Questions PV056 semestral test

L1 Advanced tree learning

- MDL maximum description depth
  - Determine if the additional complexity of the hypothesis is less complex than just explicitly remembering any exceptions resulting from pruning.
- Minimal tree
  - minimal decision tree (over nodes, leaves, or depth) is an NP-hard optimization problem. => Heuristic algorithm
  - IG biases to shallow trees => Want to pick a feature that creates subsets of examples that are relatively "pure" in a single class so they are "closer" to being leaf nodes.
- Regression trees
  - Just replace IG with weighted variance aka **Impurity measure** and minimize it
  - $$\mathrm{Var}(\{Y_1,\ldots,Y_l\}) = \sum_{j=1}^{l}\frac{|Y_j|}{|Y|}\mathrm{Var}(Y_j) = \ldots = \frac{1}{|Y|}\sum_{y\in Y} y^2 - \sum_{j=1}^{l}\frac{|Y_j|}{|Y|}\overline{y}_j^2$$
  - Simple type: Predict the mean value of the leaf
  - Advanced: can have a regression model
  - Report RMSE = sqrt(L2)
- Information gain
  - $$Gain(S,F) = Entropy(S) - \sum_{v\in Values(F)}\frac{|S_v|}{|S|}Entropy(S_v)$$
  - Information gain gives a bias for trees with minimal depth.
- Decisions based on
  - Minority class: min(p, 1-p)
  - Gini index: 2p(1-p)
  - IG from Entropy: Sum -p log p
- MDL and learning decision tree
  - Goes from **Kolmogorov complexity** of an object, such as a piece of text, is the length of a shortest computer program (in a predetermined programming language) that produces the object as output.
  - Shorter representation is the right one
    - => code the decision tree and exceptions to reach the minimum message length

L2 Bias variance tradeoff

- Not a law, just observation with some theoretical background in statistics and classical ML
- Was put into question in recent years by models with **many folds** larger than the data
- All from those equations

  $y = f(x) + e, \quad E[e] = 0, \mathrm{Var}[e] = s^2$

  f'() … estimate of f() learned by a classifier

- $E[(y-f'(x))^2] = \mathrm{Bias}[f'(x)] + \mathrm{Var}[f'(x)] + s^2$
- The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The variance is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

L3 Ensambles

- Stacking
  - Same model, disjunct data => train a model on top
  - Can be done mostly in parallel
- Bagging
  - Same model*, sample N from N with repetition => each model learns different aspects
  - Can be done in parallel
  - Decreases variance
- Boosting
  - Same model*, iteratively weighting examples based on their error
  - Decreases bias

L3 Asociation rules

- A ⇒ B[support, confidence]
  - support % of transactions in D that contain both A and B
  - confidence % of transactions in D from those containing A that contain also B
- Apriori
  - Generate frequent
    1. From L_{k-1} generate candidates C_k délky k
       1. If in L_{k-1} appear two $(k-1)$-tuplesthat differ only in one item, build new k-tuple and add it to C_k
       2. From C_k remove those candidates that contain any $(k-1)$-tuple that are not in L_{k-1}

       Example
       $L_3 = \{\{ABC\}\{ABD\}\{ACD\}\{ACE\}\{BCD\}\}$
       $C_4 = \{\{ABCD\}\{ACDE\}\}$
       $L_4 = \{\{ABCD\}\}$ *because* $\{ADE\} \notin L_3$
    2. Select all C_k such that $support(C_k) > minsup$
  - Generate association rules

    l ... *frequent pattern*, $a \subset l$
    $(l-a) \Rightarrow a[supp(l), \frac{supp(l)}{supp(l-a)}]$, $\frac{supp(l)}{supp(l-a)} \geq minconf$

    Example
    $l = \{ABCD\}$
    $a = \{CD\} : AB \Rightarrow CD[support(ABCD), \frac{support(ABCD)}{support(AB)}]$
    $a' = \{D\} : ABC \Rightarrow D[support(ABCD), \frac{support(ABCD)}{support(ABC)}]$
    1. Generate a set of rules H_1 with a single item in a consequent
    2. From rules H_{k-1} build H_k
  - Post-processing

    One quantity that is often used in post-processing is **lift**, defined as

    $$Lift(B \Rightarrow H) = \frac{n \cdot Support(B \cup H)}{Support(B) \cdot Support(H)}$$

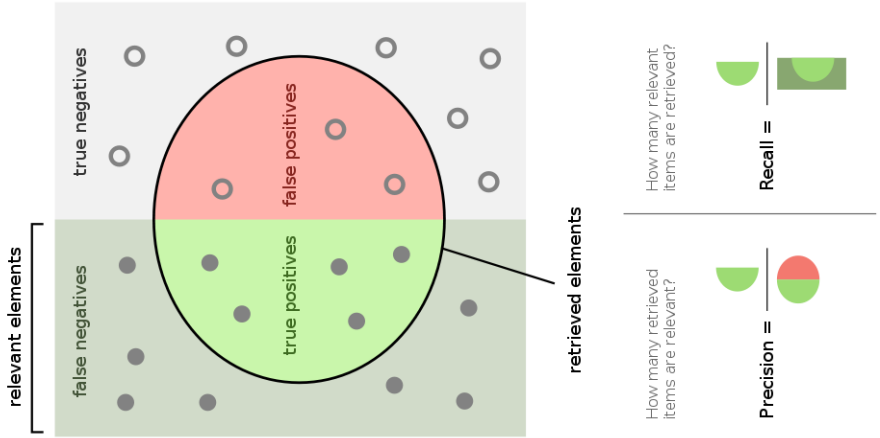
    - where n is the number of transactions.
    - Has to be > 1
- Class-association rules
  - Rules of type: many => one
  - Can be used for classification — directly or as weak labeling

L3 Experiment evaluation

- Correlation coefficient

$$\frac{S_{PA}}{\sqrt{S_P S_A}}, \text{ where } S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1},$$
$$S_p = \frac{\sum_i (p_i - \bar{p})^2}{n-1}, \text{ and } S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1}$$

- ROC
  - False positive × True positive (X × Y) while changing a threshold (theta)
- Recall-precision curve



- Cross-validation significant difference
  - Null hypothesis: all model's performance is from a standard distribution
  - T-test with |folds| - 1 (we have std as an unknown parameter)
- Comparing algos/models across weirder settings
  - Wilcoxon's signed-rank (a more advanced idea of: subtract the results, look if positive values are ~1/2)
- Friedman test for multiple datasets on multi algos

L3 Inductive logic programming

- Inductive logic programming. Main task, dis/advantages of ILP. When ILP is useful?
- Logical consequence and generalization/specialization. A general refinement (specialization) operator
- Refinement operator for a given domain (arithmetics, lists, hierarchy).
- Generic algorithm for ILP. Covering paradigm
- Aleph
- Inverse resolution. Idea, example
- Subsumption
- Sunsumption in a propositional logic
- Ideal specialization operator for atoms (in predicate logic)
- Theta -subsumption

L4 Graph mining

- Local clustering coefficient
  - 0 – nolinks between neighbors
  - 1 – all links between neighbors
- Inter-graph similarity (between graphs)
  - Isomorphism – in NP (but not known whether in P or in NP-complete)
  - Maximum Common Subgraph – NP-hard
  - Graph-edit distance – for given costs of operations

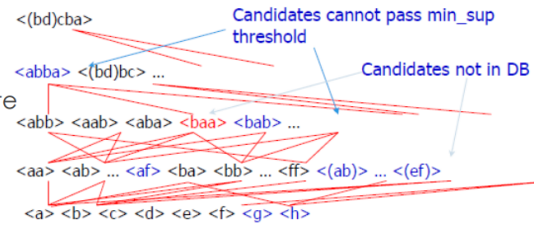
    - Frequent subgraphs
- Frequent subgraph mining (FSM)
    - Frequent subgraph: A subgraph that occurs in at least n graphs (or n times in one
    - graph) on input for a given minimum support n.
- gSpan (algo for Frequent subgraph mining (FSM))
    - Complete (finds all frequent subgraphs) FSM algorithm on labeled undirected graphs
    - Generates candidates by adding edges (one at a time) to patterns already discovered
    - Encodes patterns (graphs) according to DFS traversal order (DFS code)
- Between-graph classification (class per graph)
    - gBoost: Decision tree where features is presence of a particular subgraph
    - Any CLF on FSM algorithm output
- Within-graph classification (class per node)
    - Kernel methods (eg. SVM)
- Clustering
    - Between graph
        - K-means on kernels
        - Any clustering on FSM
    - Within a graph
        - By removing the largest values from minimal spanning tree
        - Shared Nearest Neighbor (SNN) Clustering
- Community detection (≈ clustering nodes)
    - Label Propagation Algorithm
        - 1. Initialize the labels at all nodes in the network. For a given node x, set $Cx\ 0 = x$ ($Cx\ t$ denotes the community / label of the node x at time t)
        - 2. Set $t = 1$
        - 3. Arrange the nodes in the network in a random order and set it to X
        - 4. For each $x \in X$ chosen in that specific order, set $Cx(t)$ to the most frequent label among neighbors of x. Ties are broken uniformly
        - 5. If every node has a label that the maximum number of their neighbors have, then stop the algorithm. Else, set $t = t + 1$ and go to (3).

L4 Sequence mining

- Sequence mining. Idea. frequent and closed frequent patterns. GSP algorithm.
    - Frequent = support over threshold
    - Closed = no propper superset with the same support

GSP algorithm (Apriori-based)

- Initial candidates: all singleton sequences
- Scan DB once, count support for each candidate
- Generate length-2 candidate sequences
- Repeat (for each level (i.e., length-k))
    - Scan DB to find length-k frequent sequences
    - Generate length-(k+1) candidate sequences from length-k frequent sequences using Apriori
    - set k=k+1
- Until no frequent sequences or no candidate can be found



- Sequence features and distance functions. Representation of sequences
    - K-grams, k-gapped pairs, closed patterns, #
    - Difference of k-kapped pairs # vectors
    - Humming distance, edit distance
- Constraints in sequence mining.

    - anti-monotonic (>max), monotonice (<sum)
    - Length, required element, super-pattern, duration…
- Mining partial orders (only for oral exam)
    - TODO :D

L5 Temporal data mining

- Trend: long-term change in the mean of the data;
- Seasonality: regular and predictable changes;
- Residuals: de-seasonalised and de-trended series;
- Stationarity: When time series properties remain constant over time;
- Autocorrelation: Correlation with past observations;
- Heteroskedasticity: Changes in the variance;
- Regularity: Whether the series is captured at regular intervals;
- Frequency: Frequency at which the series is observed;
- Reflexivity: When the forecast affects the outcome;
- Outliers: Rare but possibly interesting observations;
- Regimes and Change Detection: When the data distribution changes;
- Dimensionality: Number of variables in the time series.

L6 Anomaly analysis. Class-based outliers. Outlier explanation

- 1. Class-based outliers (CBO). Idea. Example of a task solved by CBO.
    - look anomalous when the class labels are taken into account but th eydo not have to be anomalous when the class labels are ignored

      4. LOCAL REACHABILITY DENSITY (LRD)

    - $$LRD_k(A) = \frac{1}{\sum_{Xj \in N_k(A)} \frac{\mathrm{RD}(A,Xj)}{\|N_k(A)\|}}$$

      5. LOCAL OUTLIER FACTOR (LOF)

    - $$LOF_k(A) = \frac{\sum_{Xj \in N_k(A)} LRD_k(Xj)}{||N_k(A)||} \times \frac{1}{LRD_k(A)}$$
    - Funny reviews on IMDB
    - Missclassified clients/potential clients if normally clustered
- 2. CODB

  COF(T) = SimilarityToTheK-NearestNeighbors +
  α * 1/DistanceFromOtherElementsOfTheClass +
  β * DistanceFromTheNearestNeighbors

  COF(T) = K*PCL(T,K) + α * 1/Dev(T) + β *Kdist(T)

  PCL(T,K) ... the probability of the class label of T w.r.t. the K nearest neighbors

  Dev(T) ... ... the sum of distance from all other elements from the same class

  Kdist(T) ... the distance between T and its K nearest neighbors
- 3. Random Forest for outlier detection. Proximity matrix. RF-OEX

- After each tree is built, all of the data are run down the tree, and **proximities** are computed for each pair of cases:
- If two cases occupy the same terminal node, their proximity is increased by one.
- At the end of the run, the proximities are normalized by dividing by the number of trees.
- Define the average proximity from case n in class j to the rest of the training data class j as:

$$\bar{P}(n) = \sum_{cl(k)=j} \text{prox}^2(n,k)$$

- The raw outlier measure for case n is defined as

$$\text{nsample}/\bar{P}(n)$$

- 4. Outlier explanation. Choose two methods.
  - ptakopysk/dziobak
  - using random forest branch reduction / frequent branches

Explaining outliers by subspace separability
(Micenkova and Ng 2013)

- Cannot derive explanatory subspace just by analyzing vicinity of the point in full space ⇒ need to consider different subspace projections
- no monotonicity property for outliers wrt. subspaces
- need for heurstics because of exponential complexity,

look for a subspace *A* where the outlier factor is high and the dimension of *A* is low

- separability - instance outlierness is related to its separability from the rest of the data

L7 Dataset and Instance hardness

- Dataset hardness.
  - Dataset hardness measures: idea, 3 main factors of dataset hardness.
  - 6 groups of measures. Describe one form each group in details.
    - Measure of overlap of individual feature values
      - (mean0 + mean1) /(std0 + std1)
    - of separability of classes
      - linear separability (SVM, or better linear)
    - of geometry, topology, manifolds
      - non linearity: knn tested on interpolated test set

Instance hardness.

- Instance hardness measures: idea, theoretical model. Classifier output difference.

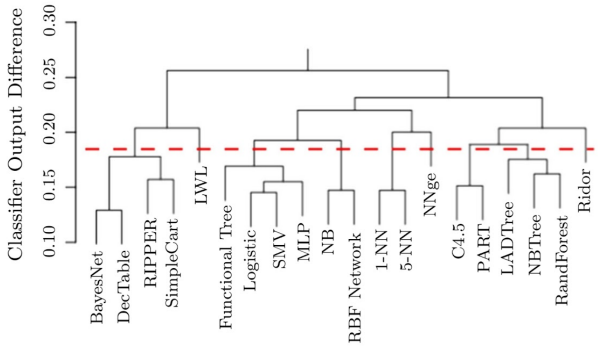
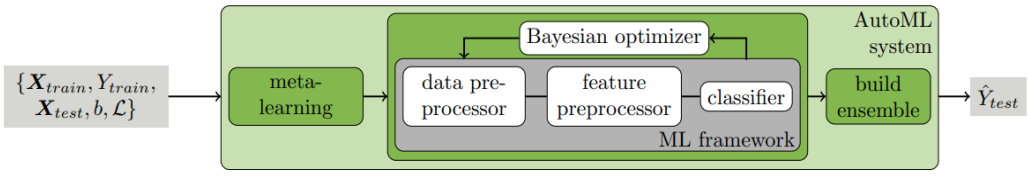


Fig. 3 Dendrogram of the considered learning algorithms clustered using unsupervised metalearning based on their classifier output difference

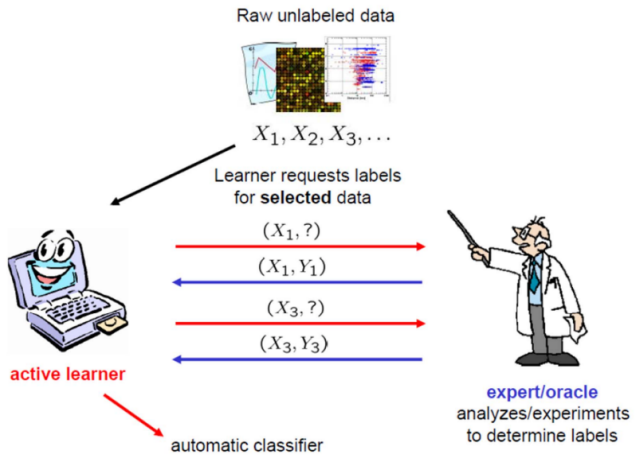
        - COD measures the distance between two learning algorithms as the probability that the learning algorithms make different predictions.
        - Idea: outliers do not fit into theoretical bayes view of model prediction => graphical model with outlier chance
            - x is the set of input features, yˆ is the observed, possibly noisy, class label given in the training set, and y is the actual unkown class label
        - $$p(y_i|\hat{y}_i, x_i) \approx p(\hat{y}_i|x_i) = \sum_{h \in \mathcal{H}} p(\hat{y}_i|x_i, h)p(h).$$
    - what measures you know, one in details. Indicator function and classifier scores.

L9 Auto ML

- Automated machine learning, main idea.
    - Combined Algorithm Selection and Hyperparameter optimization (CASH) problem used by AUTO-WEKA's AutoML approach
    - Premise: No single algo is the best for all datasets
    - No human intervention, limited computational budget
- Tools. AutoWeka AutoScikitlearn - main principles. One tool (not necesssarily some from those two) in detail
    - Auto preprocessing, feature generation/feature preprocessing, algo selection
    - Hyperparams using Bayes optimization
    - AutoScikitLearn
        - Warm start bayes optimization
        - AutoScikitlearn => extend on Weka, automatically build ensambles from the Bayes tried algorithms
        - 

- Bayesian optimization, main idea
    - A probabilistic model that captures the relationship between hyperparams and the loss. This model is used to select more hyperparams to be evaluated based on exploration-exploitation tradeoff

L10 active learning

- Active learning. Definition. Uncertainty sampling. Query by committee. Applications
    - Done because acquiring labeled data is expensive
    - 

- Uncertainty sampling

  least confident [Culotta & McCallum, AAAI'05]

  $$\phi_{LC}(x) = 1 - P_\theta(y^*|x)$$

  smallest-margin [Scheffer et al., CAIDA'01]

  $$\phi_M(x) = P_\theta(y_1^*|x) - P_\theta(y_2^*|x)$$

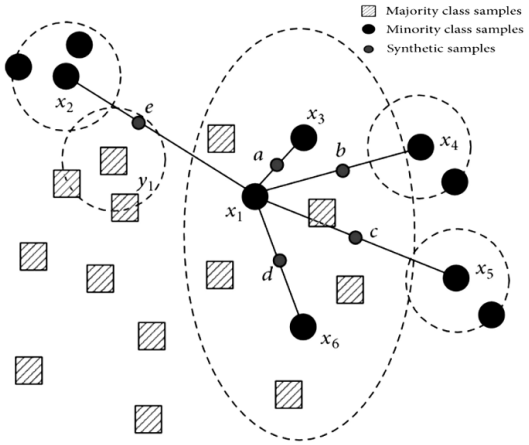
  entropy [Dagan & Engelson, ICML'95]

  $$\phi_{ENT}(x) = -\sum_y P_\theta(y|x) \log_2 P_\theta(y|x)$$

  -
- Committee voting
  - Bagging/boosting many models
  - Maximal disagreement (XOR or uncertainty, eg. max entropy)
- Semi-supervised learning. Definition. How useful it is. Applications
  - Part of the data is labeled, most is not
  - Language model pretraining! Error detection (wind turbines)
- Supervised, semi-supervised and active learning. Comparison.
  - Theoretically Supervised <<< semi < active

L11 Learning from imbalanced data. Multirelational learning (ILP)

- Performance measures for imbalanced data.
  - F1 :D
  - $$F_\beta = \frac{(\beta^2 + 1) \cdot Prec \cdot Rec}{\beta^2 \cdot Prec + Rec}$$
  - ROC curve
  - $$Gm = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} = \sqrt{sensitivity \times specificity}$$
- Pre-processing imbalanced data. Undersampling, oversampling.
  - Undersampling: remove more provelant class examples
  - Oversampling: replicate the underrepresented classes
    - Oversampling techniques refer to create artificial minority class points. Some oversampling techniques are Random Over Sampling, ADASYN, SMOTE, etc.



    -

- Modification of classifiers to be able to handle imbalanced data. Cost-sensitive learning.
  - Add weight to the underrepresented classes! For example in trees, or in NN
  - Cost-sensitive learning looks on utility = sum B - sum C
  -

		Pred.		
		c_1	c_2	c_3
Obs.	c_1	$B_{1,1}$	$C_{1,2}$	$C_{1,3}$
	c_2	$C_{2,1}$	$B_{2,2}$	$C_{2,3}$
	c_3	$C_{3,1}$	$C_{3,2}$	$B_{3,3}$