DECISION TREE

Conditions to stop partitioning: {No sample; All same class; No remaining attr -> Majority Voting}

1. Info. Gain [ID3, C4.5] {IG(A) = Ent(D) - Ent_A(D)} [biased towards multivalued attr]; ***2. Split Info*** [C4.5; normalized IG (→lots of values)] {*Max*: ***Gain Ratio*** = Gain(A) / SI_A(D); $SI_A(D) = -\sum \frac{|D_j|}{|D|} * \log_2 \frac{|D_j|}{|D|}$} [Prefer unbalanced splits]; ***3. Gini Index*** [CART, IBM] {$GI(D) = 1 - \sum p_j^2$; $GI_A(D) = \sum \frac{|D_i|}{|D|} GI(D_i)$; *Max*: ***Reduction in Impurity*** $\Delta GI(A) = GI(D) - GI_A(D)$} [CON: Cannot large #class] [Favor equal-sized partition & purity in both partitions]

CHAID [X^2 test]; *C-SEP* [@some cases: better than IG & GI]; *G-statistic* [close appx. To X^2 distribu.]; *Minimal Description Length principle*; *Multivariate Splits* [多个变量的组合] {CART}

OVERFITTING

Prepruning (衡量标准低于 threshold 则不分);
Postpruning (从 fully grown tree 里去除 branch).

ENHANCEMENTS TO BASIC D.T. INDUCTION

{Allow for continuous-valued attributes (动态定义新的离散变量); Handle missing attribute values (用[最常见的取值/各取值的概率]赋值); Attribute construction (fragmentation, repetition, replication)}

LARGE DATABASE

RainForest {Attr: AttrValClass-set; Node: AVC-group}; ***BOAT (Bootstrapped Optimistic Algorithm for Tree Construction)*** [2 scans of DB] {Use *Boot-strapping* to create smaller subsets; Each subset used to build a tree; Trees to construct new tree};

Naïve Bayes Classifier

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i)$$

{If 离散: $\{P(x_k|C_i) = \#x_k / |C_{i,D}|\}$;
If 连续: $P(x_k|C_i) = g(x_k, \mu_{Ci}, \sigma_{Ci})\}$} $g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

*[Each conditional prob. be non-zero {**Laplacian correction/estimator**: 每个类加一}]*

RULE-BASED CLASSIFICATION

[Measurement: *Coverage* = #cover / |D|, *Accuracy* = #correct / #cover] {Present using IF-THEN rules}

Conflict Resolution

1. Size ordering (条件苛刻↑); ***2. Class-based ordering*** (*prevalence of misclassification cost* per class↓); ***3. Rule-based ordering*** / ***Decision List*** (按衡量标准整理成 list)

Rule Induction

Sequential Covering Method [对每个类 Ci: sequentially 学] {一次学一条: {开始空集; greedy depth-first strategy 选择最提升 rule quality (*FOIL, AQ, CN2, RIPPER*)的;}; 每学一条, 移除其覆盖的数据; 重复直到结束条件满足;}

$$FOIL\_Gain = pos' \times (\log_2 \frac{pos'}{pos'+neg'} - \log_2 \frac{pos}{pos+neg})$$

FG @ *FOIL, RIPPER*
{Prune if FOIL_Prune higher} $FOIL\_Prune(R) = \frac{pos-neg}{pos+neg}$

MODEL EVALUATION AND SELECTION

Estimating Accuracy

1. Hold-out method [随机分成两个子集] {*Random Sampling* [重复 k 次]}; ***2. Cross-Validation*** (k-fold) {*Stratified* ~ [每 fold 的 label 分布与总集一样]}; ***3. Bootstrap*** [适合小数据集; 有放回; 重复 k 次] {*.632* ~ (有放回地取样|D|次, 63.2%的 data 会出现在 train)}

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

Confusion Matrix [表头: Actual \ Predict]
{Accuracy = (TP + TN) / ALL, Error Rate = 1 – Accuracy = (FP + FN) / ALL}
@Machine Learning: [Sensitivity=TP / P; Specificity = TN / N];
@Info Retrieval: [Precision (Exactness)= TP / (TP + FP); Recall (Completeness)= TP / (TP + FN)].

F Measure (F-score): (调和平均 Preci. & Reca.)
{***F1 Measure*** (Balanced F-score): $\beta = 1$} $F = \frac{1}{\alpha \cdot \frac{1}{P} + (1-\alpha)\cdot\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$

Comparing Classifiers

1. Confidence intervals [t-dist. w/ d-1 DOF; Use t-test] {Null Hypo. $M_1 = M_2$; (1-Tail) sig. Level (e.g. 5%); Conf. Limit z = sig/2 (2-Tail e.g. 2.5%); if t > z || t < -z, reject Null};

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}} \quad var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}},$$

两个 test set, DOF 选小的

$$var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}\left[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))\right]^2$$

k: #sample 一个 test set, 两个模型

2. Cost-benefit analysis & Receiver Operating Characteristics Curves [*Tradeoff*: true pos.% VS false pos.%; Area under ~: accu.; 凹为好] {把 test tuple 按属于 positive class 的可能性降序排列; 用每个 tuple 的可能性作为分割全部 test 的标准(比它高则计入 P,反之 N,计算 TPR & FPR); Convex hull 作图};

[Accuracy, Speed, Robust, Scalable, Interpretable]

ENSEMBLE METHODS

1. Bagging (averaging) {***Random Forest*** [每个树都在每个 node 随机选择 attr 生成的; robust > Adaboost] {*Forest-RI* (ran. input sel.); *Forest-RC* (ran. Linear comb.)}}; ***2. Boosting*** (加权投票) {Adaboost (α=log[(1-e)/e])}; ***3. Ensemble*** (hetero.);

CLASS-IMBALANCED DATASETS

1. Over-sampling 少的; 2. Under-sampling 多的; 3. Threshold-moving 允许很少; 4. Ensemble methods.

BAYESIAN BELIEF NETWORKS

[A structure (DAG) + A set of Cond. Prob. Tables]
{1. Subjective construction;
2. Synthesis from other specifications;
3. Learning from data} $P(x_1,\dots,x_n) = \prod_{i=1}^{n} P(x_i | Parents(x_i))$

SCENARIOS: {结构已知+全变量可见: 计算 CPTs; 结构已知+部分变量可见: gradient descent; 结构未知+全变量可见: 查找 model space, 重建 network topology; 未知结构+全变量不可见: no good algorithm;}

NEURAL NETWORK

[Feed-forward; 非线性回归; Back-propagation to min. MSE]

DISCRIMINATIVE CLASSIFIER

[Accuracy high; Robust; Fast evaluation] VS [Long training time; Difficult to understand; Hard to incorporate domain knowledge]

SUPPORT VECTOR MACHINES

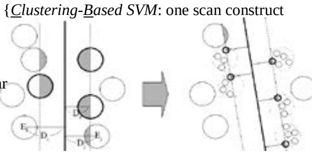
[constrained (convex) quadratic optimization] [complexity: #Sup. Vect.] [CON: Not scalable to #data] *Kernels*= $\Phi(X_i)\ \Phi(X_j)$

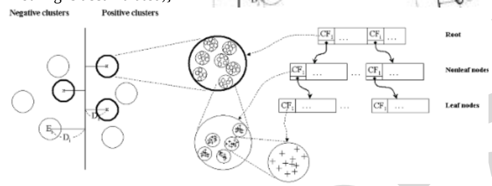
Polynomial kernel of degree h: $K(X_i, X_j) = (X_i \cdot X_j + 1)^h$

Gaussian radial basis function kernel: $K(X_i, X_j) = e^{-\|X_i - X_j\|^2/2\sigma^2}$

Sigmoid kernel: $K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$

Scaling SVM by Hierarchical Micro-Clustering
[PRO: Scalable to #data] {*Clustering-Based SVM*: one scan construct 2 ***CF-trees****; Train a SVM from the centroids of root entries; De-cluster the entries near the boundary into the next level; Repeat until nothing is accumulated;}





*__CF-Tree__ (Clustering Feature) {De-cluster only the cluster Ei s.t. $D_i - R_i < D_s$ (Di: 边界到中央的距离; Ri: Ei 的半径; Ds: margin); 仅 de-cluster 其 sub-clusters 可能成为 *Support Cluster* 的 cluster (*Support Cluster*: whose centroid is s.v.)}

PATTERN-BASED CLASSIFICATION

(***Associative or ~***; FP-mining + Classification) [Feature construction (higher order, compact, discriminative); Complex data modeling (graphs, sequences, semi/un-structured data)]

1. Classification Based on Associations [accurate > C4.5: explore high conf. among multiple attr] {Mine high-conf., high-sup. class asso. Rules; "Conjunctions of attr pairs→class label": (p1^…^pn→predict as C); 按 conf. & sup.降序排列 rules;};

2. Classification based on Multiple Association Rules [Model construction efficiency↑; classification accuracy↑] {插入 rule 到 tree时 rule pruning: (若 R1 的前提比 R2 更一般且 conf 更大, prune R2); 若只有一个 rule 满足则 apply, 若 rule set S 都满足: {根据 class label 给 S 分组; 使用 weighted X^2 measure 找到最强的一组 rule; 取其 label}};

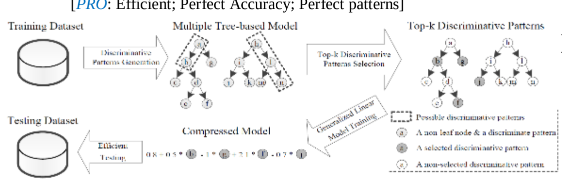
3. Discriminative Pattern-based Classification [] {Feature construction by *frequent itemset mining*; Feature selection (using ****Maximal Marginal Relevance***): {select discriminative features (relevant but minimally similar to previously selected ones); remove redundant or closely correlated ones}; learn a general classifier (SVM, C4.5)};
[Info. Gain (Discriminative Power) of k-itemsets > single features; IG upper bound monotonously increase with pattern freq;]

MINING CONCISE SET OF DISCRIMINATIVE PATTERNS

1. FP mining + filtering [Expensive; Large model]

2. DDPMine: Direct Discriminative Pattern Mining
[PRO: Efficient; Direct mining]

3. DPClass: Discriminative Pattern-based Classification
[PRO: Efficient; Perfect Accuracy; Perfect patterns]



{Adopt every prefix path in a random forest as a candidate pattern; Run top-k pattern selection based on training data; Train a generalized linear model (e.g. logistic regress.) based on "bag-of-patterns"}
Pattern Selection: {Forward > LASSO}

Lazy Learning (Instance-based Learning) [训练省时, 预测费时; 有效使用 richer hypothesis space] {只存储/简单处理训练集, 直到 given a test tuple}

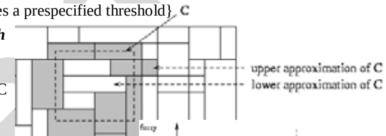
1. K-Nearest Neighbor [Real-valued prediction] [PRO: noise 鲁棒] [CON: Curse of dim. {Axes stretch or elimination of the least relevant attr}; Weight the contribution of each neighbor {w = 1 / d(x_q, x_i)^2}]
2. Locally Weighted Regression
3. Case-Based Reasoning [Use database of problem sol. to solve new ones {Customer service, Legal ruling}] {Instances 用 rich symbolic description (function graph) 表示; 查找相似的 case; Tight coupling between case retrieval, knowledge-based reasoning and problem solving} [CON: Find good similarity metric; Indexing; Backtracking & adapting to additional cases]

Eager Learning [commit to a single hypothesis that covers entire instance space] {先构造分类模型}

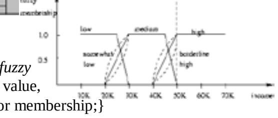
OTHER CLASSIFICATION MODELS

1. Genetic Algorithms [PRO: Easy parallelizable] [CON: Slow] {Initial population consisting 随机生成的用 a string of bits (an attr's #value)表示的 rules; Form a new population consisting fittest (by accuracy) rules and offspring (gen. by crossover and mutation); Loop until population satisfies a prespecified threshold}

2. Rough Set Approach
[To appx. define equivalent classes] {Rough set for a class C is appx. by [a lower appx. (IN C) & an upper appx. (不能认为 NOT IN C)]}



3. Fuzzy Set Approach {Attr assigned a *fuzzy membership value* (0~1) to each discrete value, 其和不必为 1; Rule contributes a vote for membership;}



MULTICLASS CLASSIFICATION

1. One-Vs-All [m] ***2. All-Vs-All*** [mC2; Err-sensitive]
Error-Correcting Codes for ~ {argmin (H(X, Ci))}
[Hamming distance; Correct up to (h-1)/2 1-bit error]

SEMI-SUPERVISED CLASSIFICATION

1. Self-training [PRO: 容易理解] [CON: 强化错误] {用已标记数据训练分类器; 用它分类未标记}; ***2. Co-training*** {每个 tuple 互斥的 features set 来训练 f1, f2; 预测未标记 X; If ***f1*** 对 X 最自信, add it to ***f2*** set}

ADDITIONAL TOPICS

Active Learning [PRO: 用最少的 labeled 得到高准确率]
[用 *learning curve 衡量*] {U: 未标记 data pool; Use query func 从 U 小心选择 tuples, 让 oracle (a human annotator) 标记}

Transfer Learning [从多个 src task 里提取知识, 应用到 target task] {* ***TrAdaBoost*** {Assume src & target data: 同样 attr, 不同分布; 只要求标记少量 target data}

CLUSTERING

[High intra-class similarity: cohesive in Cs, Low inter-class similarity: distinctive between Cs]

Considerations: [Partitioning criteria {single level < hierarchical}; Separation of clusters {exclusive VS non-exclusive}; *Similarity measure* {Distance-based (Euclid, road network, vector) VS *Connectivity-based* (density, contiguity)}; Clustering space (full-space @ low dim. VS subspace @ high dim.)]

Requirements & Challenges: [Quality: {different attr types; Discover arbitrary shape; Noise}; Scalability; Constraint-based clustering; Interpretability, usability]

Categorization: [Technique ~; Data type ~; Additional insight ~: {Visual insights; Semi-supervised; Multi-view (不同视角); Ensemble-based (鲁棒); Validation-based (case study, measures, labels)}]

TYPICAL CLUSTERING METHODOLOGIES

1. Distance-based [Partitioning; Hierarchical];
2. Density-based {Data space explored @ high-level of granularity; then put dense regions together}; ***Grid-based methods*** {Individual regions formed into a grid-like structure};
3. Probabilistic & Generative models {Assume a specific form of generative model (mixture of Gaussians); Parameter estimated with EM; estimate generative probabilities of data points}.
4. High-dimensional clustering [Subspace cluster: {Bottom-up; Top-down; Corr.-based; δ-cluster}; Dimensionality reduction: {*Probabilistic Latent Semantic Indexing*, LDA; *Nonnegative Matrix Factorization* {A (word freq.) non-neg. mat. appx. factorized two non-neg. low-rank matrices}; Spectral clustering (spectrum of the similarity matrix)}]

Centroid: Radius: Diameter:

$$\vec{x}_0 = \frac{\sum_i^n \vec{x}_i}{n} \quad R = \sqrt{\frac{\sum_i^n (\vec{x}_i - \vec{x}_0)^2}{n}} \quad D = \sqrt{\frac{\sum_i^n \sum_j^n (\vec{x}_i - \vec{x}_j)^2}{n(n-1)}}$$

PARTITIONING METHODS

[Objective func: *Sum of Squared Errors*; Centroid / Medroid; Global VS Heuristic (如 greedy)]

1. K-Means [O(tKn), t: iter, n: #obj, K: #clus] [CON: 常在局部最优终止; 对 noise & outlier 敏感; 仅适用于 continuous n-dim. space, 不适用于 non-convex 形状的 cluster] {Distance: (L1: Manhattan, L2: Euclidean, Cosine)}; *分类型需要用 K-Modes!*
*** K-Means++*** *{改进 centroids 选择: 1st 随机选; 下一个选择离当前最远的; 直到选出 k 个结束}*

2. *K-Medians* [Distance: L1]; ***K-Modes*** [Freq.-based dissim. measure: $\Phi(x_j, z_j) = 1 - n_j^r/n_C$ if $x_j = z_j$, 1 if $x_j \neq z_j$ (z_j: categorical val. of j^{th} attr in z_C; n_C: #obj in cluster C; n_j^r: #obj whose attr = r)] {*fuzzy K-modes*; *K-Prototype* [数值型&分类型混合]};

3. *K-Medoids* {Each C: …assign to closest medroid; 随机选一个非代表 o_i; 计算交换 m 与 o_i 的 total cost S; 若 S<0 则选 o_i 为新代表并更新}; *Partitioning Around Medoids* [O(K(n-K)^2), Samples (O(Ks^2 + K(n-k)); good for small datasets√] {***CLARA***; ***CLARANS***}

4. *Kernel K-Means* [Detect non-convex clusters] {Map data points onto high-dim. Feature space; Perform K-Means;} **Spectral Clustering*

HIERARCHICAL METHODS

[Generate a clustering hierarchy (画作 dendrogram 系统树图); Not required to specify K; More deterministic; No iterative refinement;] [*CON*: 无法 undo what was done previously; Don't scale well]

1. *Agglomerative* [Start with singleton; Bottom-up] {***AGglomerative NESting*** [single-link; dissi. Matrix] [*CON*: 不适合数据量大] {merge 最接近的 nodes}};

{Single-link (最近邻) [对 noise & outlier 敏感]; Avg-link (group avg) [计算成本高]; Complete-link (直径) [outlier 敏感]; Centroid-link (centroid 相似); *Group Averaged Agglomerative Clustering*

[$N_a = |C_a|$, c_a= C_a centroid];

Ward's Criterion: 合并后 SSE 的增加} $c_{a\cup b} = \dfrac{N_a c_a + N_b c_b}{N_a + N_b}$

$$W(C_{a\cup b}, c_{a\cup b}) - W(C, c) = \frac{N_a N_b}{N_a + N_b} d(c_a, c_b)$$

2. *Divisive* [Start with huge macro; Top-down] {***DIvisive ANAlysis*** [recursively split higher level]};

3. *Other extensive algorithms*

****BIRCH (Balanced Iterative Reducing & Clustering using Hierarchies)*** [增量构造 CF-tree; Multi-level clustering (Low-level micro-clustering: 复杂度↓, scalability↑, preserve inherent clustering structure; High-level macro-clustering: Leave enough flexibility for high-level clustering)] {Scan DB 构造初始 in-memory CF-tree; 使用任意 clustering 算法 to cluster leaf nodes of the CF-tree}[*PRO*: Scales linearly] [*CON*: 对数据点插入顺序敏感; cluster 可能不自然; 易聚成球形]

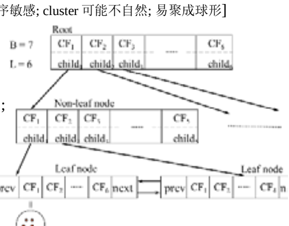
CF = (5, (16,30),(54,190))

(3,4) (2,6) (4,5) (4,7) (3,8)

Clustering Feature = (N, Linear Sum, Square Sum)

****CF-Tree Structure***

[Branching factor (max #child); max_D] {Each point in input: {Find closest leaf entry; Add point to it, update CF; If entry diameter > max_D: split leaf and maybe parent};}



CURE (Clustering Using REpresentatives) [用 well-scattered 的 REpre. point 表示; shrinking factor α: 点向中心按该比例 shrink, 越远的越狠 (对 outlier 鲁棒); cluster distance: REpre. point 的最小距离] *[点的选择→聚类任意形状]*

CHAMELEON (Hierarchical Clustering Using Dynamic Modeling) [基于 dynamic model 衡量相似性] {只有当两个 cluster 之间的 interconnectivity (RI) & closeness / proximity (RC) 高于其内部~&~时才合并}

[2-phase: graph-partitioning, agglo. hier. clustering]

Interconnectivity: (Absolute) EC(C_i, C_j) = $\sum_i \sum_j w_{i,j}$; (Relative) RI(C_i, C_j) = 2 * |EC(Ci, Cj)| / (|EC_{Ci}| + |EC_{Cj}|), EC_{Ci} = *size of its min-cut bisector*;

Closeness: (Relative) $RC(C_i, C_j) = \dfrac{\bar{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}}$

$\bar{S}_{EC_{Ci}}$ is avg. weights of edges that belong to the min-cut bisector of Ci; $\bar{S}_{EC_{\{Ci,Cj\}}}$ is the avg. weight of edges connect vertices in Ci to vertices in Cj.

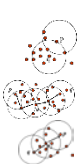
Algorithmic Hierarchical Clustering [*CON*: 不易选择好的距离量度; 不易处理丢失 attr; 优化目标不清晰];

Probabilistic Hierarchical Clustering [*PRO*: Generative model; 易于理解]{Quality(C_1, …C_m) = $\prod_{i=1}^{m} P(C_i)$, P(C_i): 最大似然; Dist(Ci, Cj) = $-log\frac{P(Ci \cup Cj)}{P(Ci)\,P(Cj)}$: if < 0, merge}

DENSITY-BASED METHODS

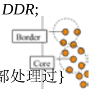
[任意形状; 对噪音鲁棒; 一遍扫描; 需要密度参数作为终止条件]

1. *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* [*Eps*(ε): max. radius; *Eps(ε)-neighborhood*: $N_{Eps}(q)$ ={p ∈ D | dist(p, q) ≤ Eps(ε)}; *MinPts*: min. #points in a point's N_{Eps}]



[点 p 和点 q: *Directly-Density-Reachable*: p ∈ N_{Eps}(q) && |N_{Eps}(q)| ≥ MinPts; *Density-Reachable*: a chain of points, 相邻的点 *DDR*; *Density-Connected*: 点 o 同时 *DR* 到 p 和 q]

{随机选点 p; 找到其 *DR* 点: 若 p 是 core, 成团; 若 p 在边界 || 没有点 DR 到 p, 则继续下个点; 直到全部处理过}



[If spatial index used: O(n log n); Else: O(n^2)]

[*CON*: Sensitive to parameter setting]

2. *OPTICS (Ordering Points To Identify Clustering Structure)*

[Process higher density points first;

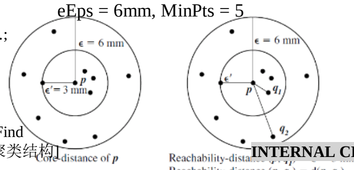
Core distance: smallest value *ε s.t.* p 的*ε-neiborhood* 有至少 MinPts obj.;

p 到 core q 的 *Reachability distance*: 使得 p 取 DR 到 q 的最小半径, = max (core-distance(*q*), distance (*q*, *p*))];

[If index-based: O(N log(N))]

[*PRO*: Good for auto. & interactive; Find intrinsic, even hierarchically nested 聚类结构]

eEps = 6mm, MinPts = 5



3. *DENCLUE*; **4. *CLIQUE***;

GRID-BASED CLUSTERING

[将 data space 分有限个 cell 来构成 grid 结构, 并从中找到 clusters]

[*PRO*: Efficiency, scalability: #cells << #data points]

[*CON*: *Uniformity*: 难以处理高度不规则的分布; *Locality*: Limited by predefined cell sizes, borders, density threshold; *Curse of dim*.]

1. ***STING (a STatistical INformation Grid approach)***

[Efficiency: O(K), K = #grid cells @ lowest level << N]

[*PRO*: Query-independent; 容易并行; Incremental update]

[*CON*: probabilistic nature → loss of accuracy]

{Spatial area is divided into rect. cells at diff. levels of resolution; Cells at high level contains smaller cells of next lower level; Param. of higher level cells 可以通过 lower level 的计算出来 (Stat: #, avg, std. dev, min, max; Dist. type: normal, uniform)}

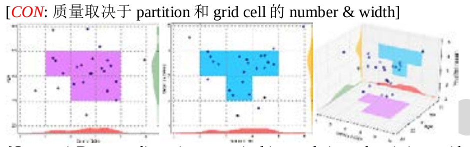
{从 root 开始使用 STING index 处理到 next lower level; 计算一个 cell 在特定置信度下与 query 相关的 likelihood; 只递归处理 likely relevant cells 的 children; 重复直到达到底层}

2. *CLIQUE (CLustering In QUEst)*

[*Density-based*: discretize data space through a grid, estimate density by counting #points in a cell; *Grid-based*: a cluster is a max. set of connected dense units in a subspace; *Subspace clustering*: a subspace cluster is a set of neighboring dense cells in an arbitrary subspace]

[*PRO*: Automatically finds subspace of highest dim.; 对 record 顺序不敏感; *Scale* linearly with size of input]

[*CON*: 质量取决于 partition 和 grid cell 的 number & width]



{Start at 1-D space, discretize numerical intervals in each axis into grid; Find dense regions in each subspace, generate their min. description; Use dense regions to find promising candidates in 2-D space based on Apriori principle; Repeat in level-wise manner in higher dim. subspace}

EXTERNAL CLUSTERING EVALUATION

[Supervised, employ criteria not inherent to dataset]

Given the *Ground Truth* **T**, *Quality Measure* **Q(C, T)** is good if: [*Cluster homogeneity*; *Cluster completeness*; *Rag bag (破烂) better than alien*: 异构 obj 在 pure cluster 里应比在"闲杂"里被 penalize 更多; *Small cluster preservation*]

1. Matching-based Measures

Purity = Precision [0, *1*] $purity_i = \dfrac{1}{n_i}\max_{j=1}^{k}\{n_{ij}\}$

$$purity = \sum_{i=1}^{r}\frac{n_i}{n}purity_i = \frac{1}{n}\sum_{i=1}^{r}\max_{j=1}^{k}\{n_{ij}\}$$

Maximum Matching (only 1 cluster can match 1 partition)

C\T	T1	T2	T3	Sum
C1	0	20	30	50
C2	0	20	5	25
C3	25	0	0	25
m_j	25	40	35	100

C\T	T1	T2	T3	Sum
C1	0	30	20	50
C2	0	20	5	25
C3	25	0	0	25
m_j	25	50	25	100

$$w(M) = \sum_{e\in M} w(e) \qquad match = \arg\max_{M}\{\frac{w(M)}{n}\}$$

E.g. (green & orange) $purity_1$ = 30/50; $purity_2$ = 20/25, $purity_3$ = 25/25, *purity* = (30 + 20 + 25)/100 = 0.75; (green) *match* = *purity* = 0.75, (orange) *match* = 0.65 > 0.6; (green) *recall*$_1$ = 30/35; *recall*$_2$ = 20/40; *recall*$_3$ = 25/25;

Recall (cluster 里最主要的分类的点 占 该分类全部点的比例)

F-measure (harmonic means of *precision* and *recall*)

$$recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}} \qquad F_i = \frac{2n_{ij_i}}{n_i + m_{j_i}} \qquad F = \frac{1}{r}\sum_{i=1}^{r}F_i$$

2. Entropy-based Measures $p_{C_i} = \frac{n_i}{n}$ (i.e., the probability of cluster C_i)

Ent. of clustering $H(\mathcal{C}) = -\sum_{i=1}^{r} p_{C_i}\log p_{C_i}$

Ent. of partitioning $H(\mathcal{T}) = -\sum_{j=1}^{k} p_{T_j}\log p_{T_j}$

Cond. ent. of T w.r.t. C $H(\mathcal{T}|\mathcal{C}) = -\sum_{i=1}^{r}\left(\frac{n_i}{n}\right)H(\mathcal{T}|C_i) = -\sum_{i=1}^{r}\sum_{j=1}^{k}p_{ij}\log\left(\frac{p_{ij}}{p_{C_i}}\right)$

Mutual Information [0, +∞] $I(C,T) = \sum_{i=1}^{r}\sum_{j=1}^{k}p_{ij}\log(\frac{p_{ij}}{p_{C_i}\cdot p_{T_j}})$

NMI (Normalized Mutual Info.) [0, *1*] $NMI(\mathcal{C},\mathcal{T}) = \sqrt{\frac{I(\mathcal{C},\mathcal{T})}{H(\mathcal{C})}\cdot\frac{I(\mathcal{C},\mathcal{T})}{H(\mathcal{T})}} = \frac{I(\mathcal{C},\mathcal{T})}{\sqrt{H(\mathcal{C})\cdot H(\mathcal{T})}}$

3. Pairwise Measures [] {}; $TP = \sum_{i=1}^{r}\sum_{j=1}^{k}\binom{n_{ij}}{2} = \frac{1}{2}((\sum_{i=1}^{r}\sum_{j=1}^{k}n_{ij}^2) - n)$

Four possibilities (TP, FN, FP, TN)

$$FN = \sum_{j=1}^{k}\binom{m_j}{2} - TP \qquad FP = \sum_{i=1}^{r}\binom{n_i}{2} - TP \qquad N = \binom{n}{2}$$

$$TN = N - (TP + FN + FP) = \frac{1}{2}(n^2 - \sum_{i=1}^{r}n_i^2 - \sum_{j=1}^{k}m_j^2 + \sum_{i=1}^{r}\sum_{j=1}^{k}n_{ij}^2)$$

Jaccard Coefficient (Jaccard = TP / (TP + FN + FP))

Rand Statistic [0, *1*] (Rand = (TP + TN) / N)

Fowlkes-Mallow Measure (geometric mean of precision and recall)

$$FM = \sqrt{prec \times recall} = \frac{TP}{\sqrt{(TP+FN)(TP+FP)}}$$

4. Correlation Measures

INTERNAL CLUSTERING EVALUATION

[Unsupervised, criteria derived from data itself] (compact, separated)

1. *BetaCV Measure* [Trade-off of intra-cluster compactness VS inter-cluster separation] $BetaCV = \dfrac{W_{in}/N_{in}}{W_{out}/N_{out}}$

{W(S, R): sum of weights on all edges with one vertex in S and the other in R; $W_{in} = \frac{1}{2}\sum_{i=1}^{k}W(C_i, C_i)$

W_{in}: sum of all the intra-cluster weights over all clusters;

W_{out}: sum of all the inter-cluster weights;

N_{in}: #distinct intra-cluster edges;

N_{out}: #distinct inter-cluster edges;} $N_{in} = \sum_{i=1}^{k}\binom{n_i}{2}$ $N_{out} = \sum_{i=1}^{k-1}\sum_{j=i+1}^{k}n_i n_j$

越小越好！ $W_{out} = \frac{1}{2}\sum_{i=1}^{k}W(C_i, \overline{C_i}) = \sum_{i=1}^{k-1}\sum_{j>i}W(C_i, C_j)$

2. *Normalized Cut* 越高越好

$$NC = \sum_{i=1}^{k}\frac{W(C_i,\overline{C_i})}{vol(C_i)} = \sum_{i=1}^{k}\frac{W(C_i,\overline{C_i})}{W(C_i,V)} = \sum_{i=1}^{k}\frac{W(C_i,\overline{C_i})}{W(C_i,C_i)+W(C_i,\overline{C_i})} = \sum_{i=1}^{k}\frac{1}{\frac{W(C_i,C_i)}{W(C_i,\overline{C_i})}+1}$$

[*vol*(C_i) = *W*(C_i, *V*): the volume of cluster C_i]

3. *Modularity* (for graph clustering) 越小越好！

$$Q = \sum_{i=1}^{k}\left(\frac{W(C_i,C_i)}{W(V,V)} - \left(\frac{W(C_i,V)}{W(V,V)}\right)^2\right)$$

$$W(V,V) = \sum_{i=1}^{k}W(C_i,V) = \sum_{i=1}^{k}W(C_i,C_i) + \sum_{i=1}^{k}W(C_i,\overline{C_i}) = 2(W_{in} + W_{out})$$

RELATIVE CLUSTERING EVALUATION

[Directly compare diff. clusterings, esp. those obtained via different parameter settings for same algorithm]

1. *Silhouette Coefficient* *as an* ***internal measure***

[Check cluster cohesion & separation] [0, *1*]

$$SC = \frac{1}{n}\sum_{i=1}^{n}s_i \quad s_i = \frac{\mu_{out}^{\min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{\max\{\mu_{out}^{\min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\}}$$

{For each point $\boldsymbol{x}_i$, its SC s_i: $\mu_{in}(x_i)$: avg. dist. from $\boldsymbol{x}_i$ to points in its own cluster; $\mu_{out}^{min}(x_i)$: avg. dist. from $\boldsymbol{x}_i$ to points in its closest cluster}

2. *Silhouette Coefficient* *as a* ***relative measure***

[Estimate the # of clusters in the data] $SC_i = \dfrac{1}{n_i}\sum_{x_j\in C_i}s_j$

{Pick the *k* value that yields the best clustering, i.e., yielding high values for *SC* and SC_i ($1 \le i \le k$)}

CLUSTER STABILITY

[Cs obtained from several datasets sampled from the same underlying distribution as D should be similar or "stable"]

1. *Bootstrapping Approach* [find the best value of k (judged on stability)] {从 D 有放回地取样 t 个 size 为 n 的样本 Di; 对每个样本 Di, 使用从 2 到 k_{max} 的 k 值运行聚类算法; 比较每一对聚类 $C_k(D_i)$和 $C_k(D_j)$ 的(某种)距离; 展示出聚类之间最小 deviation 的 k*是最佳选择;}

2. *Empirical Method*

{# of clusters: k ≈ $\sqrt{n/2}$}

3. *Elbow Method*

{使用"#Cluster – Avg. within-cluster squared sum"曲线的拐点值}

4. *Cross Validation Method*

{将一个数据集分成 m 部分; 使用 m-1 个部分 to obtain a clustering model; 用剩下的部分来测试聚类效果; 对每个 k>0, 重复 m 次;}

CLUSTERING TENDENCY / CLUSTERABILITY

[*Assess* the suitability of clustering *(if data has any inherent grouping structure)*; *Hard task* because so many different definitions of clusters]

1. *Spatial Histogram*

[比较从数据中与从随机样本中 生成的 ***d***-dim. 直方图: Dataset D is clusterable if the distributions of two histograms are rather different] {分别对 Dataset 和随机样本: 将每个维度分成 equi-width bins, 算出每个 cell 中点的数量, 得到 empirical joint probability mass function (EPMF); 使用 *Kullback-Leibler (KL) Divergence* 计算差异}

2. *Distance Distribution*

3. *Hopkins Statistic*