h)$ (true error).

▷ Bayesian learning

- Bayes theorem is

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

  where $P(h|D)$ is the posterior, $P(D|h)$ is the likelihood and $P(h)$ is the prior. h is a hypothesis (e.g. "the email is SPAM") and D is data (e.g. "the words "sex" and "free" are in the email").
- The maximum a posterior hypothesis is

$$h_{\text{MAP}} = \underset{h \in H}{\text{argmax}}\ P(h|D)$$

  Note that h_{MAP} does not always lead to the "most probable classification."
- The maximum likelihood hypothesis is

$$h_{\text{ML}} = \underset{h \in H}{\text{argmax}}\ P(D|h)$$

- A consistent learner is one which commits no errors over the training data.
- The minimum description length principle is a formalization of Occam's razor. Describe data using the classifier and misclassifications requiring the least bits to transfer.

Naive Bayes classifier

- The naive assumption is that attribute values are conditionally independent (do not influence each other), leading to

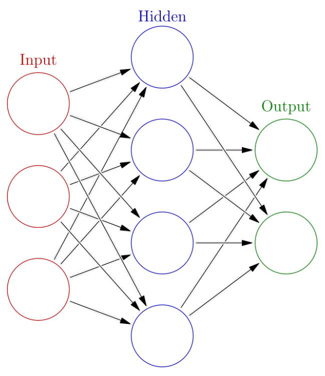
$$\begin{aligned} &\text{argmax}\ P(v_j|a_1, a_2, ..., a_n) = && \text{(Bayes thm)} \\ &\text{argmax}\ P(v_j)P(a_1, a_2, ..., a_n|v_j) \approx && \text{(naivety)} \\ &\underset{v_j \in V}{\text{argmax}}\ P(v_j) \prod_i P(a_i|v_j) \end{aligned}$$

- Classifies according to

$$v_{\text{NB}} = \underset{v \in V}{\text{argmax}}\ P(v) \prod_{i=1}^{n} P(a_i|v)$$

  where $v \in V$ are the possibles values for the target attribute. Attributes must be categorical.
- A problem arises if $P(a_i|v) = 0$ for some a_i, because 0's influence classification too strongly. Two solutions are
  - Add artificial count of 1 (numerator and denominator).
  - Design a prior expectation using an m-estimate, $\frac{n_c + mp}{n+m}$, where n and n_c are real samples and m is a weight, p is a prior estimate. The second term in the equation are the virtual samples.

▷ Artificial Neural Networks



- Output of an artificial neuron is given by

$$y = f\left(\Sigma_i w_i x_i\right)$$

  where w_i are weights and x_i are inputs.
- Some activation functions f are (1) the thresholded perceptron (step function), (2) the unthresholded perceptron (linear function), (3) the sigmoid function and (4) the tanh (hyperbolic tangent) function.
- The universal approximation theorem states that with 1 hidden layer, any continuous function can be approximated.
- Gradient descent algorithm for training
  - Use the following error estimate

$$E(\vec{w}) = \frac{1}{2} \underbrace{\sum_{d \in D}}_{\text{All data}} \underbrace{(t_d - o_d)^2}_{\text{True minus predicted}}$$

    where $o_d = f(\vec{w} \cdot \vec{x}_d)$ and $f' = (1-f)f$ if f is the sigmoid.
  - Gradient descent – Consider all $d \in D$ before moving along error surface.
  - Stochastic gradient descent – One training sample d at a time.
  - If $f(x) = 1/(1+e^{-x})$, then $f'(x) = (1-f(x))\,f(x)$. The update rules are $\Delta w = -\epsilon \frac{\partial E}{\partial w}$, e.g. negative of gradient. For output neurons

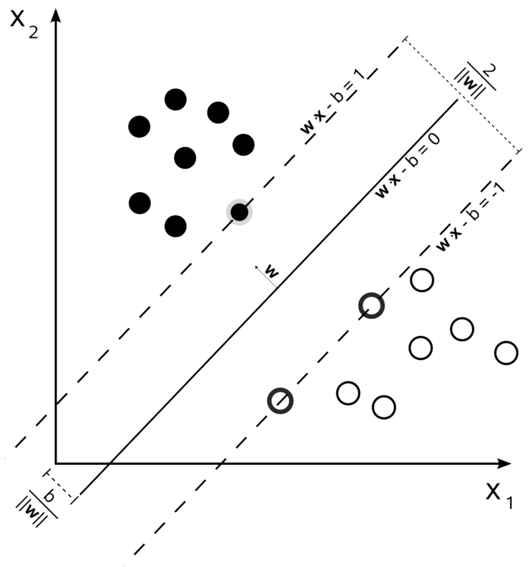
$$\frac{\partial E}{\partial w_j} = (O - t) \cdot f'(o) \cdot H_j$$

    and for hidden neurons (one hidden layer)

$$\frac{\partial E}{\partial v_{jk}} = (O - t) \cdot f'(o) \cdot w_j \cdot g'(h_j) \cdot I_k$$

- To alleviate problems with local minima, one or several of the following techniques may be applied: momentum term, stochastic gradient descent, train several networks.
- Overfitting may be controlled by monitoring error with respect to test data. When error as a function of model complexity on test data is minimal, the model is probably good.
- Strengths include extreme approximation power, few prior assumptions. Weaknesses include overfitting, computational expense and interpretation difficulties.

▷ Support Vector Machines



- Classification in feature space using hyperplanes.
- The optimization problem (for linearly separable data) is

$$\begin{aligned} &\min \quad \frac{1}{2}\|\vec{w}\| \\ &\text{s.t.} \quad y_i\left(\vec{w}x_i - b\right) \geq 1 \;\forall\; i \end{aligned}$$

  where $\vec{w}$ is a vector and b is a threshold.
- Advantages: easy training, no local optima, scales well, different types of input works. Disadvantages: a good kernel is often needed.

▷ Reinforcement learning

- Autonomous agent vs. environment. Think backgammon, pac-man, chess, robot, etc.
- $Q(s, a)$ maps a state s and an action a to the maximum cumulative reward.
- The maximum cumulative reward V is

$$V(s_t) := \sum_{i=0}^{\infty} \gamma^i r_{i+1}$$

  where s_t is the state in which the first action is performed, γ is a discount factor and r_{i+1} are the rewards.
- Other sensible reward systems are finite horizon rewards and average reward.

- The Q-learning equation is

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \left[r(s,a) + \gamma V * (\delta(s,a)) \right]$$

where the term inside the brackets is replaced by $Q(s,a)$. This allows us to learn the optimal action policy $\pi^*(s)$ without knowing $r(s,a)$ and $\delta(s,a)$ explicitly.
- The Q-learning algorithm is
  (a) Select an action a, execute it
  (b) Receive a reward r
  (c) Observe a new state s'
  (d) Update $\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$
- Experimentation can be added to avoid overcommitting to early actions. Experimentation factor can decline with time.
- Can store entire episode to train on, not just action that led to the reward.
- Q-learning is similar to dynamic programming (DP). But in DP $r(s,a)$ and $\delta(s,a)$ are known.

▷ Computational learning theory

- PAC-learnability – Require that learner can probably learn a hypothesis which is approximately correct. In other words, with probability at least $(1-\delta)$ output a $h \in H$ such that $\text{error}_{\mathcal{D}}(h) \leq \epsilon$, in time polynomial in $1/\epsilon$, $1/\delta$, $|X|$ and $\text{size}(C)$.
  - To show PAC-learnability, show that (1) each target concept can be learned in polynomial samples and (2) processing time per sample is polynomial.
- Sample complexity asks the question of how many samples that are needed to learn the target concept c.
- A dichotomy splits X in two ways. H shatters X if every possible dichotomy in X is expressible by H.
- The VC-dimension (Vapnik-Chernovenkis dimension) of H is the size of the largest finite subset of X shattered by H.
  - Example: H are decision hyperplanes. Shatters (non-colinear) points when $|X| \leq 3$, therefore $\text{VC}(H) = 3$.
  - Large VC $\Leftrightarrow$ Expressible hypothesis space $\Leftrightarrow$ Little inductive bias
- The mistake bound model: receive $x \rightarrow$ predict $c(x) \rightarrow$ receive true value. How many mistakes are needed before learning exactly?
- Weighted majority takes the weighted majority of several possible hypotheses $h \in H$. Makes at most $2.4\,(k + \log_2(n))$ mistakes.

▷ Unsupervised learning

Market basket analysis

- An itemset is a collection of items, such as {bread, jam, milk}.
- An association rule is a rule of the form {bread, milk} → {jam}.
- The goal of market basket analysis is to discover meaningful association rules.
- The APRIORI -algorithm builds itemsets from small to large. If the support (frequency) of a collection is below a threshold, it is excluded from all further iterations of the algorithm. From the created itemsets, association rules are created if the confidence (conditional probability) is over a threshold.
- Three equations used are

$$\begin{aligned} \text{support}(X) &= \frac{\text{count}(X)}{N} \\ \text{confidence}(X \rightarrow Y) &= \frac{\text{support}(X,Y)}{\text{support}(X)} \\ \text{lift } (X \rightarrow Y) &= \frac{\text{confidence}(X \rightarrow Y)}{\text{support}(Y)} \end{aligned}$$

support is probability, confidence is conditional probability, lift is conditional probability divided by non-conditional probability.

k-means

- Each observation X belongs to the closest centroid, so that $\text{assign}(x) = \text{argmin}_y\, D(x,y)$, where $D(\cdot,\cdot)$ is a distance function and y is the centroid.
- In each iteration, the new center of y is the mean of all observations in that cluster.
- Advantages: Simple, flexible and decent. Disadvantages: random chance, requires guess of k.
- Misc: play around with k, run several times due to random nature.

▷ References

- Tom M. Mitchell. Machine Learning. McGraw-Hill Series in Computer Science, Artificial Intelligence. New York: McGraw-Hill, 1997.
  - Thorough, good book. Focus on theory and applications. Requires knowledge of mathematics to enjoy.
- Lantz, Brett. Machine Learning with R. Olton: Packt Publishing Ltd, 01.
  - More hands-on than Mitchell. Focuses on application and experimentation using the programming language R, rather than theory.