Genetics-Based Machine Learning

Genetic and Evolutionary Computation Conference

Amsterdam, The Netherlands
July 06-10, 2013

Comparing Multi-objective and Threshold-moving ROC Curve Generation for a Prototype-based Classifier

Ricardo Aler (Universidad Carlos III de Madrid)
Julia Handl (University of Manchester)
Joshua D. Knowles (University of Manchester)

Contents

- Background: binary classification, prototype-based classifier, scorers, ROC curves, operating point selection
- Generating ROC curves:
  - Threshold-moving
  - Multi-objective
- Average expected cost
- Empirical Comparison
- Conclusion

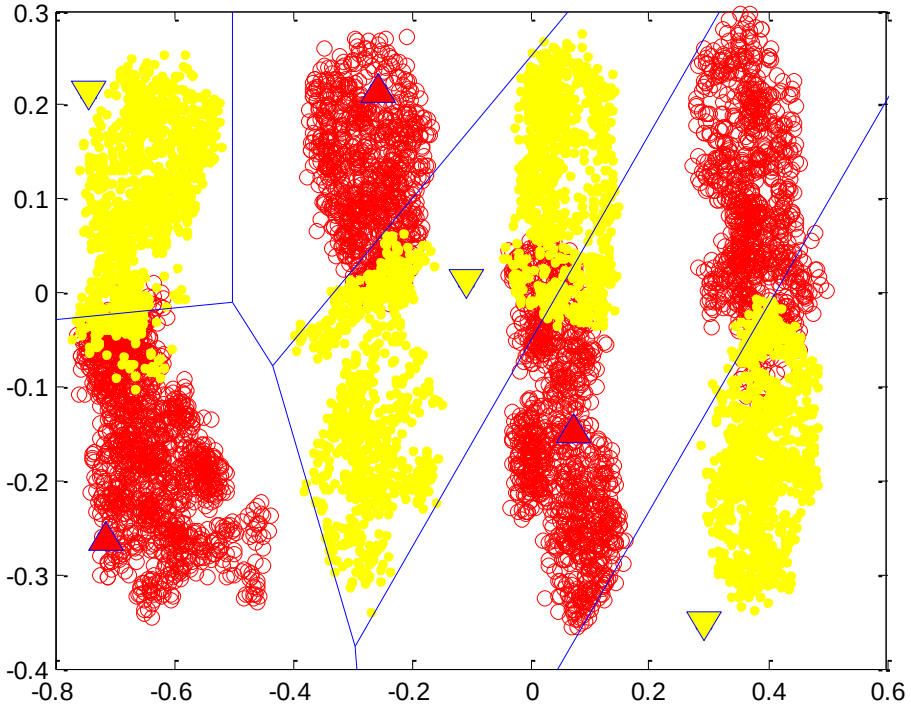
Introduction

- Two ways of generating ROC curves:
  - Threshold-moving (standard)
  - Multi-objective optimization
  - Which one is better?

- Contributions:
  - Empirical testing for a prototype-based classifier on synthetic and UCI classification domains
  - Proposed metric to compare ROC curves

Background: binary classification

- $c(x)$: $X \rightarrow \{C_0, C_1\}$
- Prototype-based classifier: it will be used in the rest of the paper



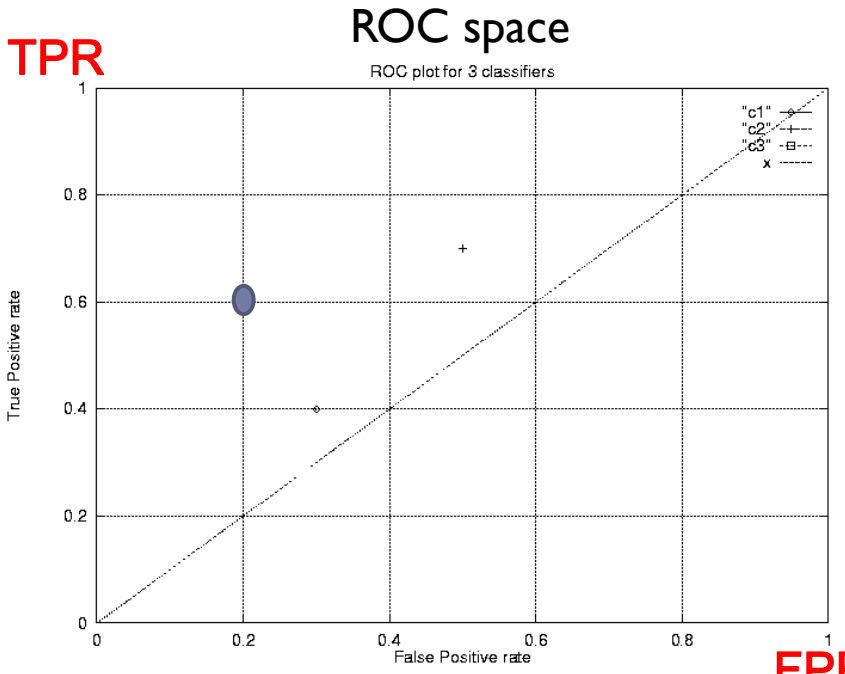
Background: Confusion matrix & ROC space

- c(x): $X \rightarrow \{C_0, C_1\}$
- True Positive Rate = Positive class success rate
- False Positive Rate = Negative class error rate

Predicted / Actual	Positive	Negative
Positive (C_1)	**TPR**=TP/pos	FNR=FN/pos
Negative (C_0)	**FPR**=FP/neg	TNR=TN/neg

pos= TP+FN
neg= FP+TN

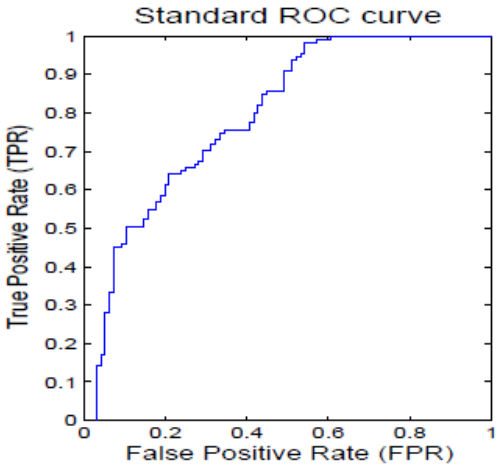
TPR

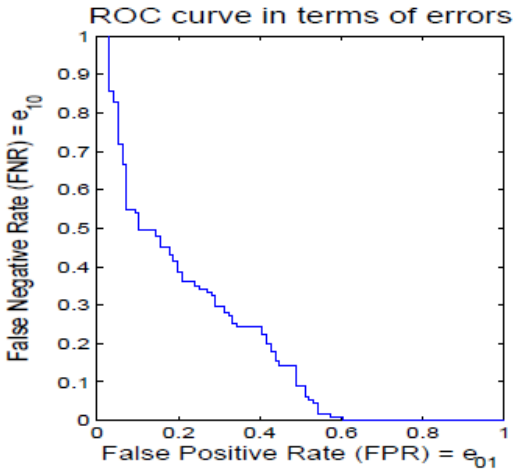


FPR

Background: from classifiers to scorers

- Scorer:
  - s(x): $X \rightarrow \mathbf{R}$
    - Score = degree of membership to the positive class (not necessaritly a probability)
  - A scorer can be transformed into a binary classifier by setting a threshold:
  - $s_t(x)$: If s(x)>t then C_1 else C_0
  - Moving the threshold results in a curve in ROC space

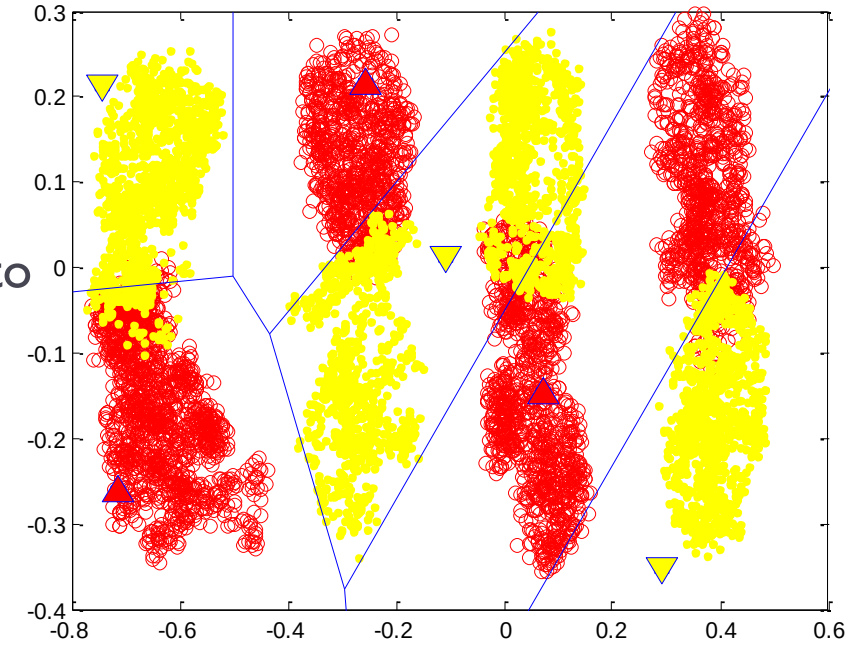




e_{10}= FNR = 1 – TPR
e_{01}=FPR

Prototype-based scorer

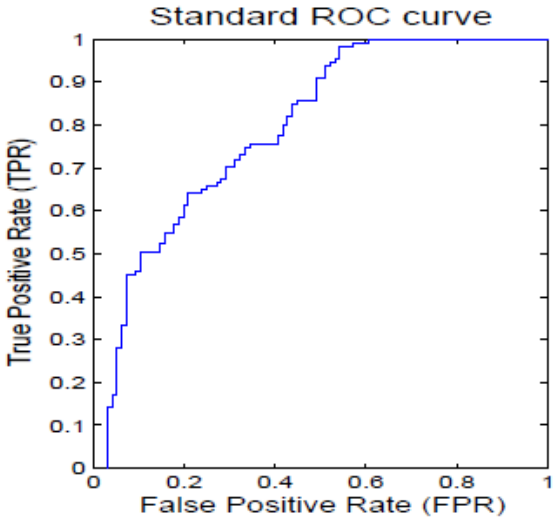
- $s(x)=d(x-p_0)-d(x-p_1)$
  - P_1: closest positive prototype
  - P_0: closest negative prototype
  - $s(x) > t = 0$ is the standard classifier
  - $s(x) > t > 0$ moves boundaries closer to the positive class:
    - TPR increases
    - FPR decreases

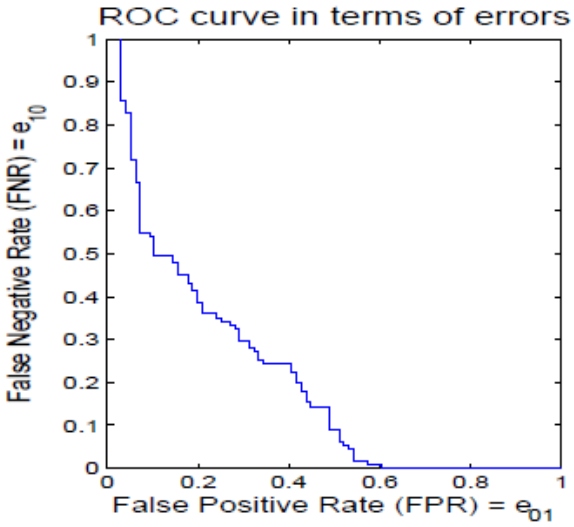


- E. Mwebaze, P. Schneider, F.M. Schleif, JR Aduwo, JA Quinn, S. Haase, T. Villmann, and M. Biehl. Divergence-based classification in learning vector quantization.
  Neurocomputing, 74(9):1429–1435, 2011.

Usefulness of scorers and ROC curves

- Scorers can be adapted by setting the threshold: if it is more important to classify correctly positive instances, then set a large threshold
- Receiver Operating Characteristic (ROC) curves are an important tool in classification problems, especially in applications that involve dynamic or imprecise operating conditions, where the class distributions and the misclassification costs of future data may differ from those of the data available at training time





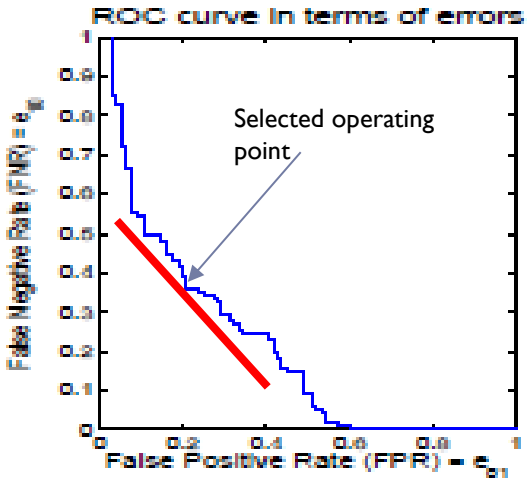
Selecting the operating point according to the operating conditions by minimizing cost

- ROC curve = $\{s_t(x) \mid t \in \mathbf{R}\}$
- Error of s_t : $\boldsymbol{e_t} = (e_{t,01}, e_{t,10})$
- Operating conditions:
  - Class probabilities: $\boldsymbol{P}=(P_0, P_1)$
  - Missclassification costs: $\boldsymbol{C}=(C_{01}, C_{10})$
- Cost:

$$Cost(\mathbf{e}, \mathbf{P}, \mathbf{C}) = e_{10} P_1 C_{10} + e_{01} P_0 C_{01}$$

- Once the operating conditions become known, the optimal threshold (operating point) can be selected from the ROC curve

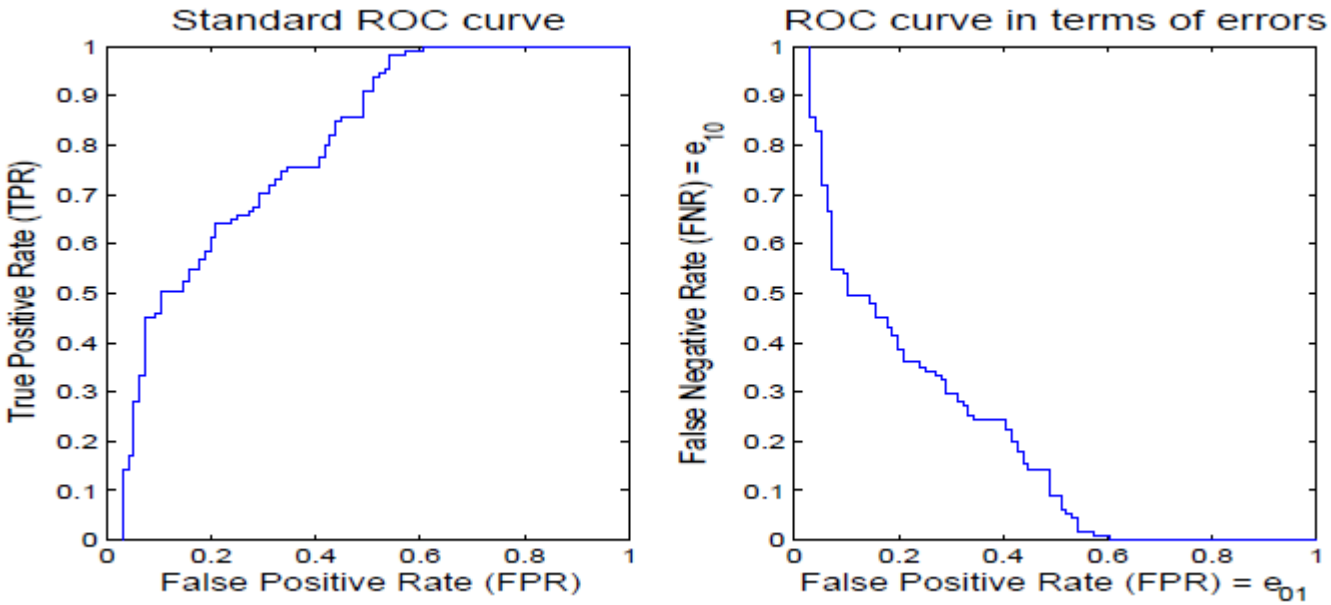
$$\mathbf{e}_{(\mathbf{P},\mathbf{C})} = \arg\min_{\mathbf{e} \in \mathrm{ROC}} Cost(\mathbf{e}, \mathbf{P}, \mathbf{C}).$$



- Generating ROC curves:
  - Threshold-moving
  - Multi-objective

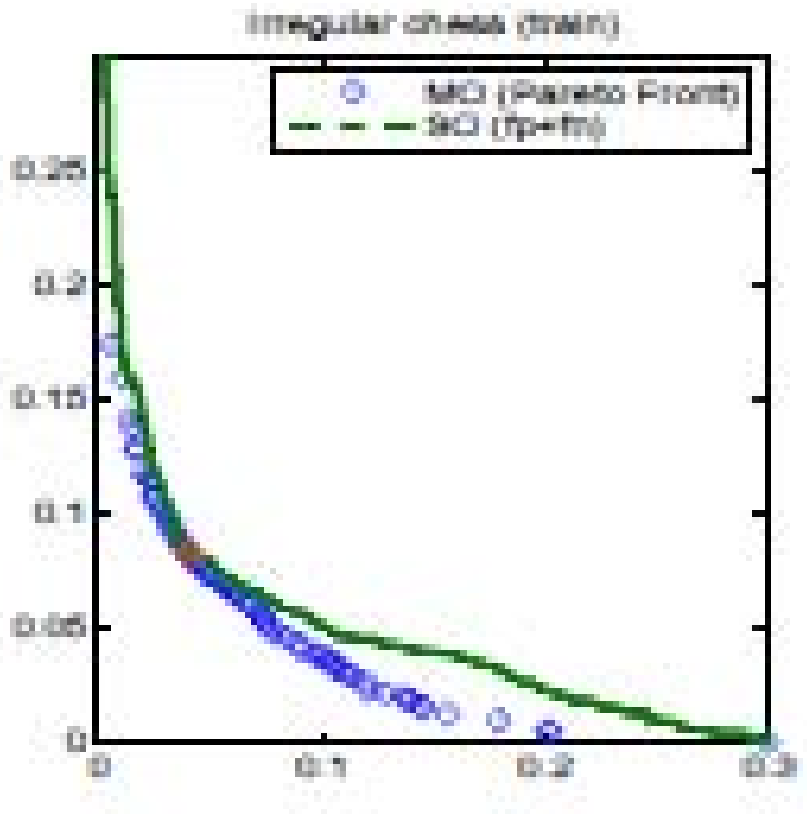
Threshold-moving ROC

- Threshold-moving. Two steps:
  1. A classifier capable of ranking or scoring instances is trained for a particular set of operating conditions, using a standard learning algorithm.
  2. The output threshold of the classifier is varied to obtain a range of different points (TPR, FPR) in ROC space



Generating ROC curves by multi-objective optimization

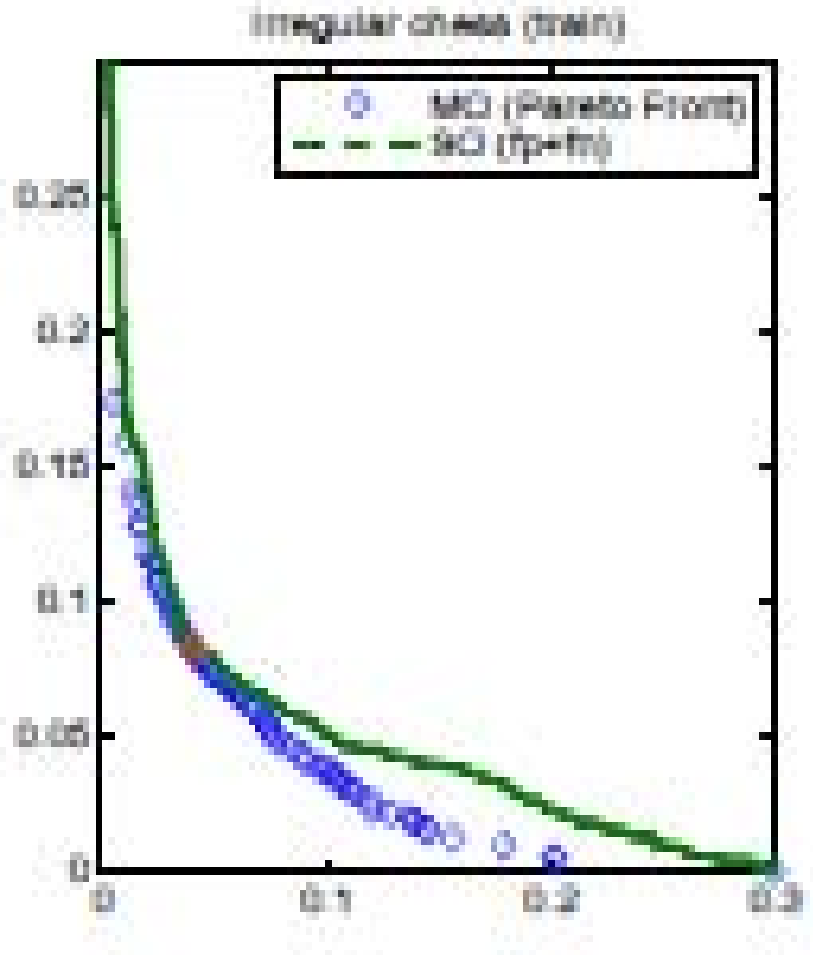
- Points in ROC space obey Pareto dominance:
  - Kupinski & Anastasio. Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves. *IEEE Transactions on Medicc Imaging*, 18:675–685, 1999
  - Everson & Fieldsend. Multi-class ROC analysis from a multi-objective optimisation perspective. *Pattern Recognition Letters*. 2006
- Objective space = ROC space = (FNR, FPR) = (e_{10},e_{01})
- Decision space = w = prototype locations
- The result is not one classifier (i.e. one set of prototypes), like in threshold moving, but a non-dominated set of classifiers



Green: Threshold-moving
Blue: Multi-objective (Pareto front)

Research question

- Let's optimize a prototype-based system for some operating conditions and generate a ROC curve by threshold-moving
- Is this ROC curve better or worse than the non-dominated set of classifiers generated by Multi-objective optimization?
- If it is not, then threshold-moving should be preferred

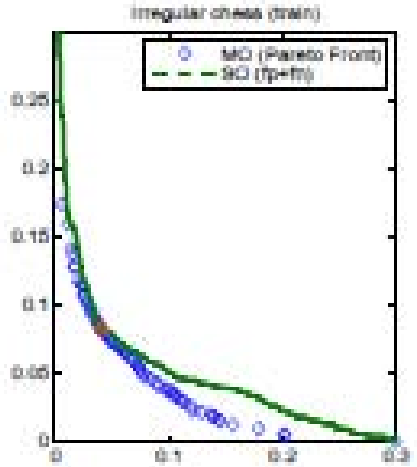


Threshold-moving operating conditions

- Let's optimize a prototype-based system for some operating conditions and generate a ROC curve by threshold-moving
- Which operating conditions?
  - It will be assumed equal costs and equal class distributions so that both classes are given the same weight (thus, problems with imbalanced domains are avoided)
    - $\boldsymbol{P}=(P_0, P_1) = (0.5, 0.5)$, $\boldsymbol{C}=(C_{01}, C_{10})= (1,1)$
- Which loss function to be optimized?
  - $Cost(e, P, C) = e_{10}P_1C_{10} + e_{01}P_0C_{01} = (e_{10}+e_{01})/2$
    - Note: classification error = $e_{10}P_1 + e_{01}P_0$
  - The solution will be named fn+fp

Threshold-moving operating conditions

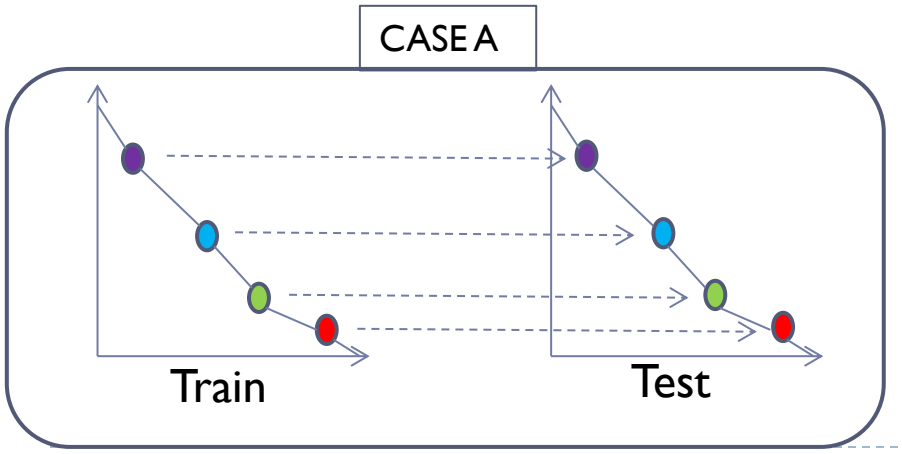
- If fn+fp = $e_{10} + e_{01}$ is optimal, then (e_{10} , e_{01}) should belong to the Pareto Front
  - therefore instead of running a multi-objective optimizer to get the Pareto Front, and a single-objective optimizer (such as a genetic algorithm) to get the fn+fp solution …
  - only the multi-objective optimizer will be run, the PF will be obtained, and then the fn+fp solution will be extracted from the PF

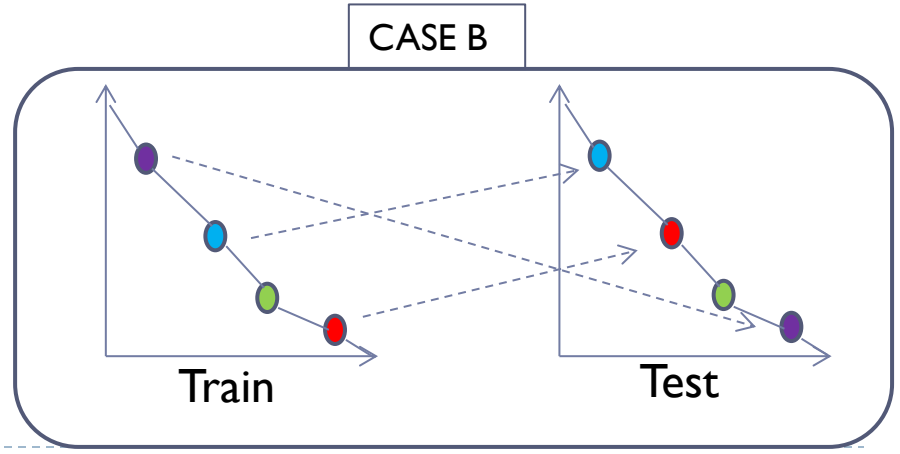


Comparing ROC curves

- A commonly used scalar measure is the Area Under the Curve (AUC)
- Process:
  1. The ROC curve is learned on training data. ROC curve is a set of classifiers (one for each threshold in the threshold-moving case, one for each point in the MO case)
  2. Each classifier in the ROC curve is then computed on test data
  3. The AUC of the testing ROC curve is computed

1. Problem: testing AUC does not distinguish between matching and non-matching ROC curves (Cases A and B have the same testing AUC but Case A matches train and test while Case B does not)





New metric: average expected cost (AEC)

- Based on the area under the cost curve defined in:

  Chris Drummond and Robert C. Holte. Cost curves: *An improved method for visualizing classifier performance*. Machine Learning, 65(1):95–130, 2006.

- Based on how ROC curves are used for operating-point selection based on cost-minimization
- For all operating conditions $\boldsymbol{P}=(P_0, P_1)$, $\boldsymbol{C}=(C_{01}, C_{10})$ do:
  - Select operating point (threshold or classifier) with error $\mathbf{e_{PC}}$ on the training set, that minimizes cost:

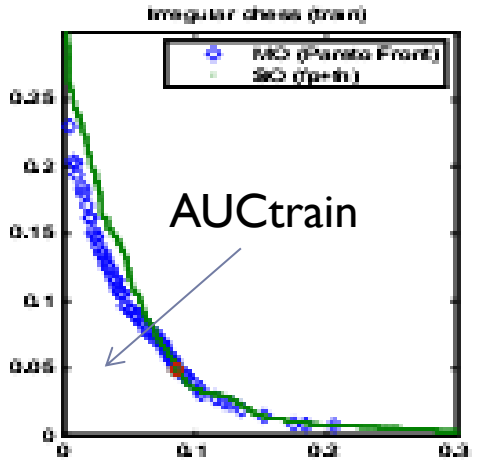
$$\mathbf{e}_{(\mathbf{P},\mathbf{C})} = \arg\min_{\mathbf{e}\in \mathrm{ROC}} Cost(\mathbf{e}, \mathbf{P}, \mathbf{C}).$$

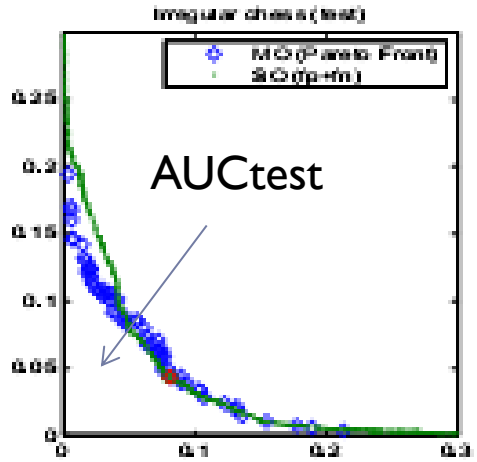
  - Now, compute the error of the selected operating point (threshold or classifier) on the test data: $\hat{\mathbf{e}}_{\mathbf{PC}}$
- Average all $\hat{\mathbf{e}}_{\mathbf{PC}}$ assuming all operating conditions are equally likely:

$$AEC = E_{(\mathbf{P},\mathbf{C})}[Cost(\hat{\mathbf{e}}_{(\mathbf{P},\mathbf{C})}, \mathbf{P}, \mathbf{C})]$$

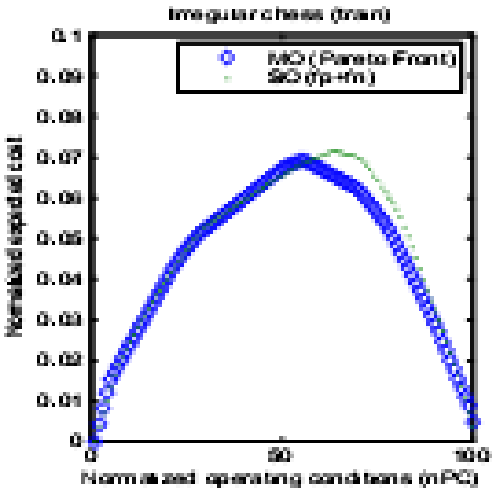
AEC and area under the cost curve

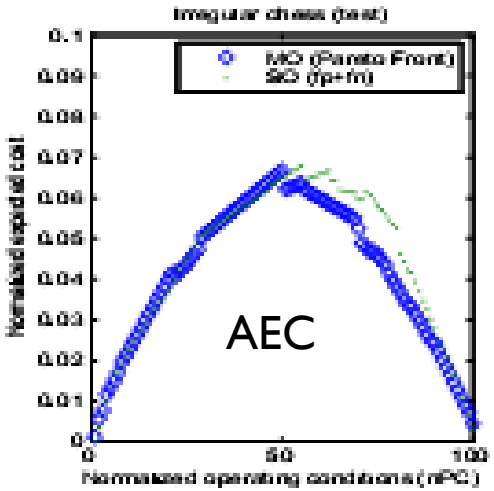
ROC curves





Cost curves





$$AEC = E_{(\mathbf{P},\mathbf{C})}[Cost(\hat{\mathbf{e}}_{(\mathbf{P},\mathbf{C})}, \mathbf{P}, \mathbf{C})]$$

Empirical testing

- Synthetic domains:

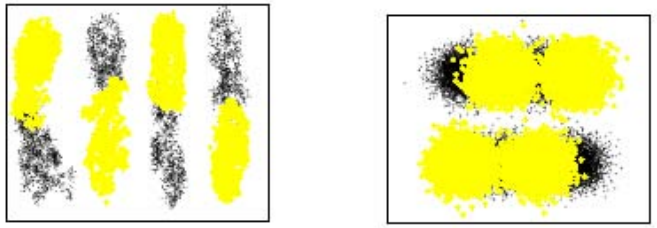


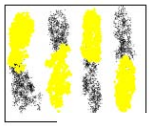
Figure 2: Irregular chess domain (left) and Gaussian chess domain (right).

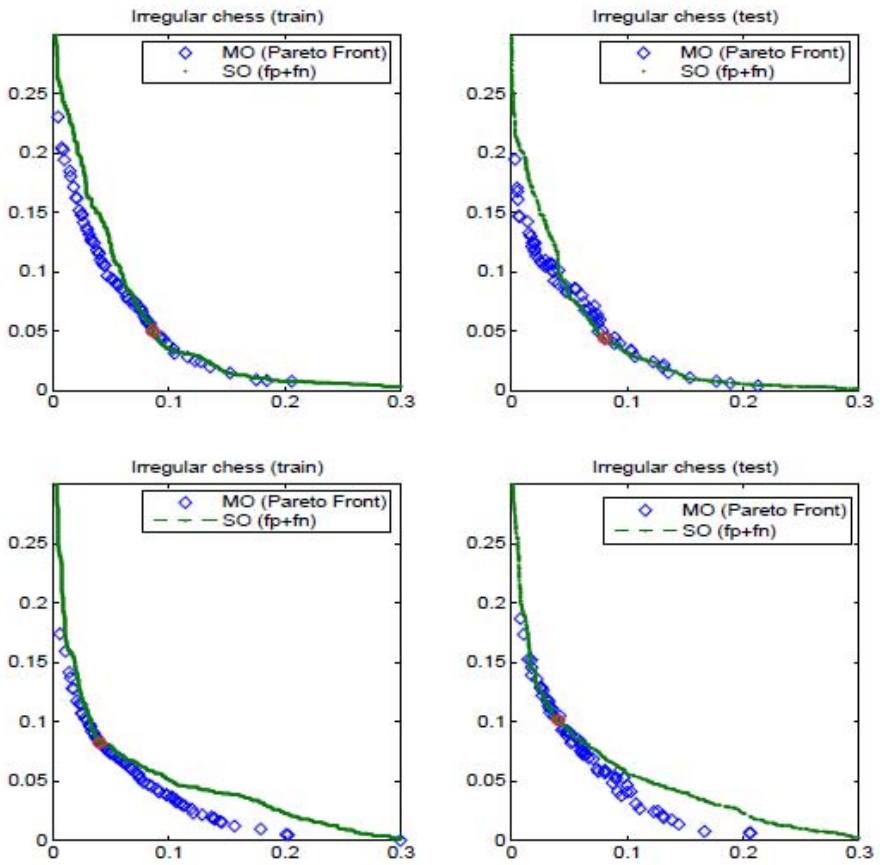
- UCI domains:
  - Blood: 748 instances / 5 attributes
  - Diabetes: 768 instances / 8 attributes
  - German.numer: 1000 instances / 24 attributes
  - Estate: 5322 instances / 12 attributes

Experimental setting

- Distance-based methods are sensitive to noisy / irrelevant attributes
  - Feature selection: Correlation Feature Selection (CFS) + Best First
- Multiobjective optimization: MSOPS-II (Multiple Single Objective Pareto Sampling) [Hughes 2007]
  - General-purpose many-objective optimiser
  - Minimal initial configuration
  - 25% of the initial population seeded with GLVQ (gradient-descent LVQ)
- Experimentation:
  - Synthetic domains: 40% training / 60% testing, repeated 5 partitions x 5 random seeds (25 runs)
  - UCI domains: 5x2(x2) Crossvalidation (20 runs)

Threshold-moving ROC vs. Multi-objective Pareto Front





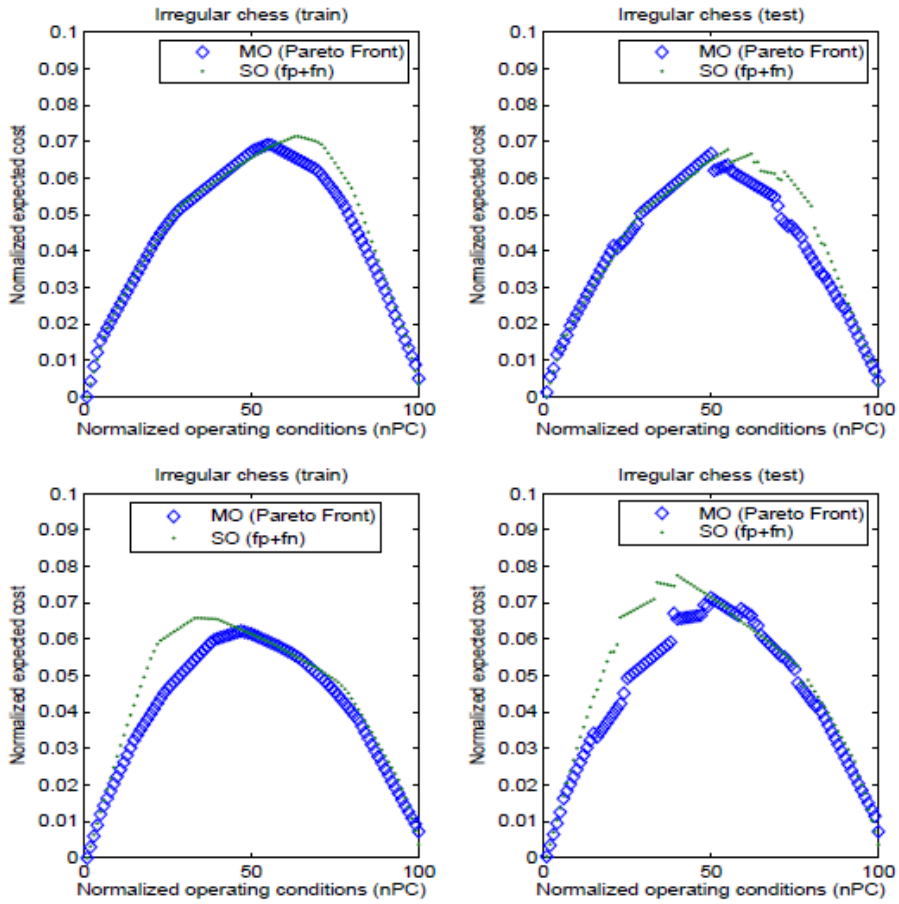
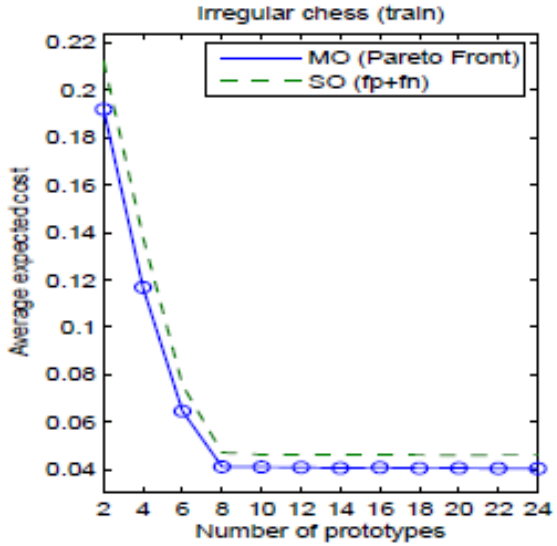
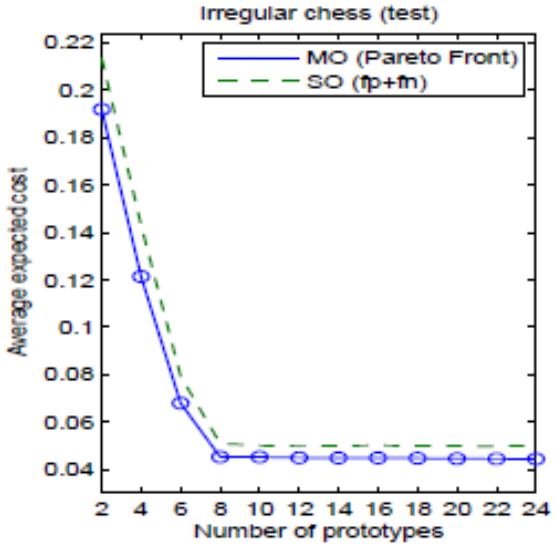


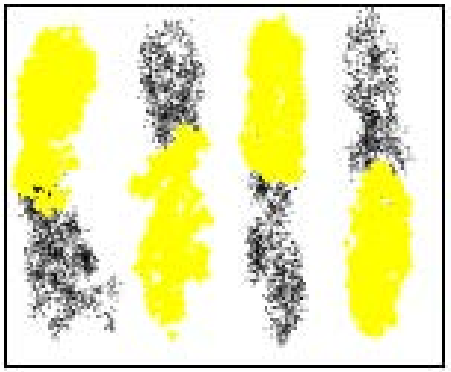
Figure 4: MO (diamonds) vs SO (dots) for the Irregular Chess domain with 12 prototypes. Category 1 (top). Category 2 (bottom).

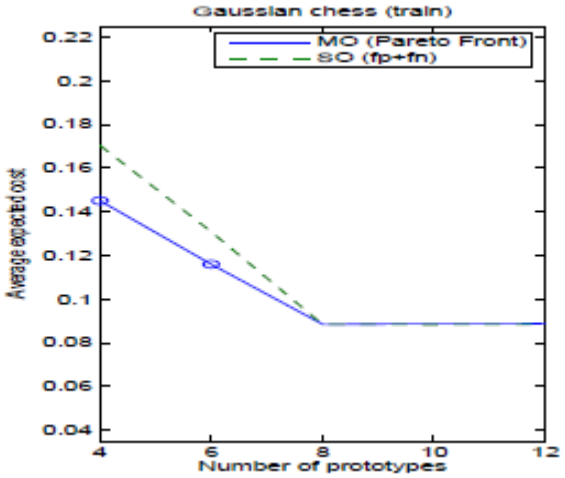
Figure 5: Cost curves corresponding to ROC curves of figures in Figure 4. Category 1 (top). Category 2 (bottom).

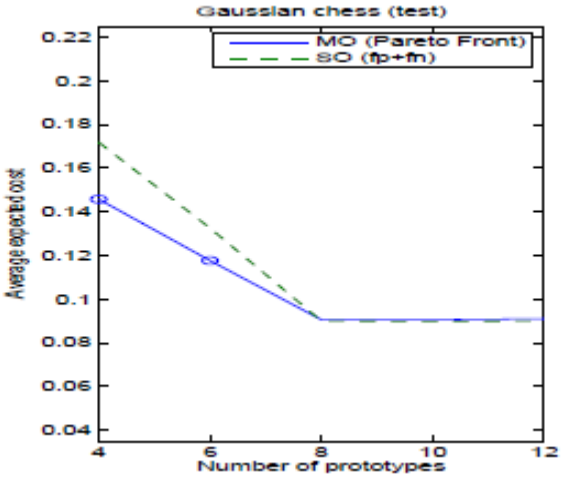
Average expected cost. Synthetic domains

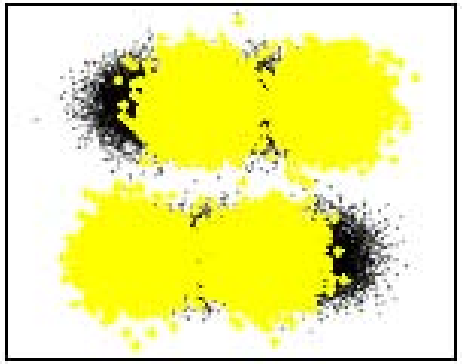




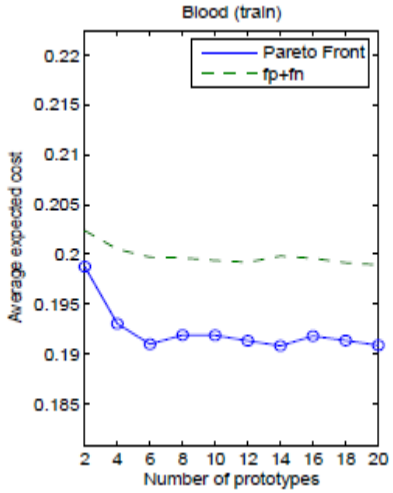


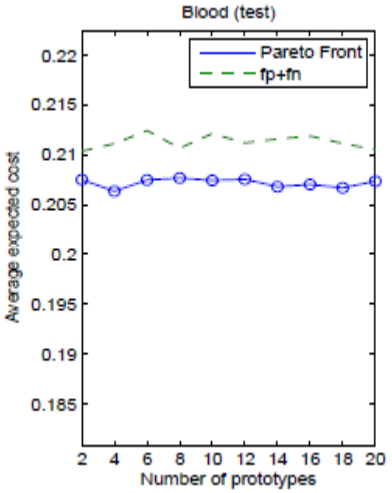


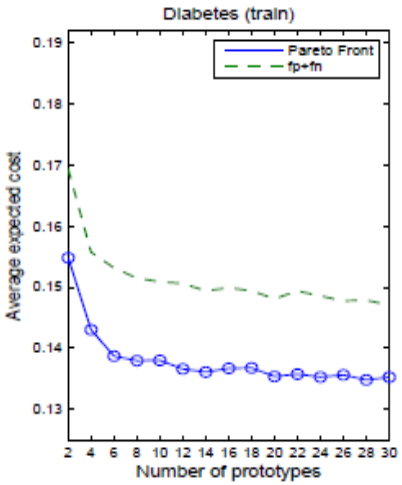


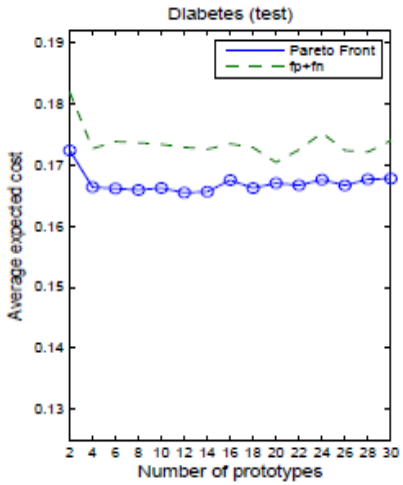


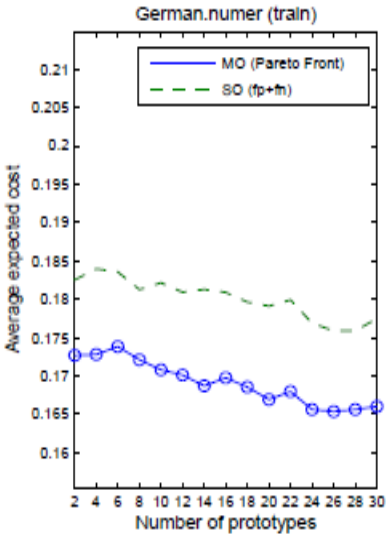
Average expected cost. UCI domains

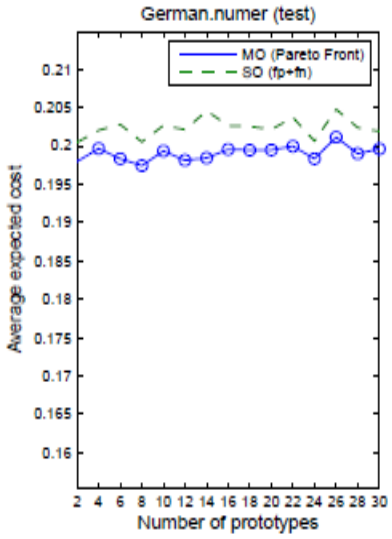


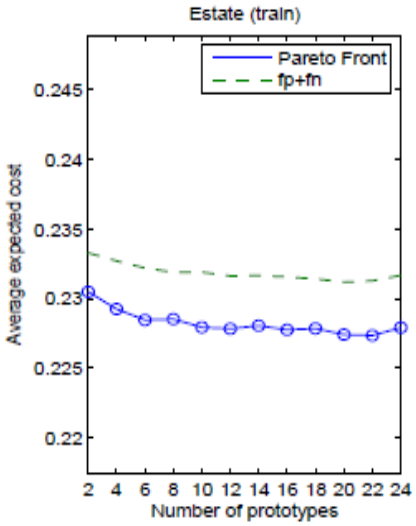


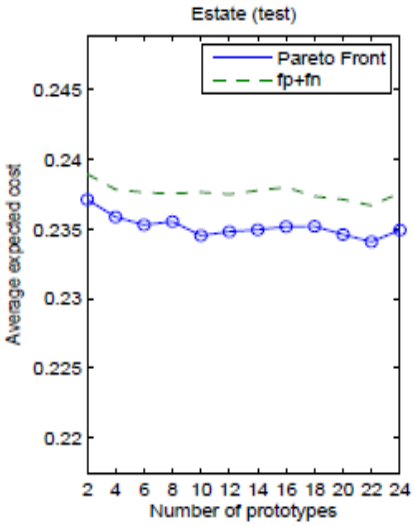












Conclusions

- Empirical comparison between threshold-moving and multi-objective optimization for generating ROC curves
- New metric based on cost-based classification and area under the cost curve
- Future:
  - Other classifiers (mixtures of gaussians, neural networks, …)
  - Multi-class
  - …

¿Questions?