Pruning Meta-Classifiers in a Distributed Data Mining System

Andreas L. Prodromidis

Salvatore J. Stolfo

Computer Science Department

Columbia University

Topics of Discussion

- The need for pruning meta-classifiers
- Classifier evaluation and selection metrics
- Pruning algorithms
- Experiments and results
- Conclusion and future plans

Scaling problem of machine learning

- **Objective**
  - Acquire information efficiently from large and distributed databases to improve predictive performance.
- **Problems**
  - Information is distributed.
  - Limited resources.
- **Approach**
  - Use meta-learning to combine classifiers learned separately over different databases (*JAM* system)

Overview of meta-learning

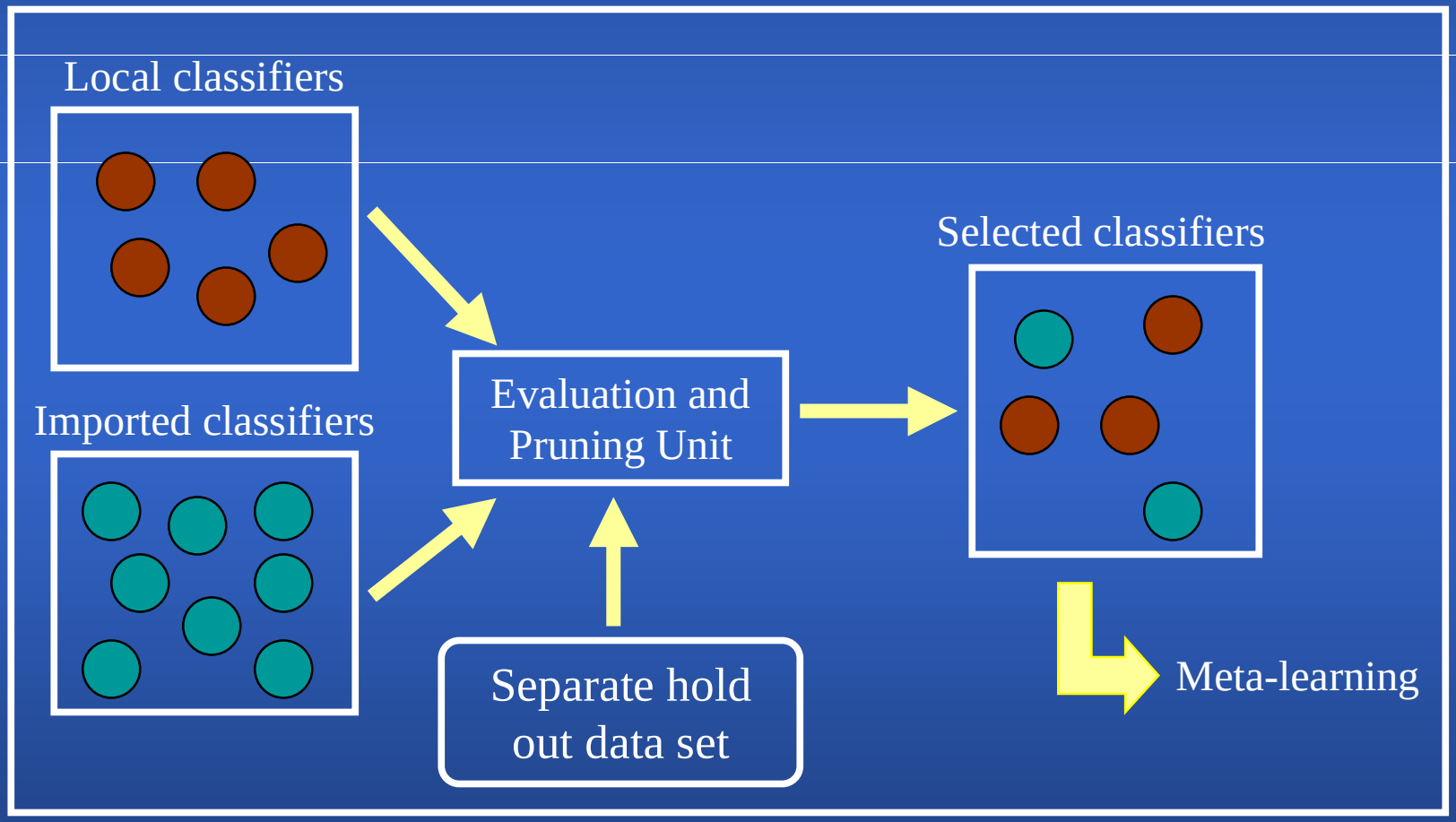
Deploys machine learning techniques to discover the predictive performance and properties of an ensemble of classifiers.

- Input: Classified predictions of the individual classifiers on an unseen validation set
- Output: A meta-classifier representing the collective knowledge

Pruning meta-classifiers

- Meta-classifiers may grow very large
  - as the number of data sites increases
  - as more learning algorithms become available
- System resources are bounded
  - memory constraints, throughput requirements
- Pre-training pruning
  - discard base classifiers before meta-learning
- Post-training pruning
  - discard base classifiers after meta-learning

Pre-training pruning



Evaluation and Selection Metrics

- Accuracy
- Correlation error
- Coverage
- Diversity
- Class Specialty / Cost
- Combinations of the above

Diversity

$$D = \frac{\sum_{i=1}^{|H|-1} \sum_{j=i+1}^{|H|} \sum_{k=1}^{n} Dif\left[C_i(y_k), C_j(y_k)\right]}{\frac{(|H|-1)\cdot|H|}{2} \cdot n}$$

- $Dif(a,b) = \begin{cases} 1 \text{ if } (a \neq b) \\ 0 \text{ if } (a = b) \end{cases}$
- H = set of classifiers
- n = number of examples

Class Specialty

- $TP(C_i, c_k)$ = rate at which C_i predicts c_k correctly
- $FP(C_i, c_k)$ = rate at which C_i predicts c_k by mistake
- One-sided specialty
  - $\max(TP(C_i, c_k))$, $\min(FP(C_i, c_k))$
- Two-sided specialty
  - $PCS(C_j, c_k) = \frac{TP(C_j, c_k) - FP(C_j, C_k)}{1 - TP(C_j, c_k)}$ $\quad NCS(C_j, c_k) = \frac{TP(C_j, c_k) - FP(C_j, C_k)}{FP(C_j, c_k)}$
- Combined specialty
  - $CCS(C_j, c_k) = f_{TP}(c_k) \cdot TP(C_j, c_k) + f_{FP}(c_k) \cdot FP(C_j, c_k)$
  - $CCS(C_j, c_k, (\mathbf{x}_i, y_i)) =$

    $f_{TP}(c_k, (\mathbf{x}_i, y_i)) \cdot TP(C_j, c_k, (\mathbf{x}_i, y_i)) + f_{FP}(c_k, (\mathbf{x}_i, y_i)) \cdot FP(C_j, c_k, (\mathbf{x}_i, y_i))$

Related work (metrics)

- Correlation error (Ali & Pazzani 1996)
- Coverage (Brodley & Lane 1996)
- Classification overlap (Brodley & Lane 1996)
- Diversity (Chan 1996, Kwok & Karter 1990, Ali & Pazzani 1996)

Diversity-based algorithm

- Let V be the validation set
- Compute diversity matrix D based on V
- Select the most accurate classifier on V
- Until* enough classifiers are chosen
  - Based on D, choose the classifier that is most diverse to the set of the selected classifiers

* Determined by system resources

Coverage/Specialty-based algorithm

- For all target classes c_k
- Let V be the validation set
- Until no more examples in V can be covered
  - select the classifier with highest specialty rate on V for class c_k
  - Remove covered examples from V
- Implemented instances:
  - specialty = PCS, CCS, cost model

Related work (algorithms)

- Linear combination of models
  - Breiman 1996, LeBlanc & Tibshirani 1993, etc
- SCANN, PCR*
  - Merz 1998, Merz & Pazzani 1998
- ROC convex hull
  - Provost & Fawcett 1997, 1998
- Pruning ADABOOST
  - Margineantu & Dietterich 1997

Fraud detection in Credit Cards

- Chase Credit Card data set
  - 500,000 records spanning one year
  - Evenly distributed
  - 20% fraud, 80% non fraud
- First Union Credit Card data set
  - 500,000 records spanning one year
  - Unevenly distributed
  - 15% fraud, 85% non fraud

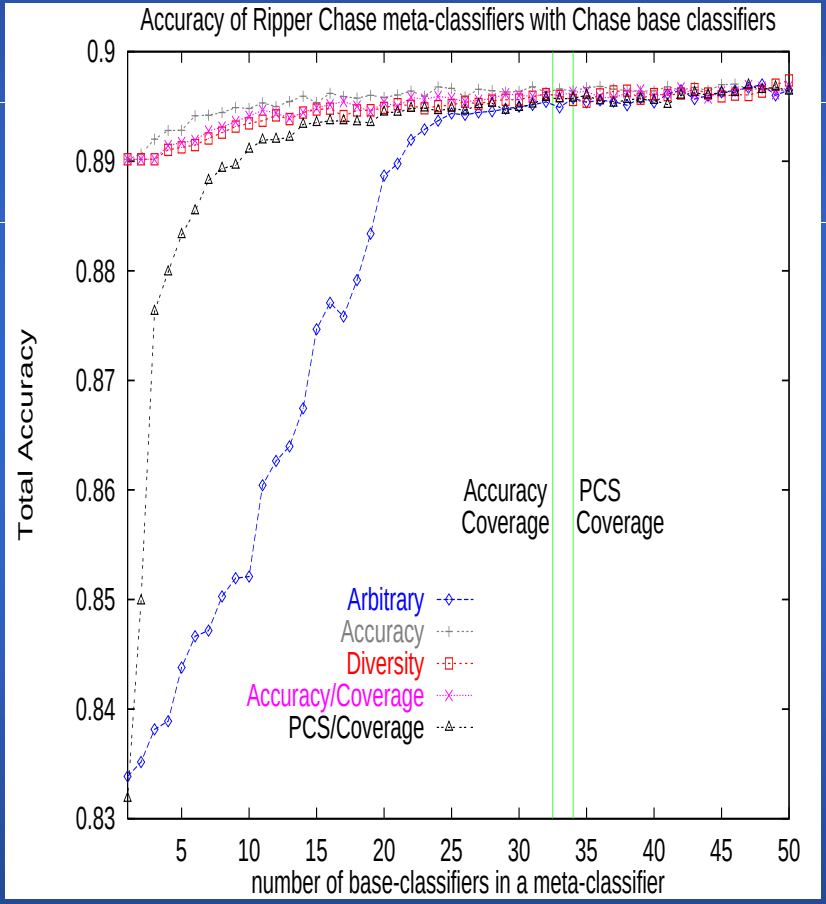
Experimental setting

- Algorithms
  - Naïve Bayes, C4.5, CART, ID3, Ripper
- Evaluation criteria
  - Accuracy, TP - FP, Cost model
- Cost model
  - $X = overhead to challenge a transaction
  - Ignore transactions with transaction amounts < $X
  - On a correct fraud prediction we save
    transaction amount - $X
  - On a false fraud prediction we lose $X

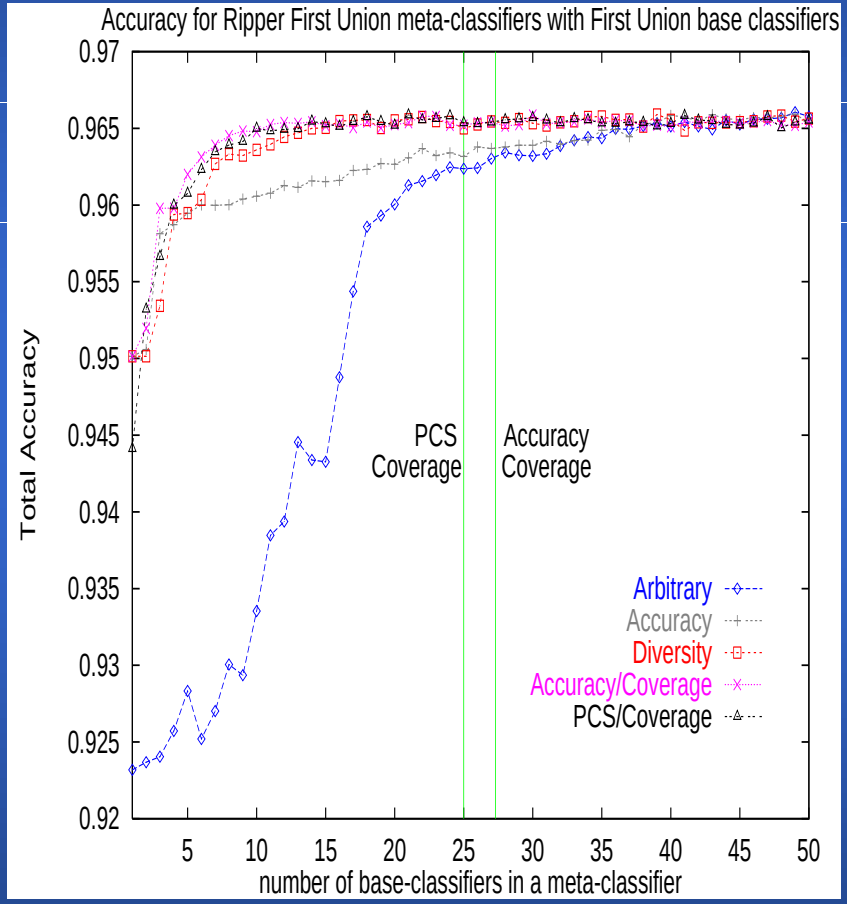
Experimental setting

- Divide data sets in 12 subsets
- Distribute into 6 data sites (2 subsets/site)
- Apply all algorithm to each subset
- Exchange classifiers
  - 10 local, 50 remote per site
- Meta-learn the remote classifiers
  - 2^{50} different meta-classifiers (approximately)
- Average the results
- Setting provides a parallel 6-fold cross validation

Accuracy results

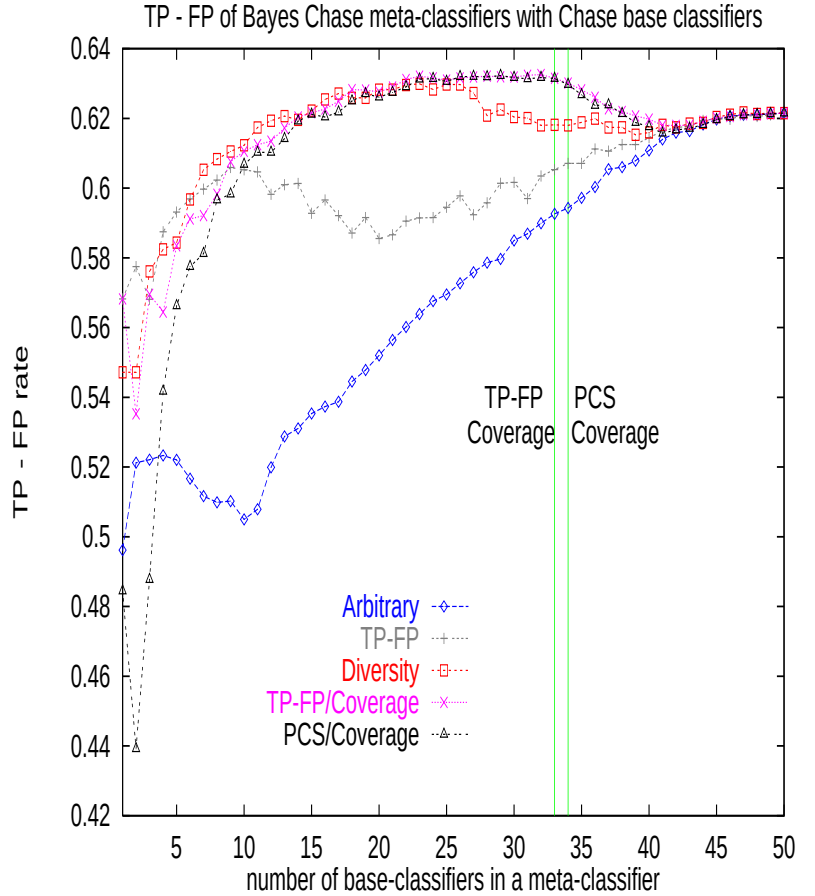


Chase

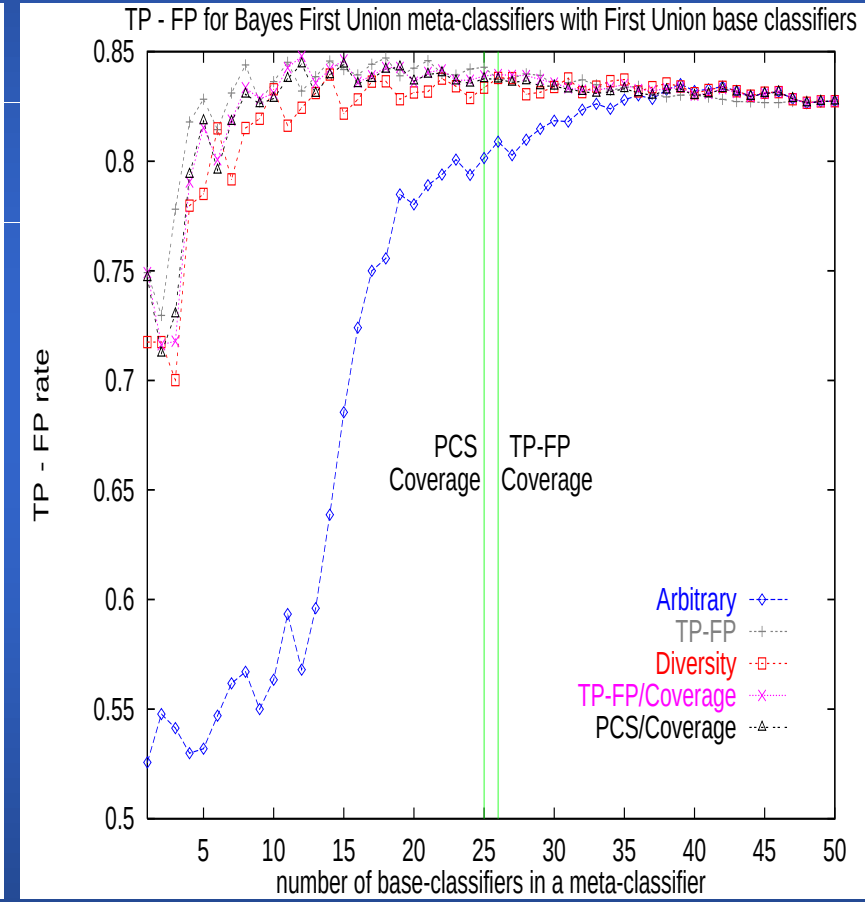


First Union

TP - FP spread

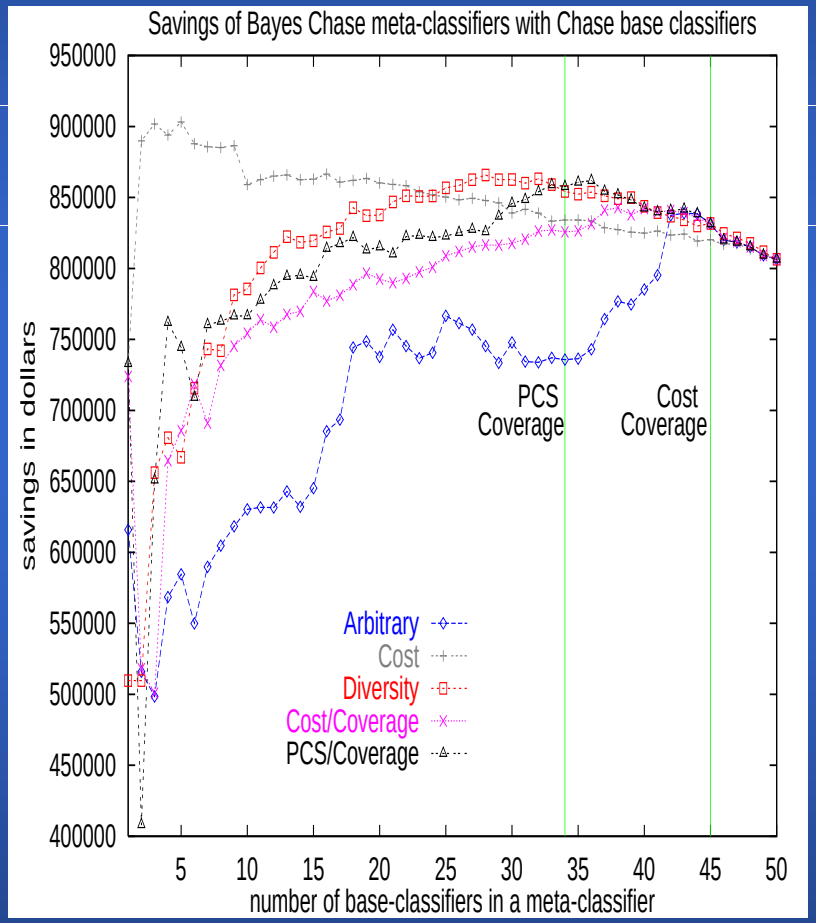


Chase

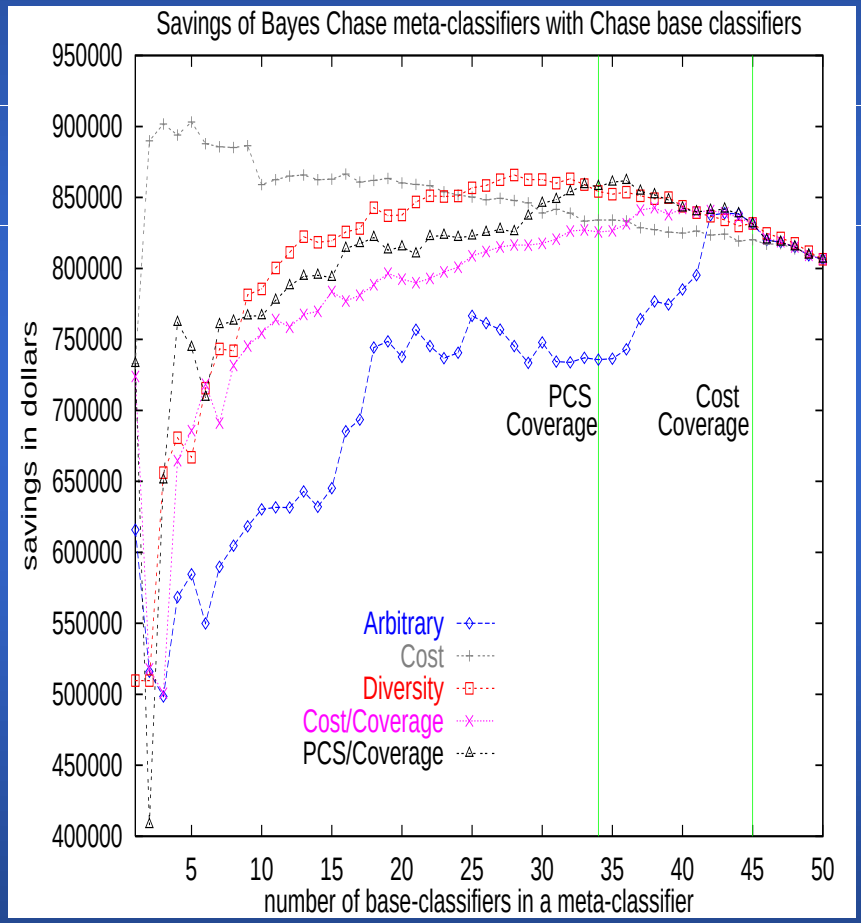


First Union

Cost Model

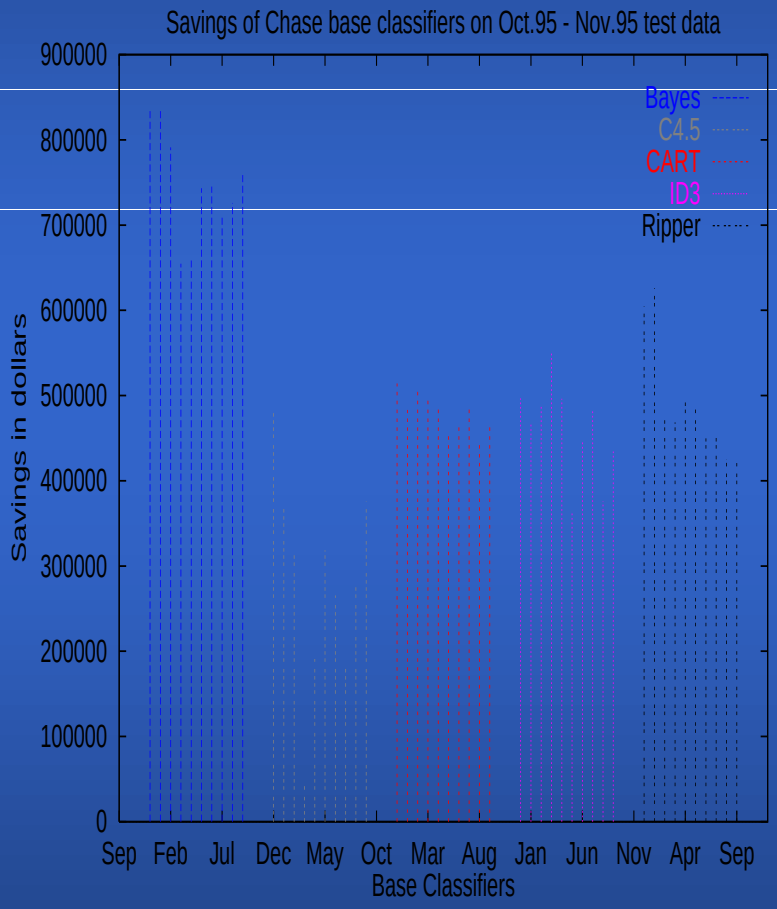


Chase

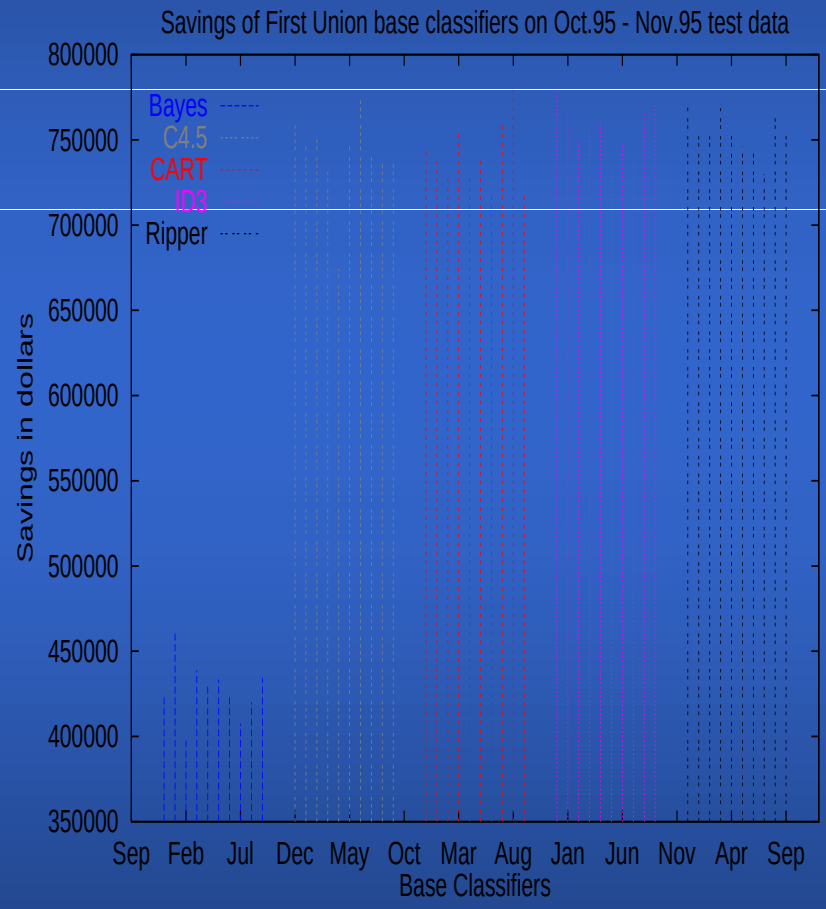


First Union

Cost Model (base classifiers)



Chase



First Union

Summary Results

- 6.2% higher TP-FP spread and $180K per month over the best single classifier for Chase
- 10% higher TP-FP spread and $140K per month over the best single classifier for First Union
- 2% higher TP-TP spread and $100K per month over the non-pruned meta-classifier for Chase and 60% better throughput
- 1% higher TP-FP spread and $10K per month over the non-pruned meta-classifier for First Union and 100% better throughput

Discussion

- Chase classifiers are more diverse
  - same learning algorithms, more diverse data set
- All pruning methods tend to converge
- Convergence is faster in First Union
  - attributed to the reduced diversity
- Maximizing savings is not solved optimally
  - Few Chase classifiers catch expensive fraud
- Pruning may improve overall performance!

Conclusions

- Efficiency and scalability of DDM systems
  - System architecture (e.g. JAM system)
  - Data site (e.g. reduce complex meta-classifiers)
  - pre-training pruning, post-training pruning
- Classifier evaluation metrics
  - class specialty, diversity, coverage, accuracy
- Greedy pruning algorithms
  - diversity-based, coverage/specialty-based
- Positive experimental results

Future Plan

- Compare and/or combine with other approaches (SCANN, ROCCH, etc)
- Incorporate the cost of the classifiers in the pruning process
- Extend to multi-level meta-classifiers