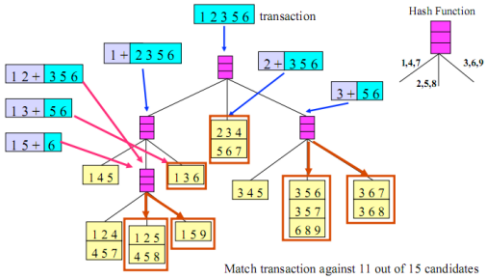
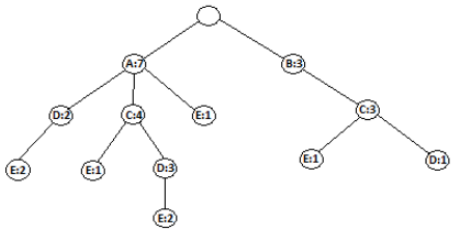
ASSOTIATION RULES. IS-items>=1. **Support count**-frequency of occurrence of an IS in transactio s. **Support**-fraction of transactions that conain an IS. **Frequent IS**-IS that has support>=minsup. **Association rule**-implication X->Y where X&Y=IS. **Rule support**-fraction of transactions that contain X&Y (s=sc(X&Y)/|Trans|). **Rule confidence (P)**-how often Y appears in transactions with X (c=sc(X&Y)/sc(X)).

Generation strategies: reduce candidates/transactions/comparisons. Apriori (-candidates): If IS is freq, then all sub-ISs must also be freq. So prune all ISs that are infreq (minsup) and their sub-ISs. Hash (-comparisons): Store possible candidates in hash structure. Compare against hashed buckets. Ie, compare only to bucket that matches condition.



IS is **maximal** if none of it's descendants are frequent and **closed** if none of it's descendants have the same support. FP-tree: transactions in a tree w/ supcount. Freq ISs form „chain". Chain members form freq ISs. Ie: {A,E} is freq in graph (lefthand chain)



ECLAT: store list of trans IDs per item (_-layout). Get support by intersecting their lists. Interestingness:

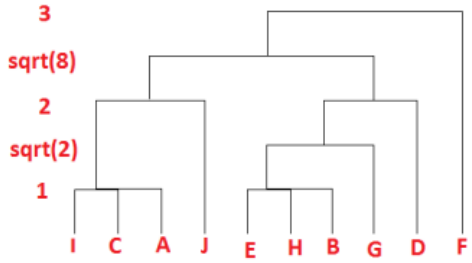
	Y	-Y	
X	F11	F10	F1+
-X	F01	F00	F0+
	F+1	F+0	|T|

Statistical independance: P(S^B)=P(S)xP(B), independant, > positively < negatively correlated. **Lift**=P(Y|X)/P(Y), **Interest** =P(X,Y)/P(X)P(Y),**PS**=P(X,Y)-P(X)P(Y),ϕ **-coef**=P(X,Y)-P(X)P(Y)/sqrt(P(X)[1-P(X)]P(Y)[1-P(Y)]).

CLUSTERING. Identify similar objects in a set. Hierarchical all-against-all distance matrix. Identify closest and group. Recalc and repeat till 1 cluster.

Inter-cluster similarity: MIN, MAX, Group avg, dist. Between centroids.

UPGMA (average distance with weighted components – computationally mor eintensive, less influenced by outliers): $D\left(C_k,\left(C_i \cup C_j\right)\right) = \frac{|C_i|}{|C_i|+|C_j|} D(C_k,C_i) + \frac{|C_j|}{|C_i|+|C_j|} D(C_k,C_i)$, WPGMA(computationally less intensive, use when clusters proportionally similar and very little outliers): $D\left(C_k,\left(C_i \cup C_j\right)\right) = \frac{1}{2}\left(D(C_k,C_i) + D(C_k,C_i)\right)$, single linkage: $D\left(C_k,\left(C_i \cup C_j\right)\right) = \min\left(D(C_k,C_i), D(C_k,C_i)\right)$



Fast cluster: estimate dists, use pivots. Partial info. K-means: Pick k random centres C (pref. Existing elements). Assign objects to centers. Move Cs to gravity centers. Repeat till no new assignments.

SOMs: input, weight vectors, nodes. Weight vectors compared to input. Find BMU, adjust weights of nodes by a factor of distance from BMU and the learning rate (both decrease over time) and difference between current weights and inputs.

DBSCAN: **density**=number of points in specified radius (r=Eps). Core point (>minpts in eps), border point (<minpts, in neighborhood), noise point.

For all core pts: if(nolabel) {cur_label+1, this.label = cur_label} for(i in all pts in eps) {if no label then i.label = cur_label}

PREPROCESSING. Data cleaning (remove outliers, fill in values, outlier detection), integration (multiple data points into one), transformation (normalization, aggregation), reduction (reduce volume w/o impacting info), discretization.

Median (middle value or average of 2 middle values), mode (most freq value).

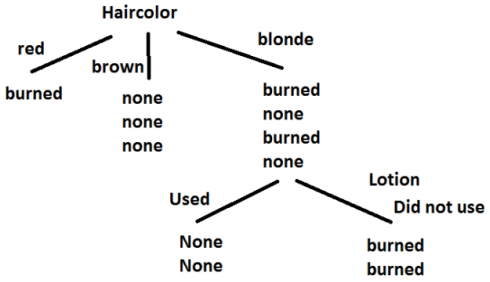
Histograms. **Kernel density** $f(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h}).\ K\left(\frac{x-x_i}{h}\right) = \frac{1}{\sqrt{2}} e^{-\left(\frac{x-k_j}{2h^2}\right)^2}$. H is the bandwidth and K is some kernel (weighting function).

Box plot: ends are first and third quartiles, median is a line in the box and min/max are shown as lines extending out of the box.

Quantile plot: shows how much % of the data is below or equal to value. **Scatter plot**: uses values as coordinates and plots them as points on plane. **Loess curve**: smooth curve on the scatter plot for improving perception of dependence. Missing data: fill in, ignore. Fill w/ the attribute mean, mean for same class, constant like unknown. Noisy data: sort into (equal-freq or equal data sample) bins, then smooth by bin attributes (mean, median, boundaries), reression functions, clustering (outlier removal), check by hand.

Discretisation: ie divide data into intervals. **3-4-5 rule**: if interval covers 3,6,7,9 distinct values, partition range into 3 equiwidth intervals. If 2,4,8, then 4 and 1,5,10 then 5 intervals.

SUPERVISED LEARNING: Let X and Y be IS and let there be a set of training samples {(x1,y1),(x2,y2),..|xEX,yEY}. Find function f: X->Y generalizing functional relationship present in the data. **TP rate** = true positives/positive examples. **FP rate** = FP/FE. **Precision** = TP/positives. **Recall** = TP/positive examples. **F-Measure** = 2*(P*R)/(P+R). **ROC area** ~Pr(s(false)<s(true)). Validate algorithm on different dataset. ID3: split into pieces by attribute. Select split by informativeness. H(p) = $-\sum_i p_i log_2 p_i$. Information gain=H(p_old)-H(p_new). Repeat on subnodes that have multiple values for decision attribute.



Naive Bayes Classifier: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$. Basically, multiply the prior likelyhood of an object being of class A (example: 2x green objects than red = 2x more chance of new object being green) to the likelyhood of an object being of class A (example: there are 2x more red objects in vicinity of new object). Linear regression: f(x) that has least possible sum of error squares. K-nearest neighbors: Given an object, find k instances from the training sample closest to it and predict the majority class of those k instances.

TEXT MINING. Automatic discovery of potentially useful, previously unknown information from textual resources. Attributes: synonymy (diff w, same meaning), homonomy (same w, diff meaning), polysemy (same w, multiple related meanings), hyponomy (one w means subclass of another). Wordnets: grouped and linked words (cat->mammal, room->house).

HOMEWORKS. Some key points (abstract): Non-overlapping patterns are more likely to emerge first in a statistical series. The inverse of statistical accuracy of a tast is the likelyhood of a false positive. Do not assume independence of events. Graph types and markers are important in data visualization. Wrong ones could misrepresent data. Perception is relative rather then absolute: use common scales and few colors. Popout (objects differing from the norm are easier to distinguish) helps distinguish data, but only in one channel. 3D only in case of 3D data.