Model Combination in Multiclass Classification

Sam Reid
Advisors: Mike Mozer, Greg Grudic

Department of Computer Science
University of Colorado at Boulder
USA

April 5, 2010

Multiclass Classification

- From examples, make multiclass predictions on unseen data.
- Applications in:
  - Heartbeat arrythmia monitoring
  - Protein structure classification
  - Handwritten digit recognition
  - Part of speech tagging
  - Vehicle identification
  - Many others...
- Our approach: model combination

Multiclass Classification: Example

Heartbeat Arrhythmia Monitoring Data Set (truncated)

age (yrs)	gender	height (cm)	weight (kg)	BPM	QRS duration (ms)	274 other wave characteristics	class
75	m	190	80	91	63	...	Supraventricular Pre.
56	f	165	64	81	53	...	Sinus bradycardy
54	m	172	95	138	75	...	Right bundle block
55	m	175	94	100	71	...	normal
75	m	190	80	88	?	...	Ventricular Pre.
13	m	169	51	100	84	...	Left ventricule hyper.
40	f	160	52	77	70	...	normal
49	f	162	54	78	67	...	normal
44	m	168	56	84	64	...	normal
50	f	167	67	89	63	...	Right bundle block
...	...	...	...	...	...	...	...
62	m	170	72	102	70	...	?
45	f	165	86	77	72	...	?

Model Combination

- Combine multiclass classifiers (e.g. KNN, Decision Trees, Random Forests)
  - Voting
  - Averaging
  - Linear
  - Nonlinear
- Combine binary classifiers (e.g. SVM, AdaBoost) to solve multiclass
  - One vs. All
  - Pairwise Classification
  - Error Correcting Output Coding

Outline

Outline

Classifier Combination

- Goal: optimize predictions on test data
- Maintain diversity without sacrificing accuracy
- Train many classifiers with different algorithms/hyperparameters
- Combine with a linear combination function
  - Ting & Witten, 1999
  - Seewald, 2002
  - Caruana et al., 2004

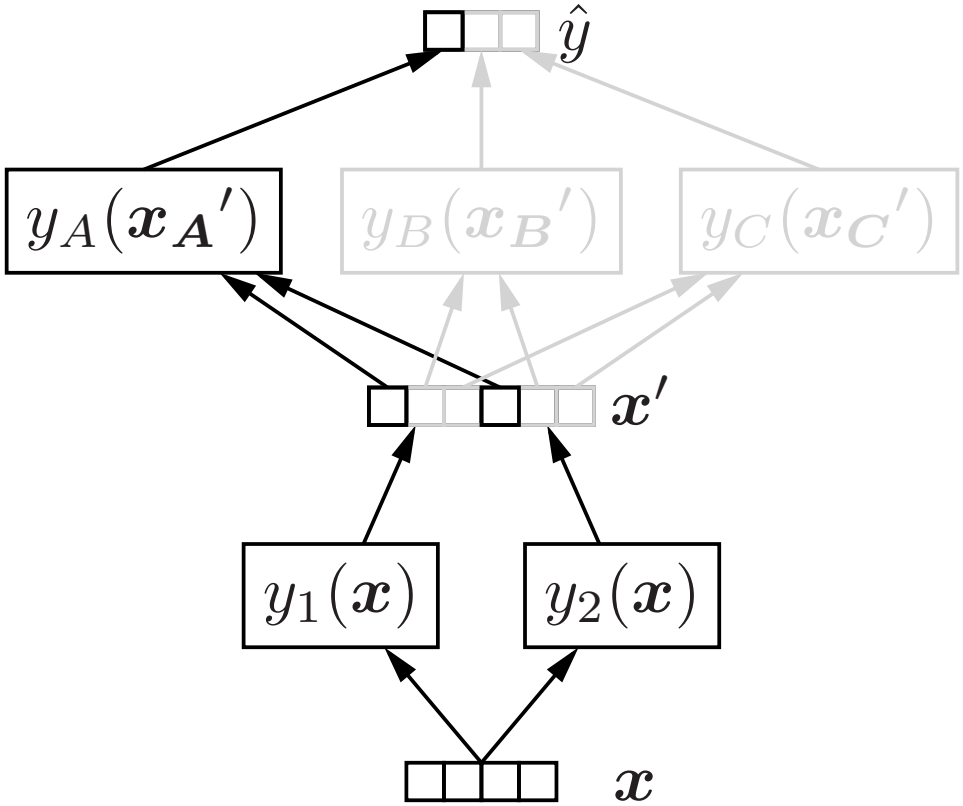
Linear StackingC 1/2

- Stacked Generalization
  - Predictions on validation data are meta-training data
- Linear StackingC, class-conscious stacked generalization

$$\hat{p}_j(\vec{x}) = \sum_{i=1..L} w_{ij} y_{ij}(\vec{x})$$

- $\hat{p}_j(\vec{x})$ is the predicted probability for class c_j
- w_{ij} is the weight corresponding to classifier y_i and class c_j
- $y_{ij}(\vec{x})$ is the i^{th} classifier's output on class c_j
- Training set = classifier predictions on unseen data + labels
- Determine weights using linear regression

Linear StackingC 2/2



Problems

- Caruana et al., 2004: "Stacking [linear] performs poorly because regression overfits dramatically when there are 2000 highly correlated input models and only 1k points in the validation set."
- How can we scale up stacking to a large number of classifiers?

Problems

- Caruana et al., 2004: "Stacking [linear] performs poorly because regression overfits dramatically when there are 2000 highly correlated input models and only 1k points in the validation set."
- How can we scale up stacking to a large number of classifiers?
- Our hypothesis: regularized linear combiner will
  - reduce variance & prevent overfitting on indicator subproblems
  - increase accuracy on multiclass problem
- Penalty terms in our studies:
  - Ridge Regression: $L = |y - X\beta|^2 + \lambda|\beta|^2$
  - Lasso Regression: $L = |y - X\beta|^2 + \lambda|\beta|_1$
  - Elastic Net Regression: $L = |y - X\beta|^2 + (1-\alpha)|\beta|^2 + \alpha|\beta|_1$

Thesis Statement - Part I

- In linear combinations of multiclass classifiers, regularization significantly improves performance.

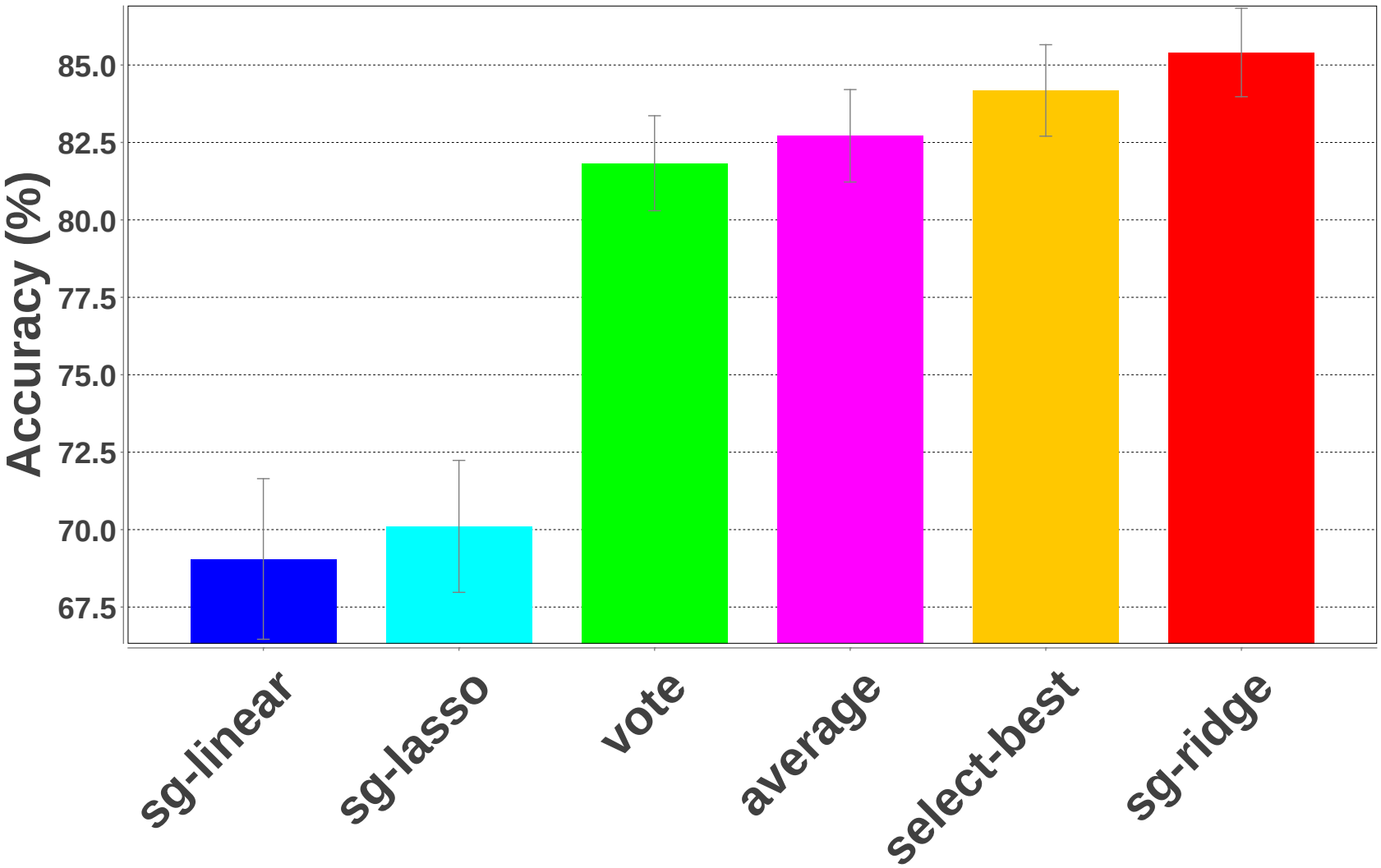
Multiclass Classification Data Sets

Dataset	*Att.(numeric)*	*Instances*	*Classes*
balance-scale	4	625	3
glass	9	214	6
letter	16	4000	26
mfeat-morphological	6	2000	10
optdigits	64	5620	10
sat-image	36	6435	6
segment	19	2310	7
vehicle	18	846	4
waveform-5000	40	5000	3
yeast	8	1484	10

Algorithms

- About 1000 base classifiers for each problem
  1. Neural Network
  2. Support Vector Machine (C-SVM from LibSVM)
  3. K-Nearest Neighbor
  4. Decision Stump
  5. Decision Tree
  6. AdaBoost.M1
  7. Bagging classifier
  8. Random Forest (Weka)
  9. Random Forest (R)

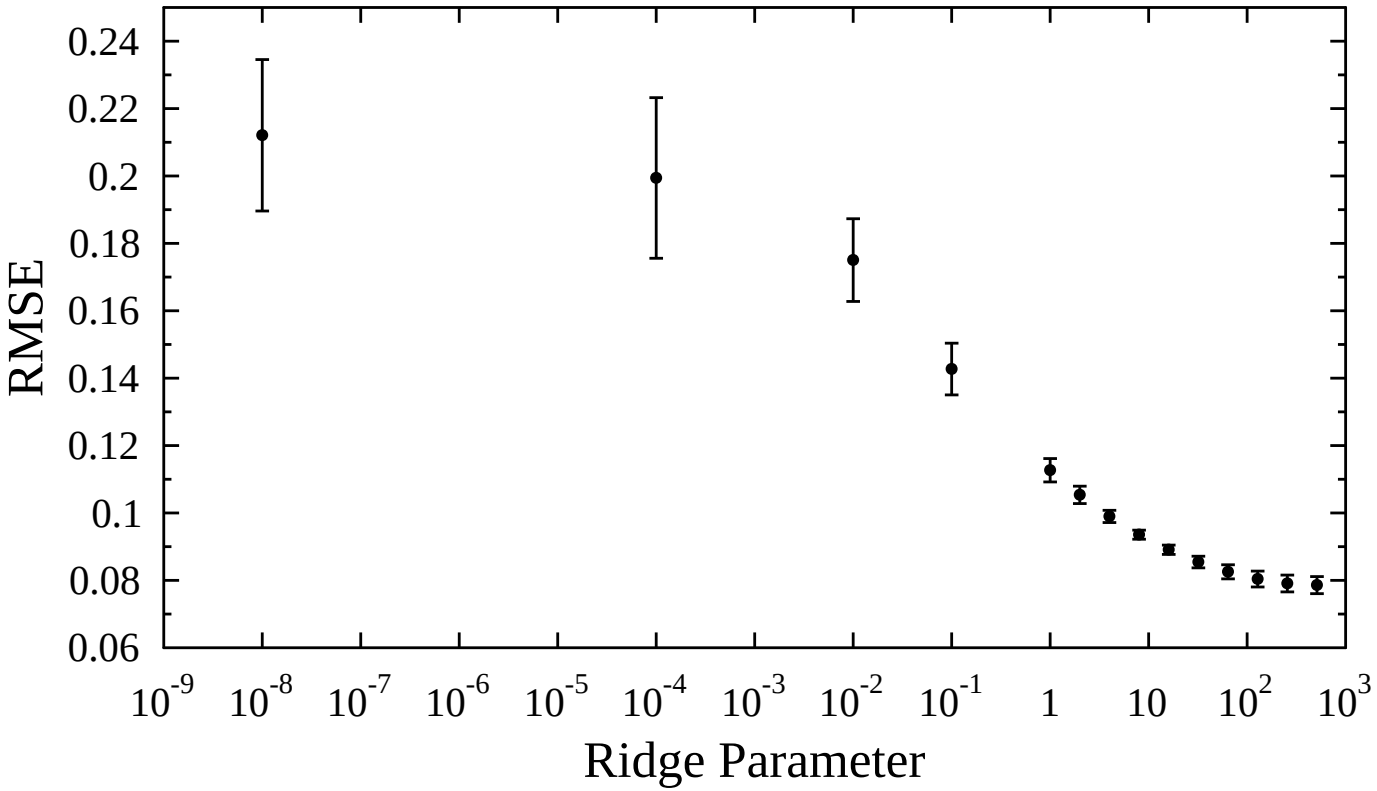
Results: Average Accuracy



Statistical Analysis

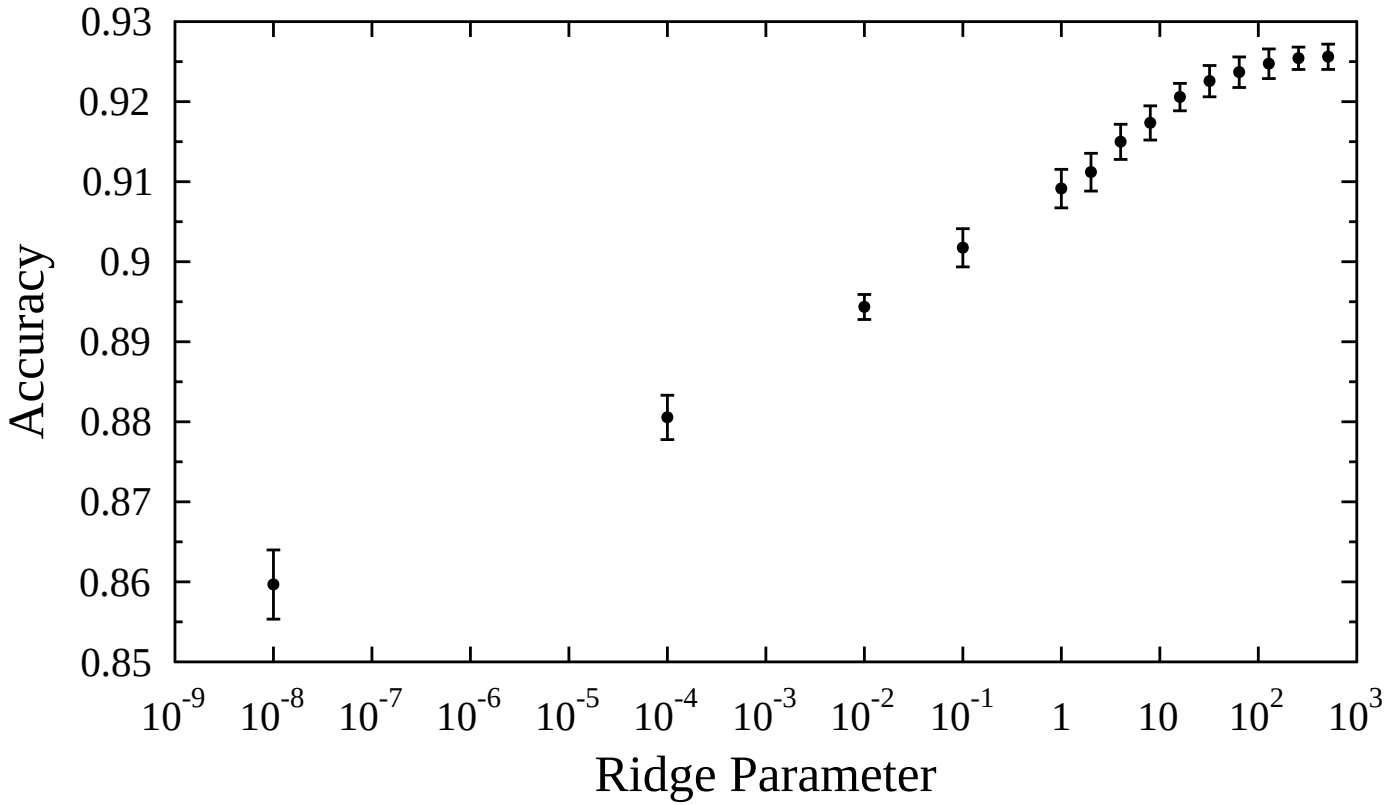
- Ridge outperforms unregularized at $p \leq 0.002$
  - Validates hypothesis: regularization improves accuracy
- Ridge outperforms lasso at $p \leq 0.0019$
  - Dense better than sparse
- Voting and averaging all models not competitive

Multiclass Accuracy $\propto$ Binary Accuracy 1/3



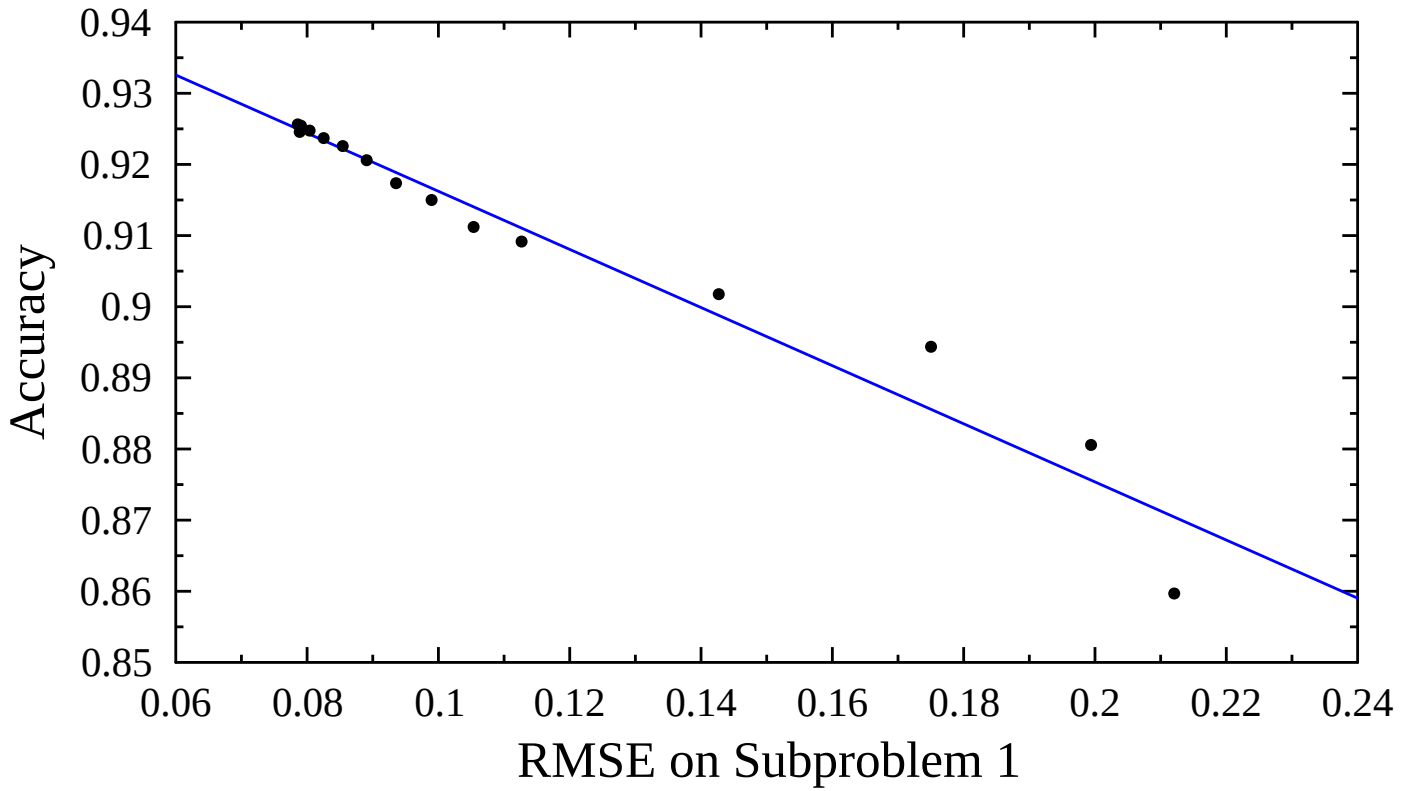
Root mean squared error for the first (class-1) indicator subproblem in sat-image, over 10 folds of Dietterich's 5x2 CV.

Multiclass Accuracy $\propto$ Binary Accuracy 2/3



Multiclass classification accuracy as a function of the regularization hyperparameter λ_{ridge}.

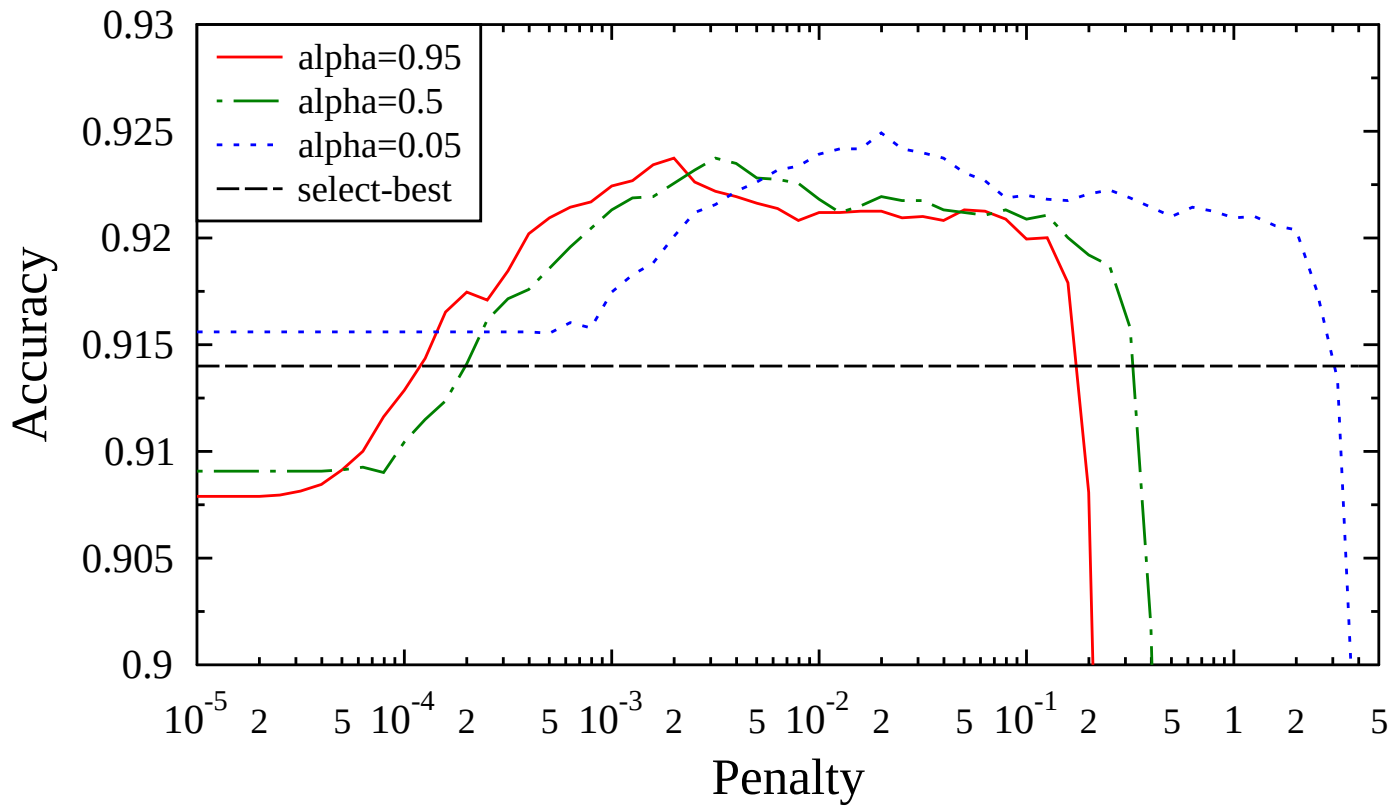
Multiclass Accuracy $\propto$ Binary Accuracy 3/3



Accuracy vs RMSE on the first (class-1) indicator subproblem.

Multiclass Accuracy $\propto$ Binary Accuracy

Ridge More Effective than Lasso

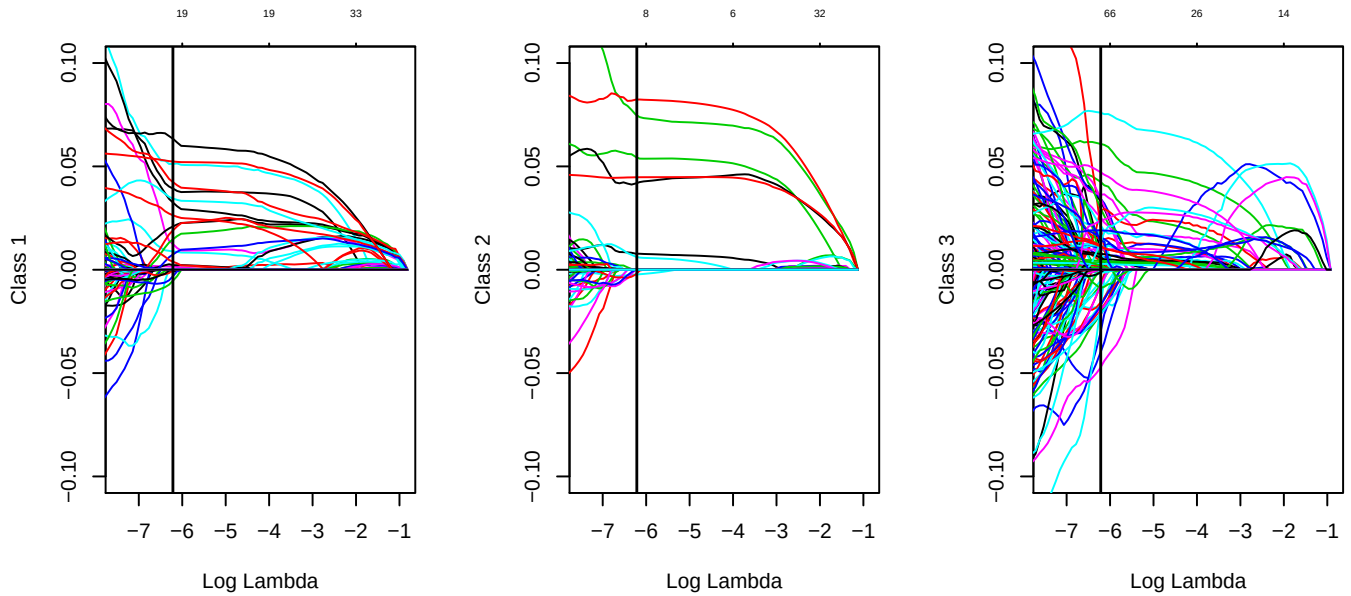


Overall accuracy on sat-image with various parameters for elastic-net.

Focus on Subproblems

- Choose from classifiers and predictions
- Allow classifiers to focus on subproblems
- Example: Benefit from a classifier that predicts well-calibrated probabilities for class A but has B & C backwards
- This advantage possible on multiclass classification but not binary classification, since $\sum_{i=1}^{k} p_i(\vec{x}) = 1$

Sparse Linear Combinations



Coefficient profiles for the first three subproblems in StackingC for the *sat-image* dataset with elastic net regression at $\alpha = 0.95$

Selected Classifiers

Classifier	*red*	*cotton*	*grey*	*damp*	*veg*	*v.damp*	*total*
adaboost-500	6.3	0	1.4	0	2.3	0	10.0
ann-0.5-32-1000	0	0	6.1	3.5	0	0.4	10.0
ann-0.5-16-500	3.9	0	0	1.8	0.9	3.4	10.1
ann-0.9-16-500	0.2	8.2	0	0	0.7	1.6	10.8
ann-0.5-32-500	0.0	7.5	0	10.0	2.7	0	11.1
knn-1	0	0	7.6	6.5	0.8	9.7	24.6

Weights (%) for the *sat-image* problem in elastic net StackingC with $\alpha = 0.95$ for the 6 models with highest total weights.

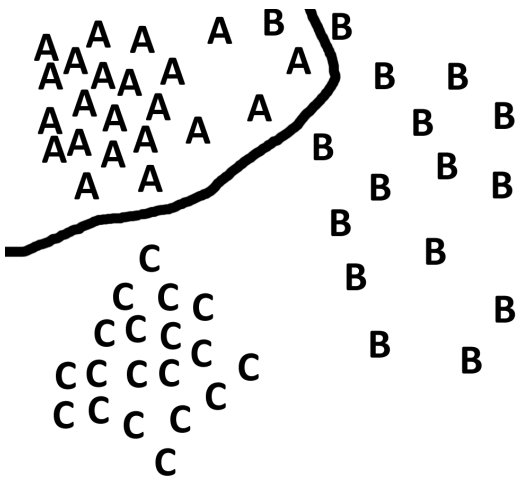
Conclusions & Future Work

- Regularization is essential in linear combinations of multiclass classifiers
- Dense combiners outperform sparse combiners
- One-weight-per-output (instead of one-weight-per-classifier) allows classifiers to specialize in subproblems
- Future Work
  - Bayesian treatment, Gaussian/Laplacian priors over weights
  - Constrain coefficients to be positive
- This work published as:
  - *Regularized Linear Models in Stacked Generalization*, Sam Reid and Greg Grudic, Multiple Classifier Systems, 2009, Springer LNCS 5519 112-121

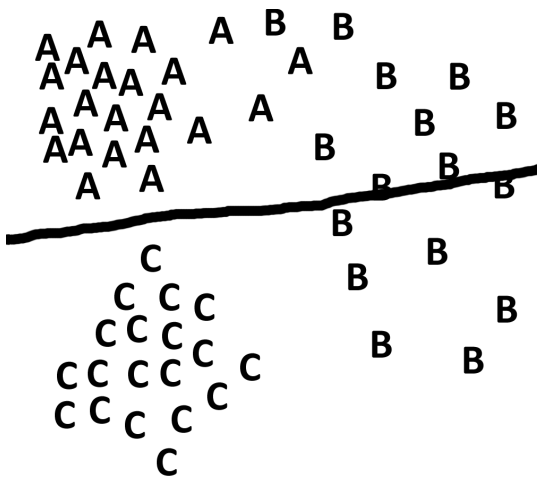
Outline

Reducing Multiclass to Binary

- Some classifiers designed for binary (e.g. SVM, Adaboost)
- Transform multiclass $\Rightarrow$ set of binary problems
- Combine binary predictions $\Rightarrow$ predict multiclass



A vs B,C in one-vs-all



A vs C in all-pairs

Model Selection in Reducing Multiclass to Binary

- No Model Selection
  - Dietterich and Bakiri, 1995
  - Allwein et al., 2000

Model Selection in Reducing Multiclass to Binary

- No Model Selection
  - Dietterich and Bakiri, 1995
  - Allwein et al., 2000
- Shared Hyperparameters
  - Rifkin uses greedy 1d hillclimbing, with OVA + LBD, Rifkin & Klautau, 2004
  - Model selection in LibSVM, Chang & Lin, 2001

Model Selection in Reducing Multiclass to Binary

- No Model Selection
  - Dietterich and Bakiri, 1995
  - Allwein et al., 2000
- Shared Hyperparameters
  - Rifkin uses greedy 1d hillclimbing, with OVA + LBD, Rifkin & Klautau, 2004
  - Model selection in LibSVM, Chang & Lin, 2001
- Optimize Subproblems Independently
  - Homogeneous, Friedman 1996
  - Heterogeneous, Szepannek et al. 2007

Model Selection in Reducing Multiclass to Binary

- No Model Selection
  - Dietterich and Bakiri, 1995
  - Allwein et al., 2000
- Shared Hyperparameters
  - Rifkin uses greedy 1d hillclimbing, with OVA + LBD, Rifkin & Klautau, 2004
  - Model selection in LibSVM, Chang & Lin, 2001
- Optimize Subproblems Independently
  - Homogeneous, Friedman 1996
  - Heterogeneous, Szepannek et al. 2007
- Optimize the Joint Distribution
  - Evolutionary search, de Souza et al., 2006, Lebrun et al., 2007

Shared Hyperparameters vs Independent Optimization

- Shared Hyperparameters
  - Optimizes to the target multiclass metric
  - Increases bias and reduces variance for model selection
- Independent Optimization
  - Accommodate subproblems with different structure
  - Improved subproblem performance $\Rightarrow$ improved performance

Thesis Statement - Part II

- When solving a multiclass problem with a set of binary classifiers, it is more effective to constrain subproblems to use the same hyperparameters than to optimize each independently.

Multiclass Classification Data Sets 1/2

dataset	classes	numeric	train	test	sampled-from
anneal	4	6	300	150	878
arrhythmia	5	206	257	129	386
authorship	4	70	300	150	841
autos	5	15	134	68	202
cars	3	6	270	136	406
collins	11	19	300	150	451
dj30-1985-2003	20	6	133	67	138123
ecoli	4	7	204	103	307
eucalyptus	5	14	300	150	736
halloffame	3	15	300	150	1340

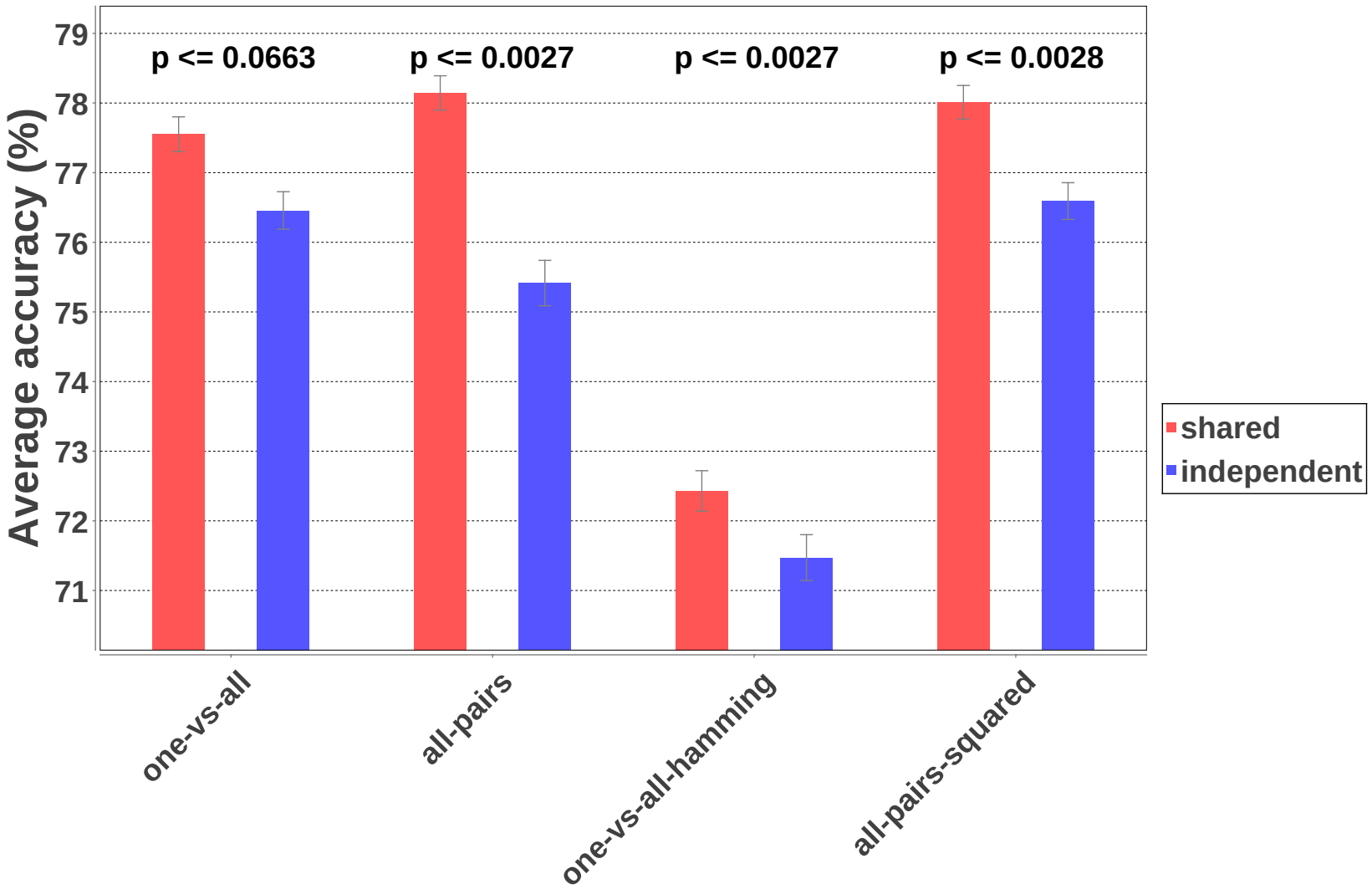
Multiclass Classification Data Sets 2/2

dataset	classes	numeric	train	test	sampled-from
hypothyroid	3	7	300	150	3707
letter	18	16	136	68	18668
mfeat-morphological	10	6	300	150	1888
optdigits	10	64	300	150	5620
page-blocks	5	10	300	150	5393
segment	7	19	300	150	2086
synthetic-control	6	60	300	150	600
vehicle	4	18	300	150	846
vowel	11	10	300	150	990
waveform	3	40	300	150	5000

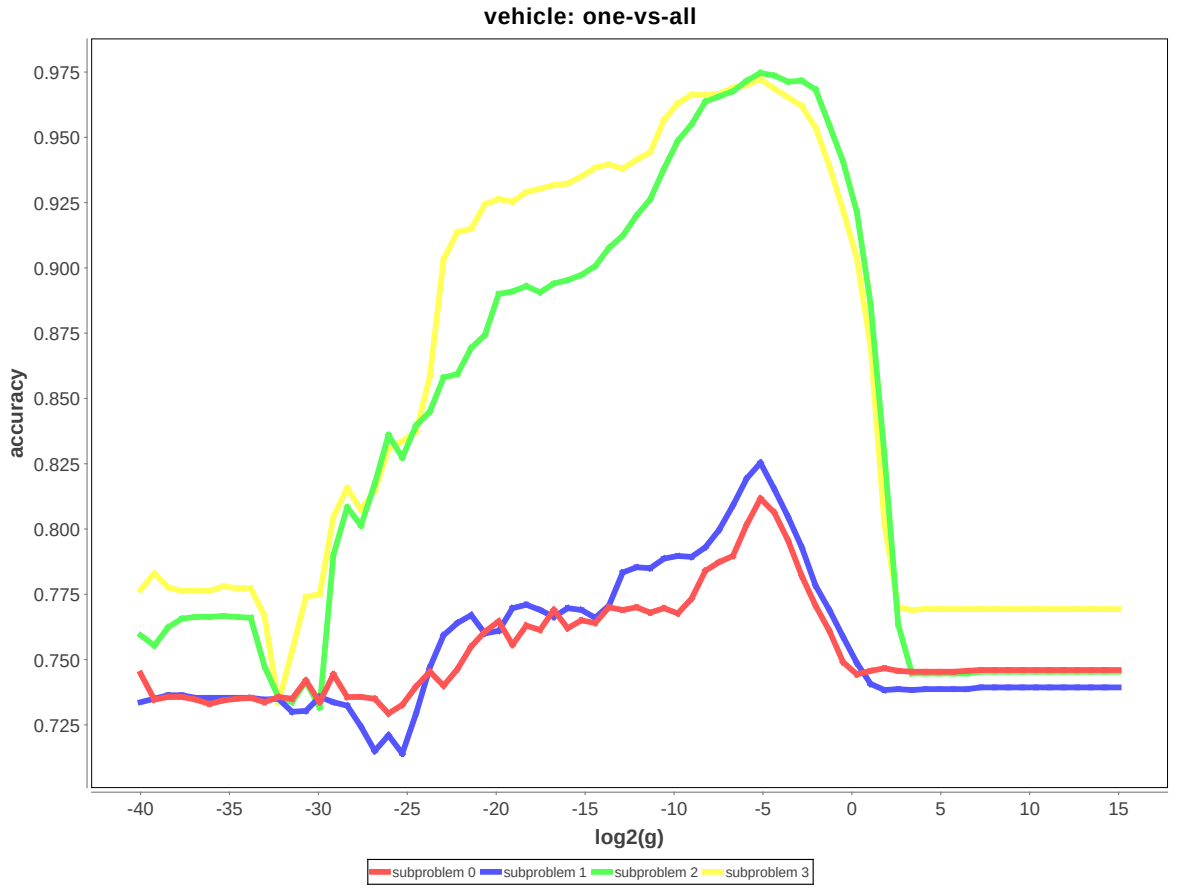
Methods

- Reductions: $\{$one-vs-all, all-pairs$\} \times \{$hamming, squared$\}$
- Model selection: $\{$shared, independent$\}$
- Base classifier: LibSVM with 2-phase grid search

Shared vs Independent: Test Set Accuracy

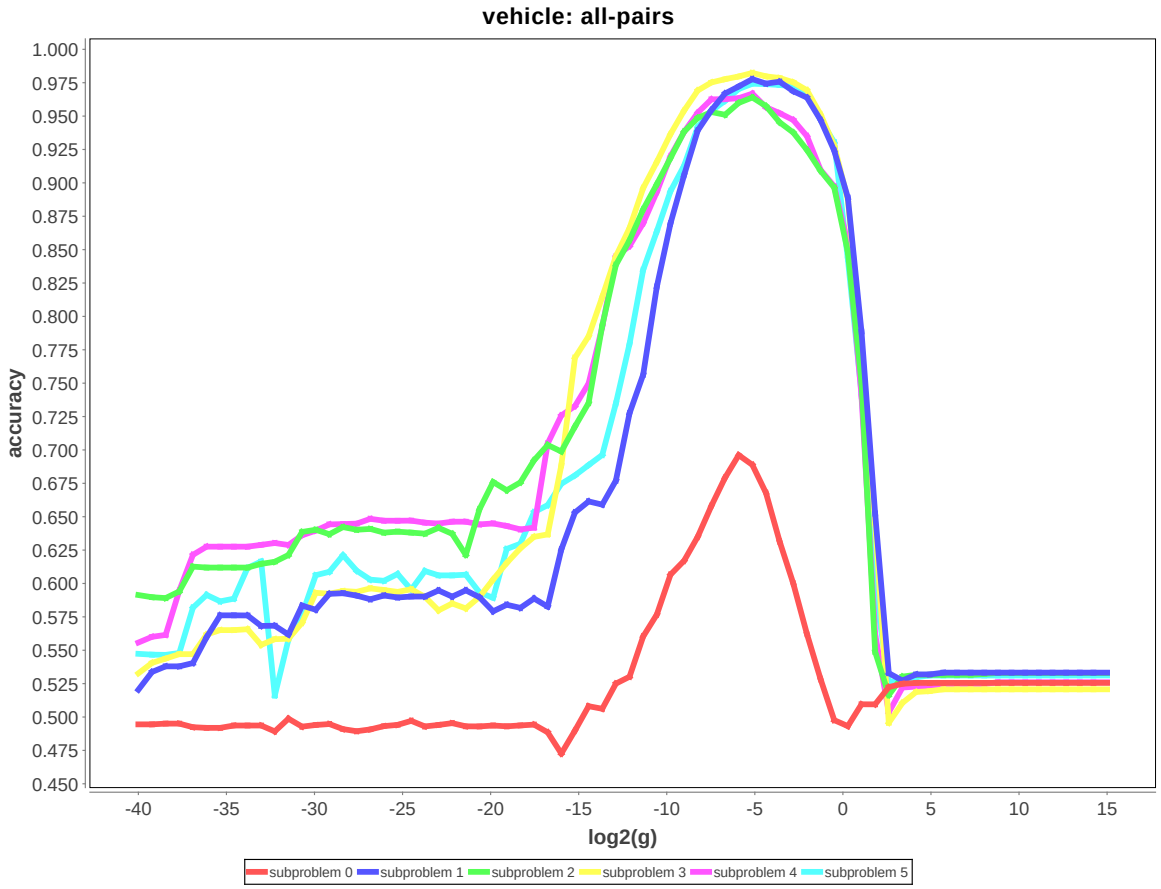


Subproblems are Similar - Vehicle, one-vs-all



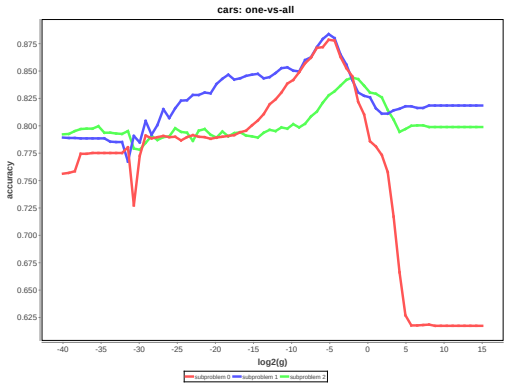
Independent model selection curves for one-vs-all on *vehicle*

Subproblems are Similar - Vehicle, all-pairs

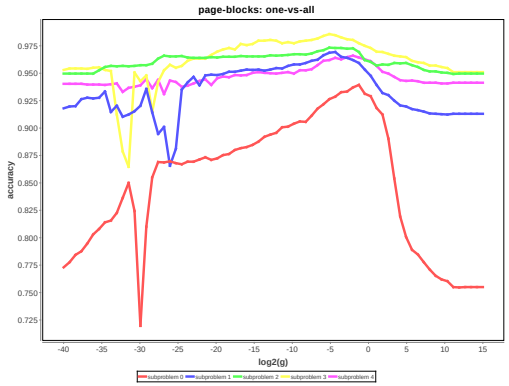


Independent model selection curves for all-pairs on *vehicle*

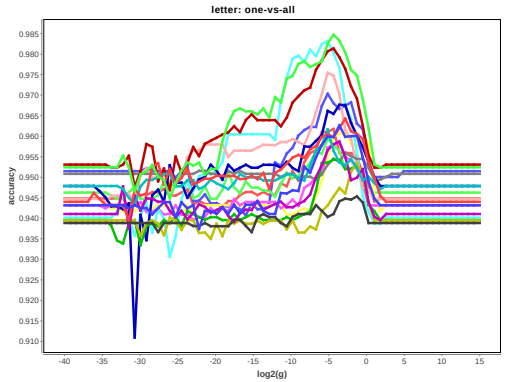
Subproblems are Similar - Examples



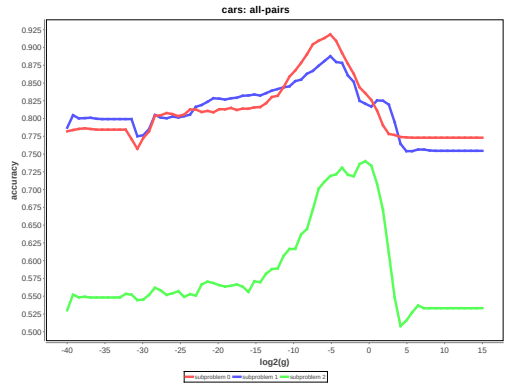
cars: one-vs-all



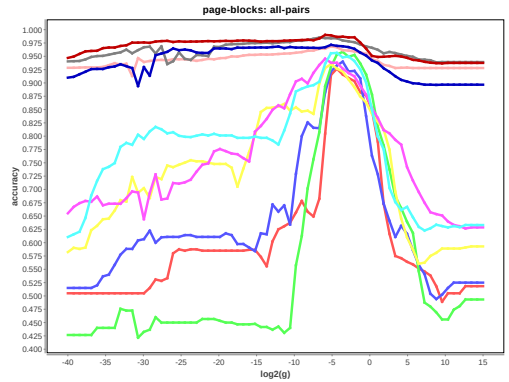
page-blocks: one-vs-all



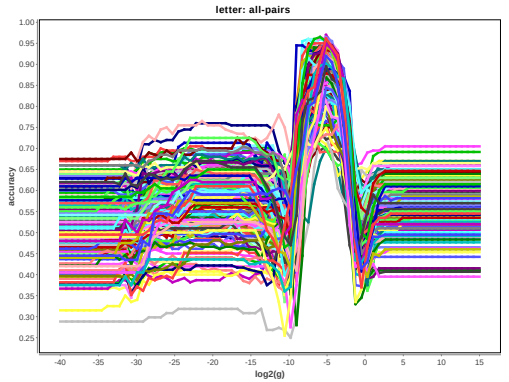
letter: one-vs-all



cars: all-pairs



page-blocks: all-pairs

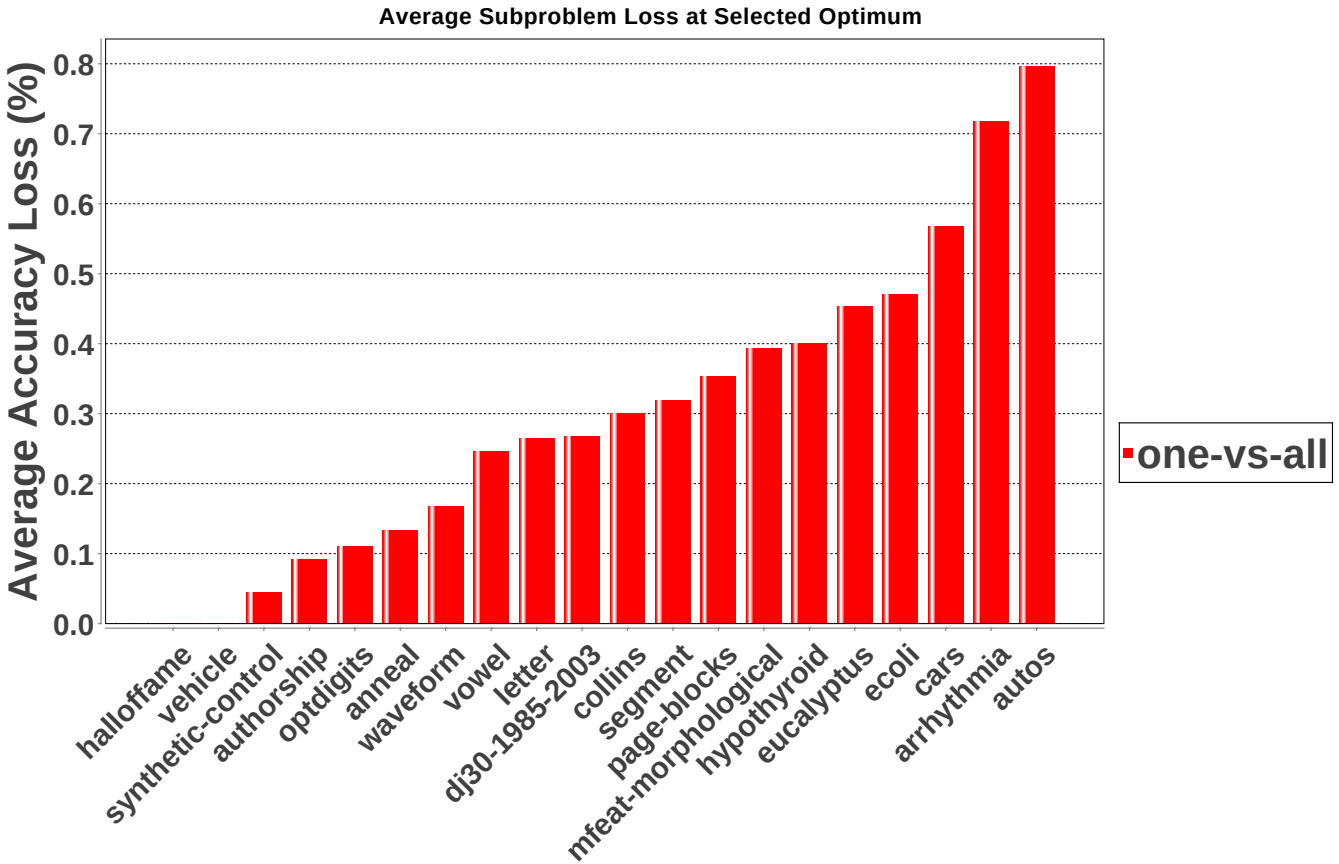


letter: all-pairs

Subproblems are Similar - Aggregate Results 1/3

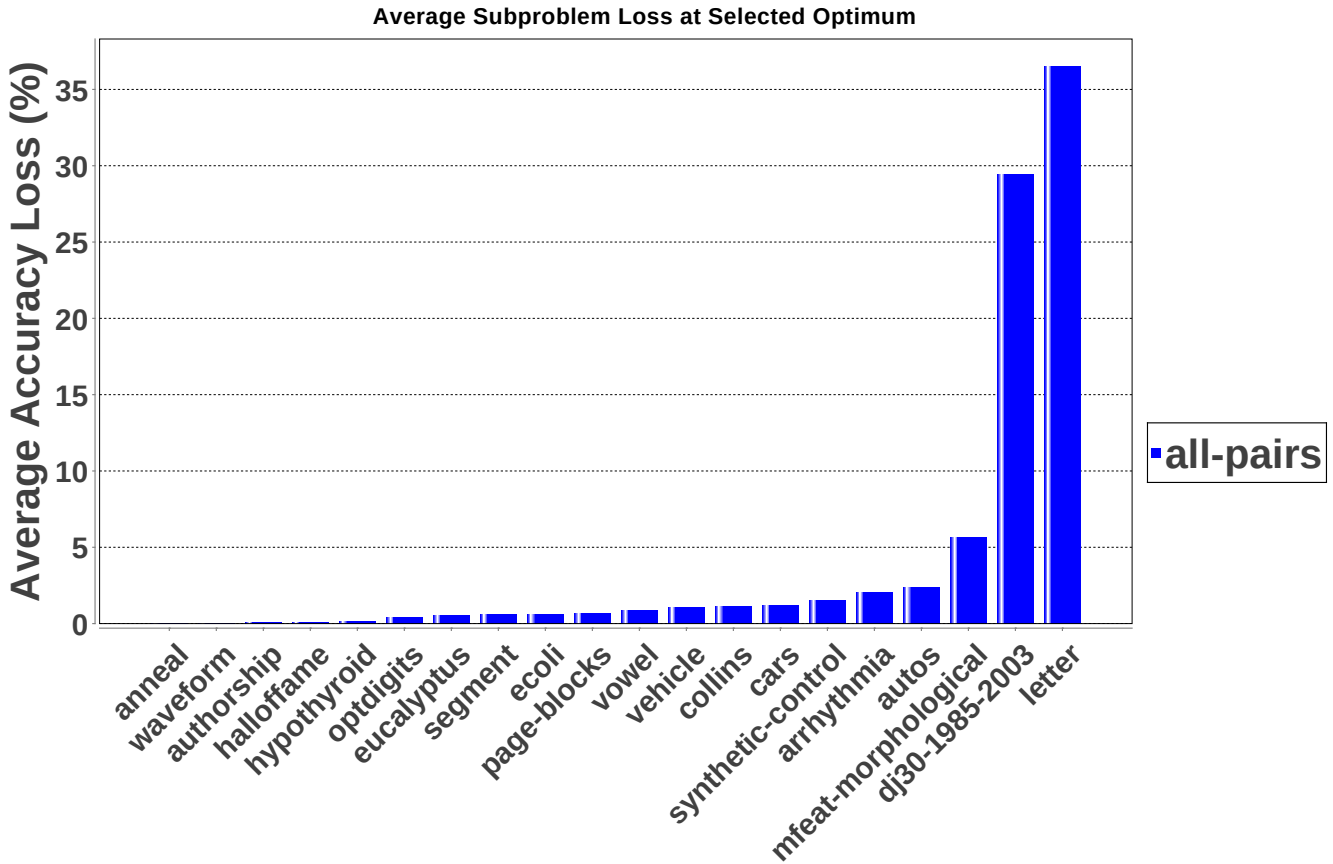
- Define γ_s = optimal shared hyperparameter
- γ_i = optimal independent hyperparameter
- Compute accuracy difference $d = \bar{a}(\gamma_i) - a(\gamma_s)$
  - Where $\bar{a}$ indicates an average over subproblems

Subproblems are Similar - Aggregate Results 2/3



- For each dataset i, $d_i < 0.80\%$
- Average $\bar{d} = 0.30\%$

Subproblems are Similar - Aggregate Results 3/3

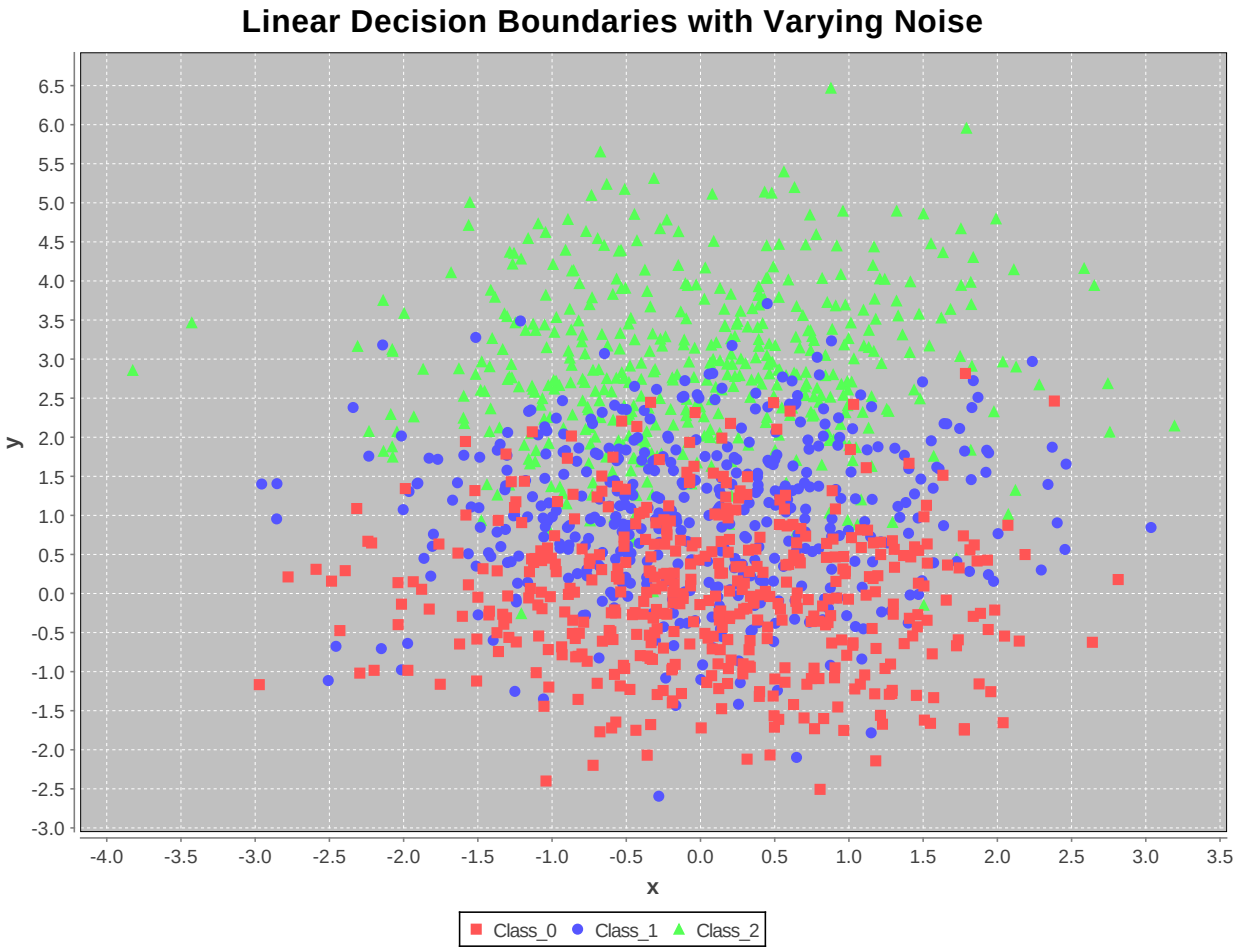


- Largest values: 36.6% (*letter*), 29.4% (*dj30-1985-2003*)
- Average $\bar{d} = 4.24\%$

Differing Subproblems Favor Independent

- Construct a synthetic problem with different shapes of decision boundaries
- Requires different hyperparameters $\Rightarrow$ Requires independent optimization
- First, a control experiment with only linear decision boundaries

Differing Subproblems Favor Independent - Linear Synthetic Data 1/2

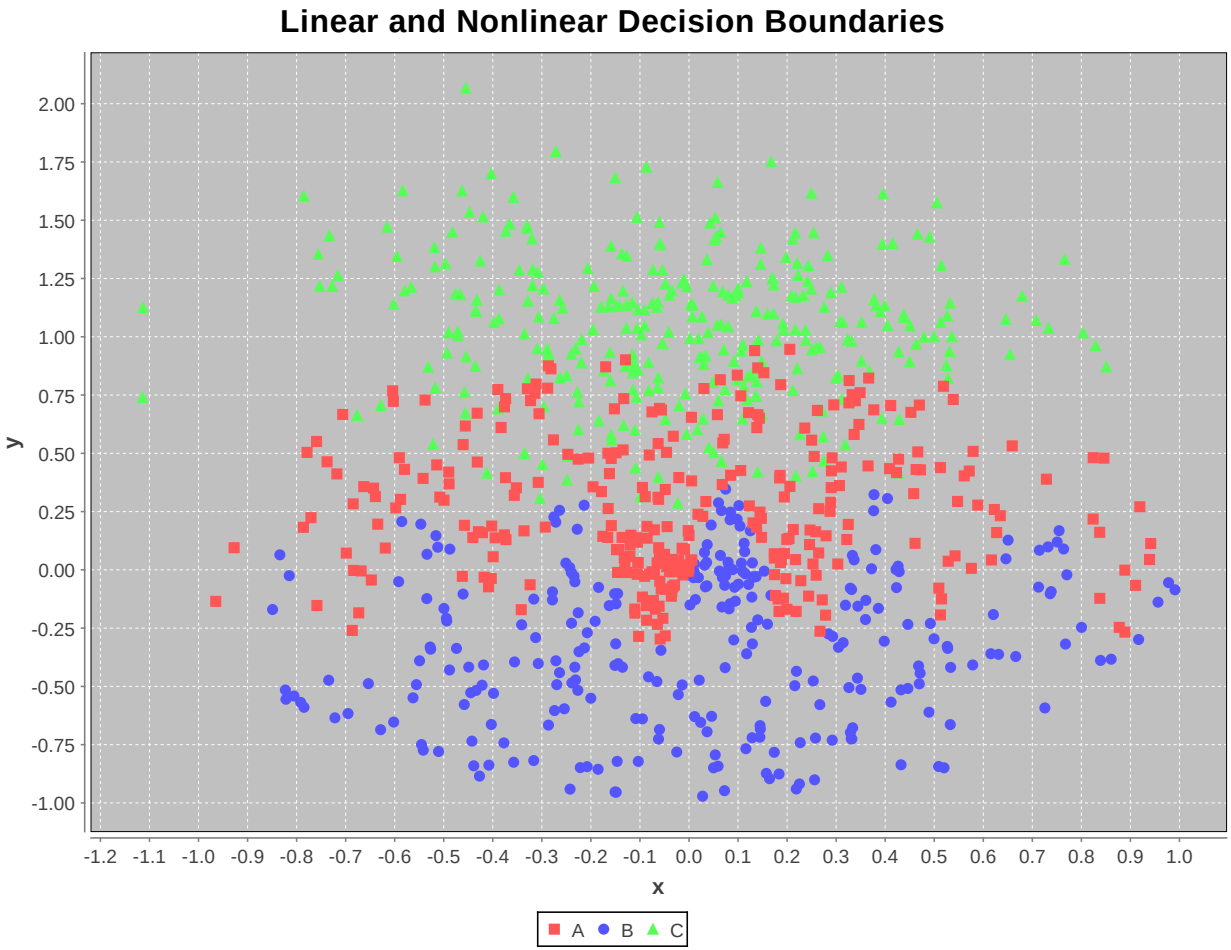


Differing Subproblems Favor Independent - Linear Synthetic Data 2/2

Accuracy (%) results for linear decision boundaries. Standard error over 10 random samplings is indicated in parentheses.

reduction	shared	independent
one-vs-all	**66.7** (1.3)	66.1 (1.3)
one-vs-all-hamming	**58.2** (2.5)	58.1 (1.9)
all-pairs	**67.6** (1.3)	66.5 (1.9)

Differing Subproblems Favor Independent - Mixed Synthetic Data 1/2

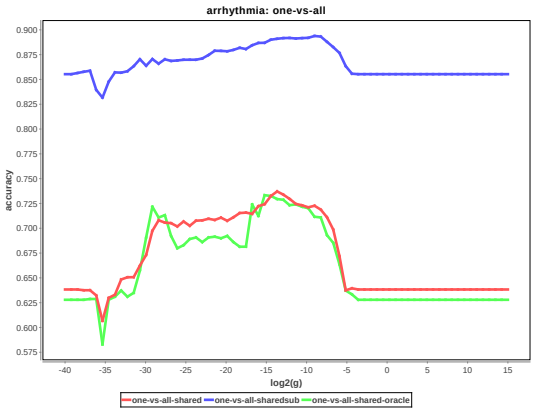


Differing Subproblems Favor Independent - Mixed Synthetic Data 2/2

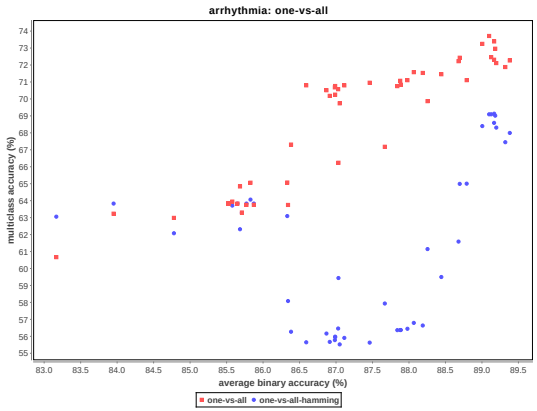
Accuracy (%) results for mixed linear and nonlinear decision boundaries. Standard error over 10 random samplings is indicated in parentheses.

reduction	shared	independent
one-vs-all	82.4 (0.6)	**83.5** (0.9)
one-vs-all-hamming	78.5 (1.3)	**79.5** (1.3)
all-pairs	82.4 (1.3)	**84.2** (0.9)

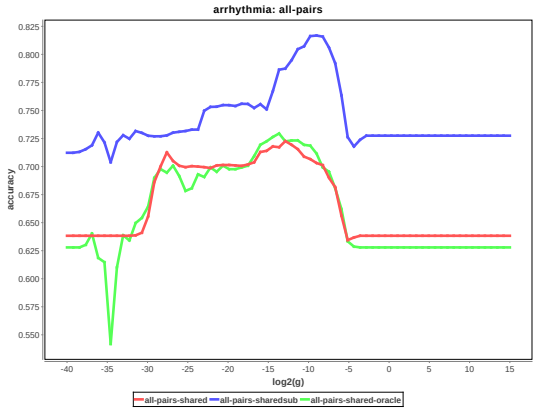
Multiclass Accuracy $\propto$ Binary Accuracy + Noise



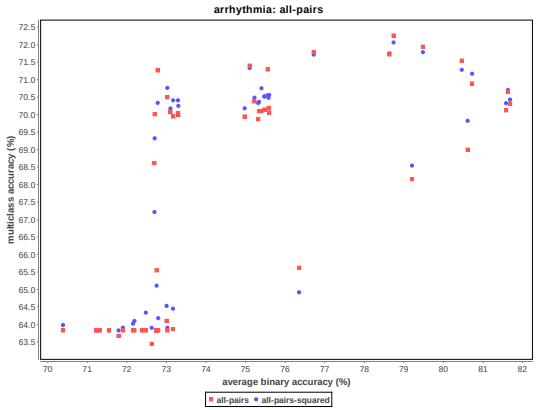
one-vs-all



one-vs-all multi vs binary

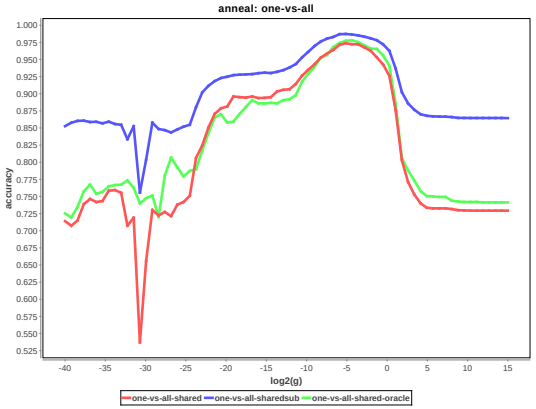


all-pairs

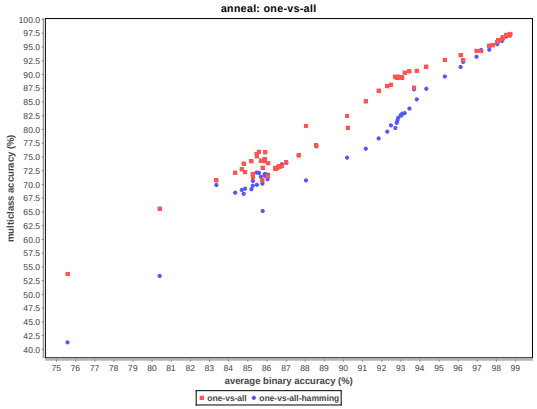


all-pairs multi vs binary

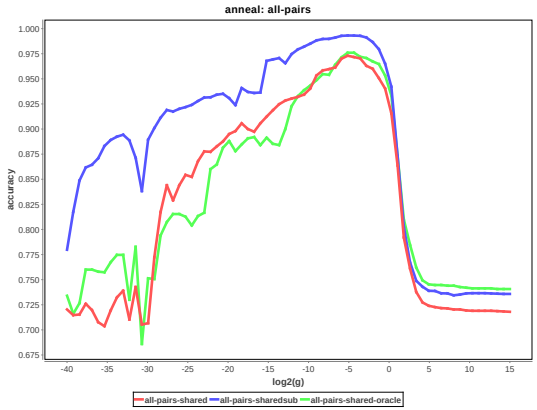
Multiclass Accuracy $\propto$ Binary Accuracy + Noise: Anneal



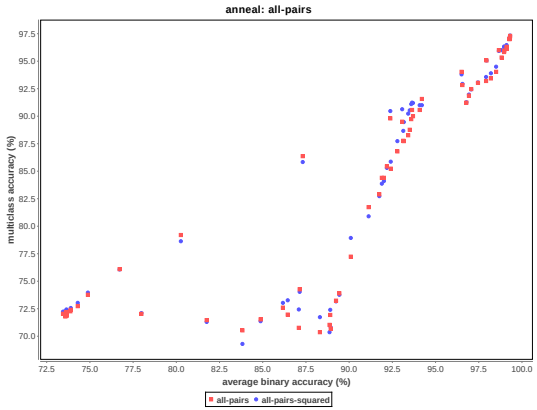
one-vs-all



one-vs-all multi vs binary

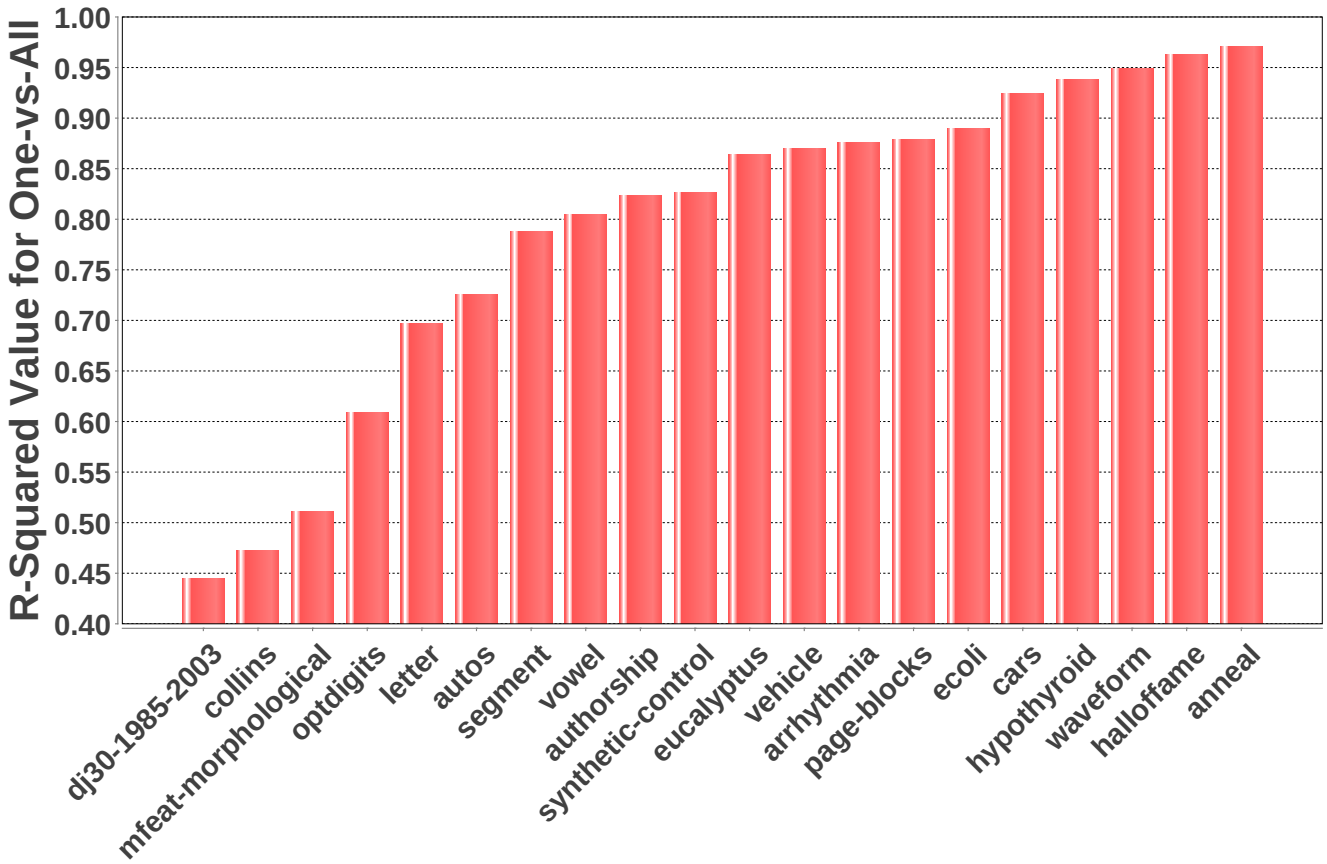


all-pairs



all-pairs multi vs binary

Multiclass Accuracy $\propto$ Binary Accuracy + Noise: Aggregate



- Average R-Squared Value: One-vs-all=0.791, All-pairs=0.910

Multiclass Metric Non-Essential

- Hypothesis: Advantage of shared due to selection on target multiclass metric
- To test, implement *shared-sub*
  - Constraints models to be shared
  - But selected based on average binary accuracy
- Results comparing *shared* vs *shared-sub*
  - one-vs-all: $p \leq 0.65$
  - all-pairs: $p \leq 0.10$
  - ova-hamming: $p \leq 0.57$
- No statistically significant differences
- Conclusion: Sharing hyperparameters valuable whether you use avg binary or multiclass metric

Oracle Selection favors Shared

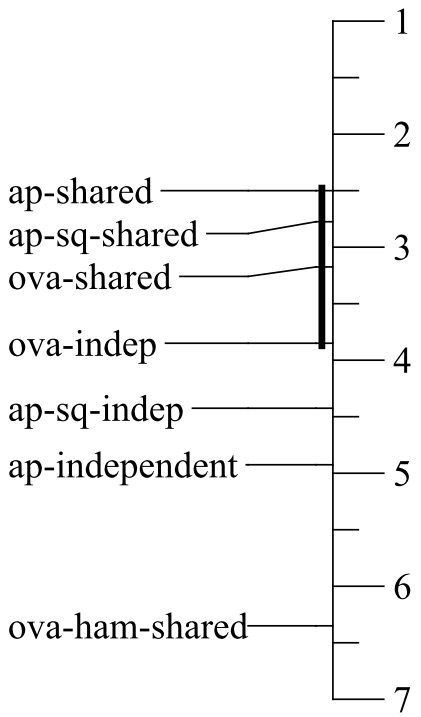
- To rule out sampling problems, use an oracle to select the optimal model
- Use oracle for both shared and independent

	one-vs-all	all-pairs	one-vs-all-hamming	all-pairs-squared
accuracy	**shared** 0.0071	**indep** 5.72×10^{-6}	**indep** 4.77×10^{-5}	indep 0.3955

P-values from the Wilcoxon signed-ranks test are indicated by the winning strategy.

- For one-vs-all, shared still beats independent
- Independent wins for all-pairs and one-vs-all-hamming
- No difference for all-pairs-squared

Supplementary Result: Comparing Methods



Average ranks of the 7 algorithms under study (omitted ova-ham-indep); algorithms not statistically significantly different from the top-scoring algorithm are connected to it with a vertical line.

Conclusions

- Shared hyperparameters often better than independent optimization
- Subproblems often similar, especially in one-vs-all
- If there are different decision boundary shapes, use independent
- Future Work
  - Multiclass metrics with no binary analog in independent optimization? (e.g. multiclass cost matrix)
  - Relationship to regret transform, Langford & Beygelzimer, 2005

Outline

Pairwise Classification

- Assuming a classification problem with $k \geq 3$ classes
- $k(k-1)/2$ subproblems, one for each pair of classes
- Estimate $\hat{\mu}_{ij}(\vec{x}) \approx \mu_{ij}(\vec{x}) = P(y = c_i | y = c_i \text{ or } c_j, \vec{x})$
- Note that $\mu_{ij} = \frac{p_i}{p_i + p_j}$
- Combine: $\mathbf{p} = \{p_1, p_2, ..., p_k\} = f(\hat{\mu}_{ij}(\vec{x}))$

Pairwise Classification Subproblem Example

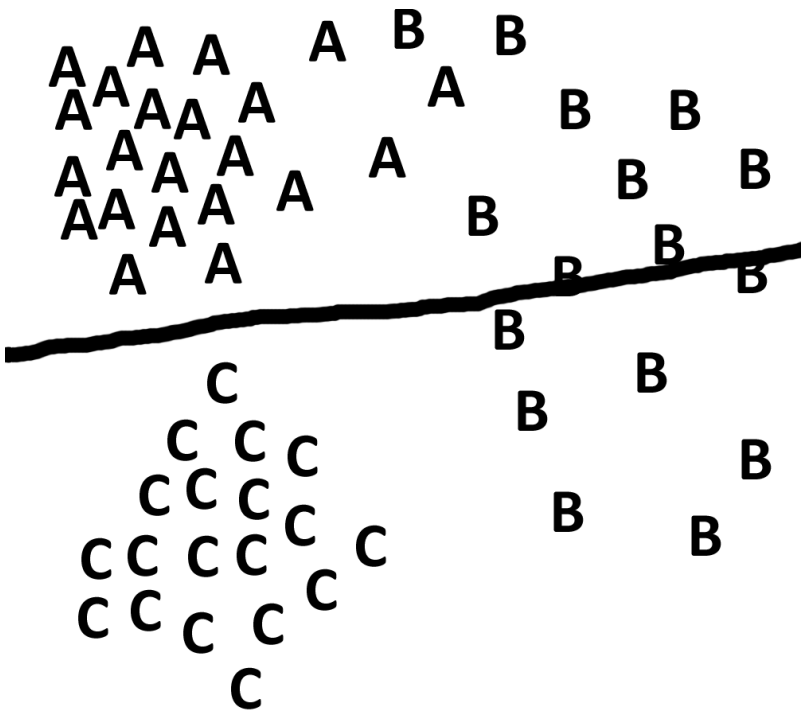


Illustration of an A-C decision boundary in a 2D, 3-class example of pairwise classification.

Pairwise Classification Methods

- Voted pairwise classification (VPC): Friedman, 1996
  - $\hat{y}(\vec{x}) = \text{argmax}_i \sum_{j:j \neq i} 1(\hat{\mu}_{ij}(\vec{x}) > \hat{\mu}_{ji}(\vec{x}))$
  - Equivalent to Bayes optimal prediction if $\hat{\mu}_{ij}(\vec{x}) = \mu_{ij}(\vec{x})$

Pairwise Classification Methods

- Voted pairwise classification (VPC): Friedman, 1996
  - $\hat{y}(\vec{x}) = \text{argmax}_i \sum_{j:j \neq i} 1(\hat{\mu}_{ij}(\vec{x}) > \hat{\mu}_{ji}(\vec{x}))$
  - Equivalent to Bayes optimal prediction if $\hat{\mu}_{ij}(\vec{x}) = \mu_{ij}(\vec{x})$
- Hastie & Tibshirani (HT), 1996
  - Iteratively update $\mathbf{p} = \{p_1, p_2, ..., p_k\}$
  - Min KL-Divergence between μ and $\hat{\mu}$, $l(\mathbf{p}) = \sum_{i \neq j} n_{ij} \hat{\mu}_{ij} \frac{\hat{\mu}_{ij}}{\mu_{ij}}$
  - Converges to minimum of KL divergence

Pairwise Classification Methods

- Voted pairwise classification (VPC): Friedman, 1996
  - $\hat{y}(\vec{x}) = \text{argmax}_i \sum_{j:j\neq i} 1(\hat{\mu}_{ij}(\vec{x}) > \hat{\mu}_{ji}(\vec{x}))$
  - Equivalent to Bayes optimal prediction if $\hat{\mu}_{ij}(\vec{x}) = \mu_{ij}(\vec{x})$
- Hastie & Tibshirani (HT), 1996
  - Iteratively update $\mathbf{p} = \{p_1, p_2, ..., p_k\}$
  - Min KL-Divergence between μ and $\hat{\mu}$, $l(\mathbf{p}) = \sum_{i\neq j} n_{ij}\hat{\mu}_{ij}\frac{\hat{\mu}_{ij}}{\mu_{ij}}$
  - Converges to minimum of KL divergence
- Wu, Lin, Weng (WLW), 2004
  - $\mu_{ij} = \frac{p_i}{p_i+p_j} \Rightarrow \frac{\mu_{ij}}{\mu_{ji}} = \frac{p_i}{p_j}$
  - Approx $\min_{\mathbf{p}} \sum_{i=1}^{k} \sum_{j\neq i} (\hat{\mu}_{ji}p_i - \hat{\mu}_{ij}p_j)^2$ s.t. $\sum_{i=1}^{k} p_i = 1, p_i \geq 0$
  - Guaranteed convergence

Pairwise Classification

- Pros (Furnkranz, 2002)
  - Smaller Subproblems
  - Simpler Subproblems
  - Improved Accuracy (disputed by Rifkin & Klautau, 2004)
- Cons
  - Larger number of subproblems than one-vs-all
  - Each pairwise classifier is trained on only two of the classes but makes predictions for instances from any class (Hastie & Tibshirani, 1996, Cutzu, 2003)
    e.g. a classifier trained on c_A and c_B may have unpredictable behavior for instances with $y(\vec{x}) = c_C$

Thesis Statement - Part III

- When solving a multiclass problem with a set of pairwise binary classifiers, incorporation of the probability of membership in each pair improves performance.

Probabilistic Pairwise Classification: Derivation 1/2

Theorem of Total Probability:

$$p(b|\vec{x}) = \sum_{i=1}^{N} p(b|a_i, \vec{x}) p(a_i|\vec{x}) \quad (1)$$

Assumes

- $a_1..a_N$ mutually exclusive and exhaustive so $\sum_{i=1}^{N} p(a_i|\vec{x}) = 1$

Let

- $b = c_i$
- $N = 2$
- $a_1 = c_i \cup c_j$
- $a_2 = L - c_i - c_j$, for $L = \{c_1..c_k\}$

$$\begin{aligned} p(c_i|L, \vec{x}) &= p(c_i|c_i \cup c_j, \vec{x}) p(c_i \cup c_j|L, \vec{x}) \\ &\quad + p(c_i|L - c_i - c_j, \vec{x}) p(L - c_i - c_j|L, \vec{x}) \end{aligned}$$

Probabilistic Pairwise Classification: Derivation 2/2

- But

$$p(c_i|L - c_i - c_j, \vec{x}) = 0 \tag{2}$$

-

$$\Rightarrow p(c_i|\vec{x}) = p(c_i|c_i \cup c_j, \vec{x})p(c_i \cup c_j|L, \vec{x}) \tag{3}$$

- Average over all $j \neq i$

$$\hat{p}(c_i|L, \vec{x}) = \frac{1}{k-1} \sum_{j \neq i} \hat{p}(c_i|c_i \cup c_j, \vec{x})\hat{p}(c_i \cup c_j|L, \vec{x}) \tag{4}$$

- Normalize so that $\sum_i \hat{p}(c_i|L, \vec{x}) = 1$.

Comparison to Other Pairwise Classification Methods

- PPC
  - Solves for each term $p_i(\vec{x})$ independently
  - Models $p_i + p_j = p(i \text{ or } j | L, \vec{x})$ directly
  - Conceptually simpler
  - Easier to implement
  - Theoretically well motivated
- Hastie-Tibshirani (HT) method approximates $p_i = \sum_{j \neq i} (\frac{2}{k(k-1)}) \mu_{ij}$ (Wu et al., 2004)
  - Equivalent to our method with the assumption $p_i + p_j = 2/k$

Computational Complexity

Computational complexity of one-vs-all (OVA), pairwise coupling (PC) and probabilistic pairwise classification (PPC)

	OVA	PC	PPC
subproblems	k	k(k-1)/2	k(k-1)
instances per subproblem	N	$2N/k$	N (half) + $2N/k$ (other half)
computational complexity/SVM	$O(kN^3)$	$O(k^{-1}N^3)$	$O(k^2N^3)$

Experiments

- Base Classifiers
  - Decision Tree (J48)
  - K-Nearest Neighbor (KNN)
  - Random Forests (RF-100)
  - Support Vector Machines (SVM-121)

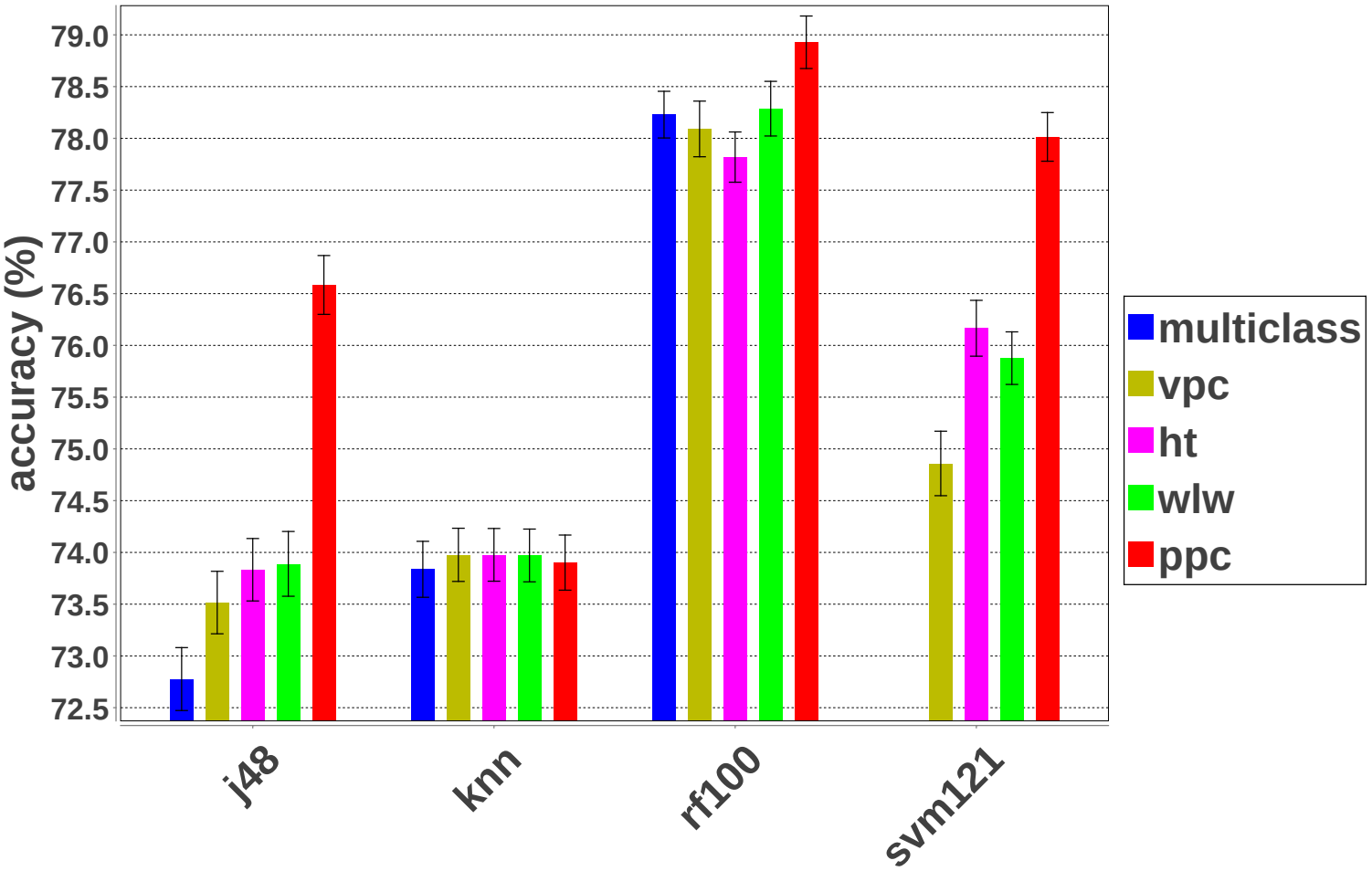
Experiments

- Base Classifiers
  - Decision Tree (J48)
  - K-Nearest Neighbor (KNN)
  - Random Forests (RF-100)
  - Support Vector Machines (SVM-121)
- Multiclass Classification Methods
  - Multi (for J48, KNN, RF-100)
  - Voted Pairwise Classification (VPC)
  - Hastie-Tibshirani (HT)
  - Wu, Lin, Weng (WLW)
  - Probabilistic Pairwise Classification (PPC)

Experiments

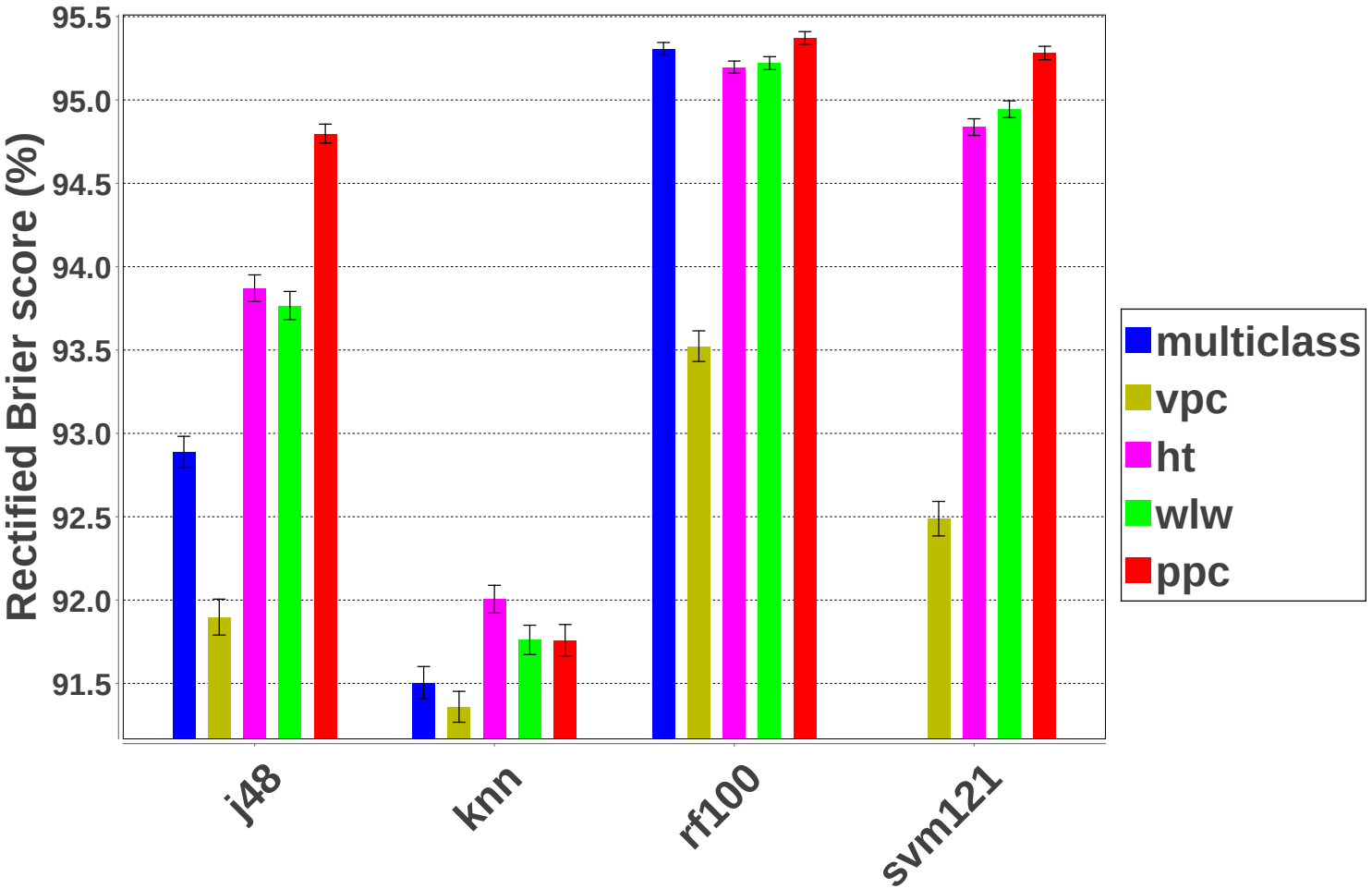
- Base Classifiers
  - Decision Tree (J48)
  - K-Nearest Neighbor (KNN)
  - Random Forests (RF-100)
  - Support Vector Machines (SVM-121)
- Multiclass Classification Methods
  - Multi (for J48, KNN, RF-100)
  - Voted Pairwise Classification (VPC)
  - Hastie-Tibshirani (HT)
  - Wu, Lin, Weng (WLW)
  - Probabilistic Pairwise Classification (PPC)
- Metrics
  - Accuracy
  - Brier $1 - b(\vec{x}) = 1 - \frac{1}{d}\sum_j (t_j(\vec{x}) - \hat{p}_j(\vec{x}))^2$, $t_j(\vec{x}) = 1(y(\vec{x}) = c_j)$

Average Accuracy



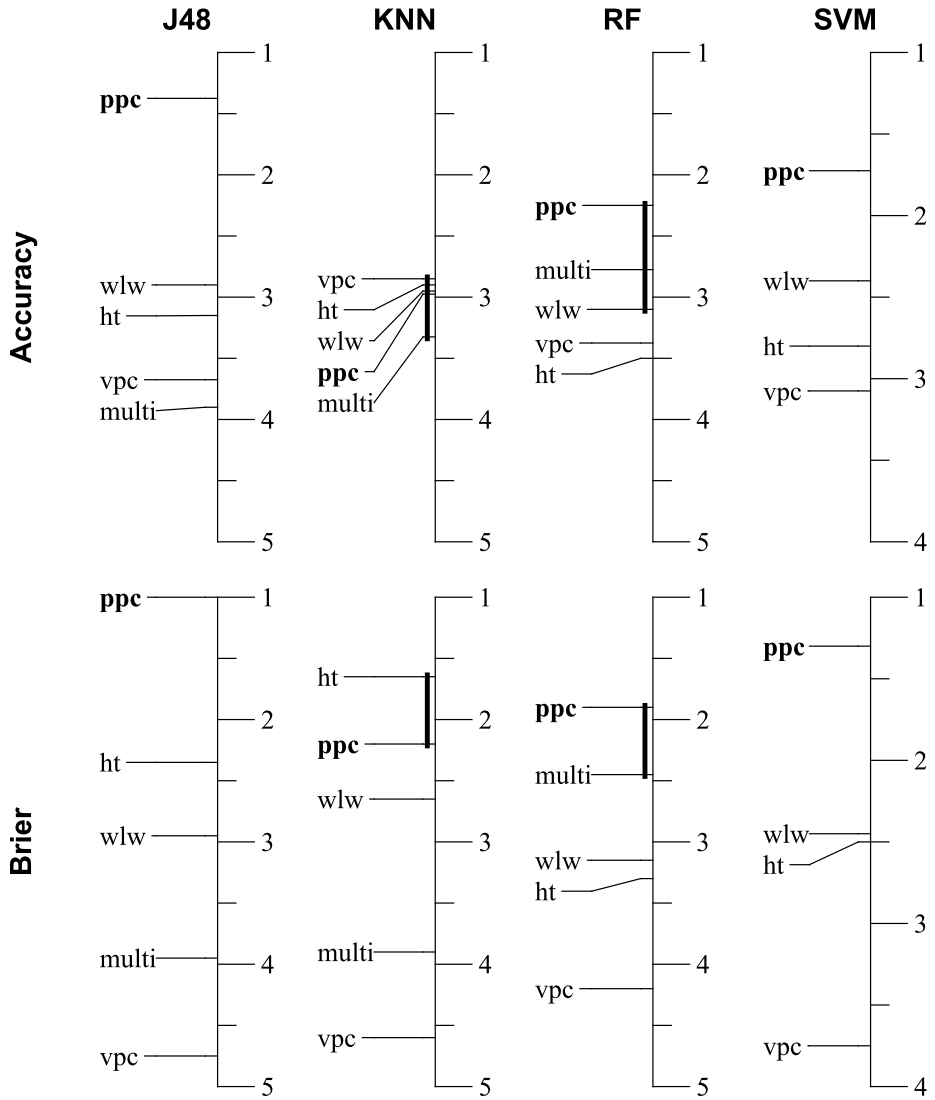
Accuracy averaged over all 20 data sets.

Average Brier Score

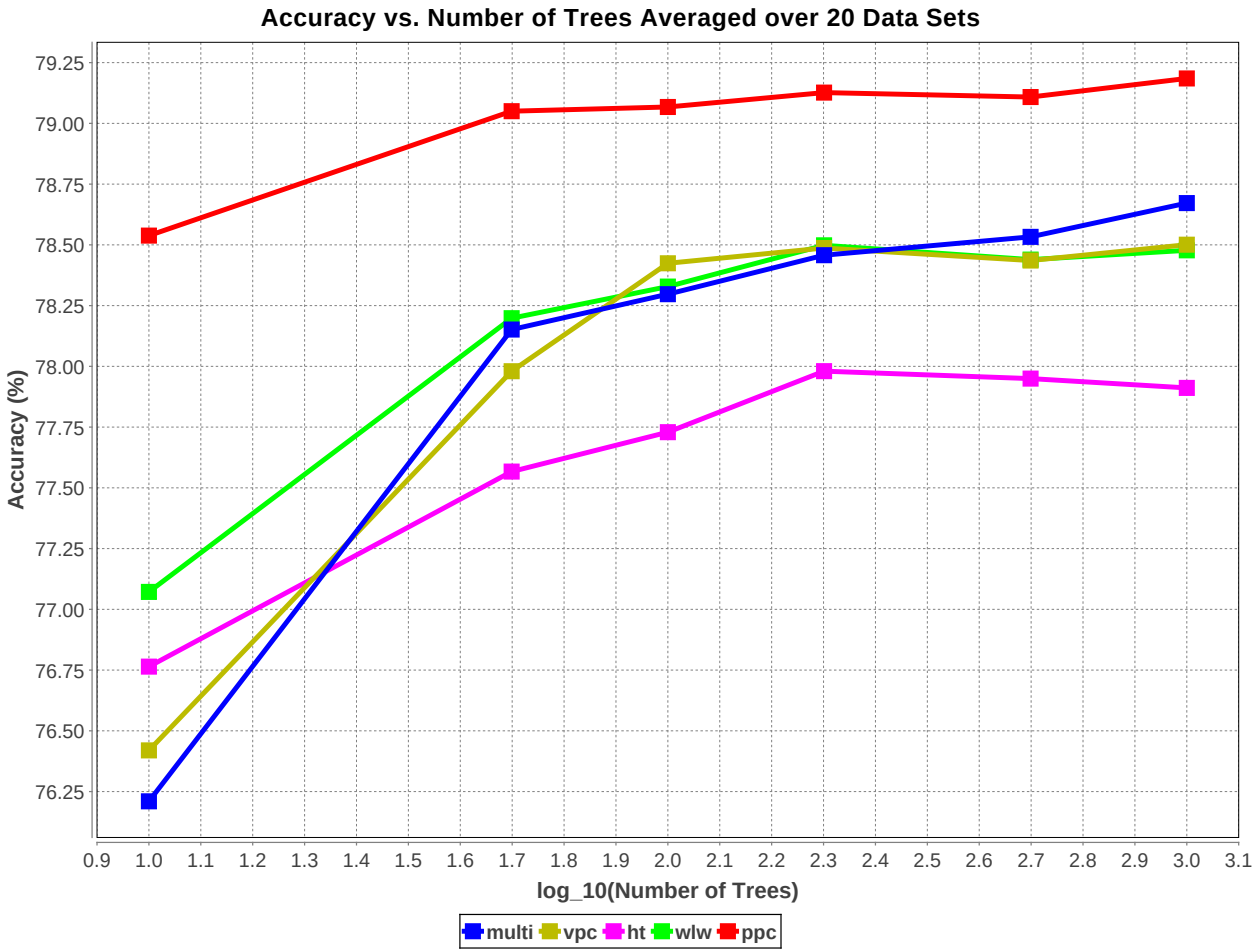


Rectified Brier score averaged over all 20 data sets.

Average Ranks

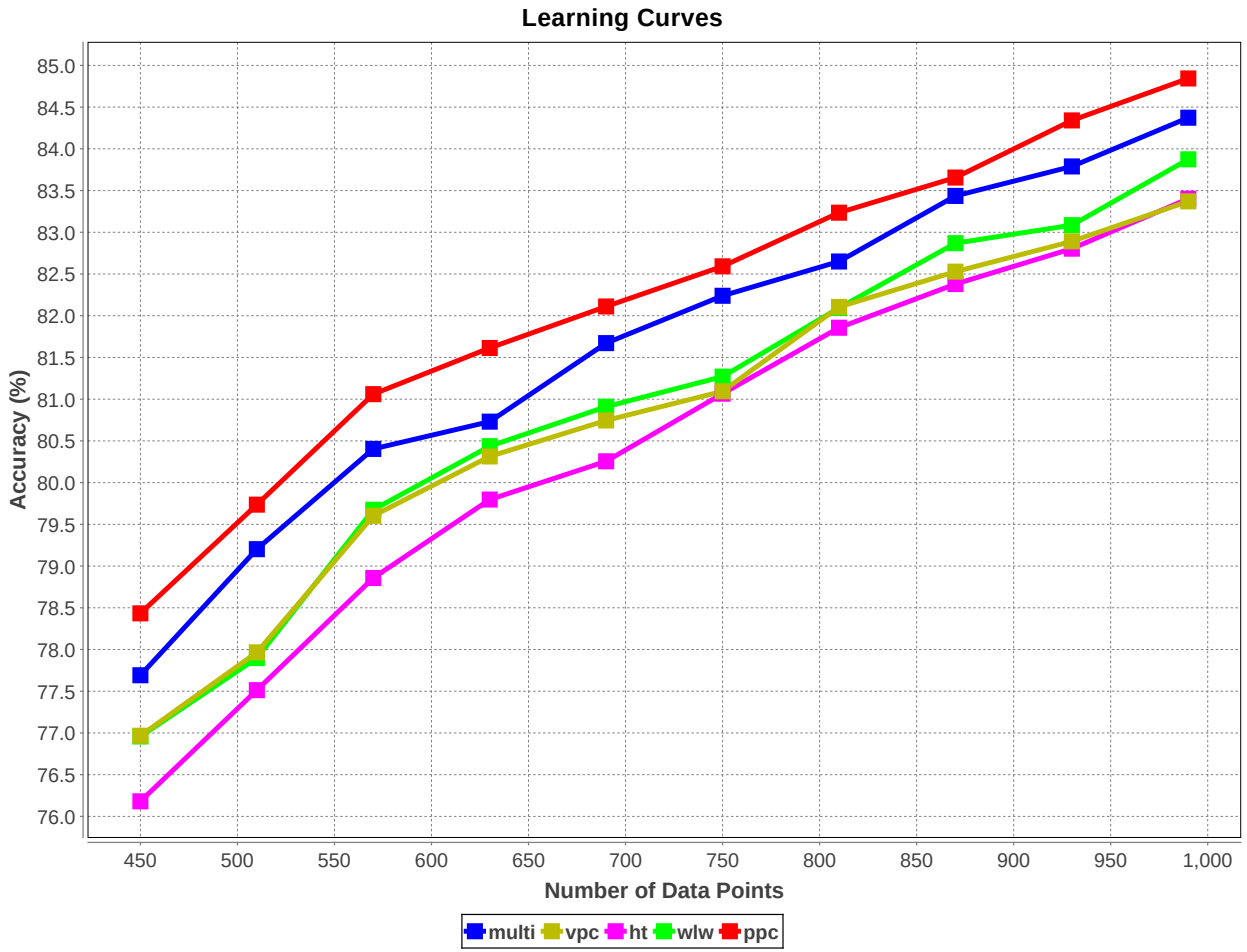


Varying Base Classifier Accuracy



Accuracy vs number of trees in random forest

Learning Curves

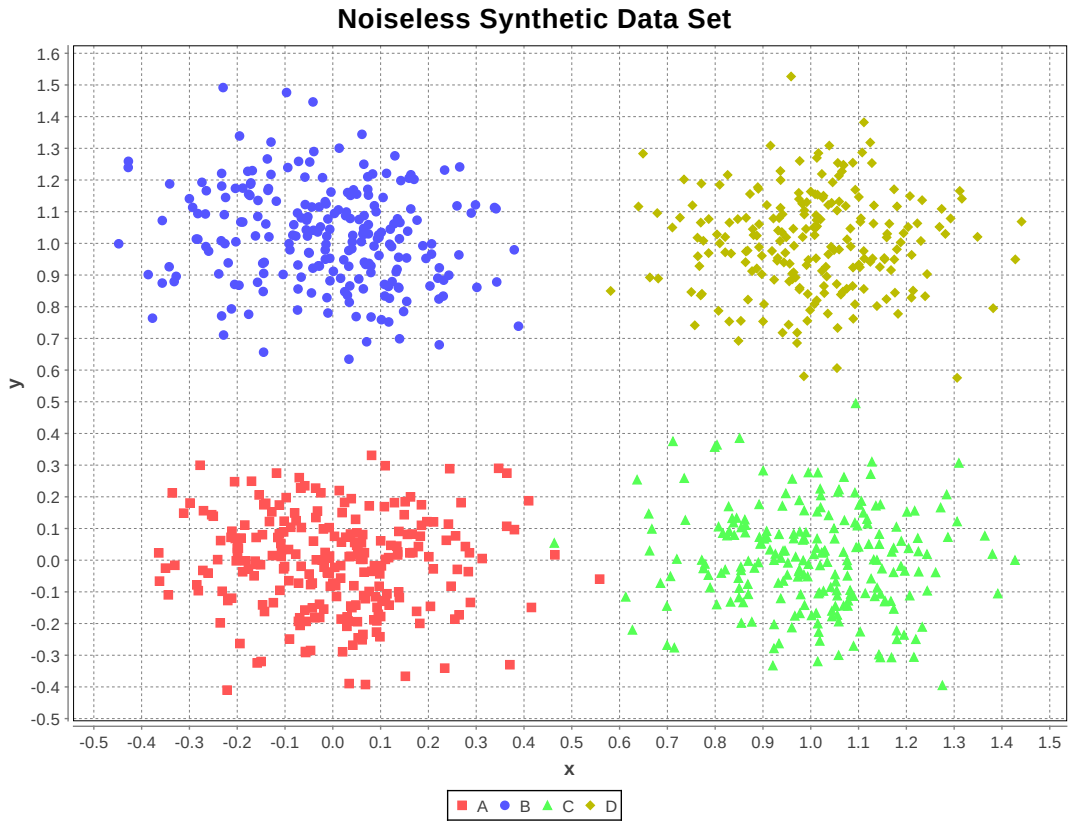


Accuracy vs sample size for 10 largest data sets

Duplicate Decision Boundaries Favors MULTI

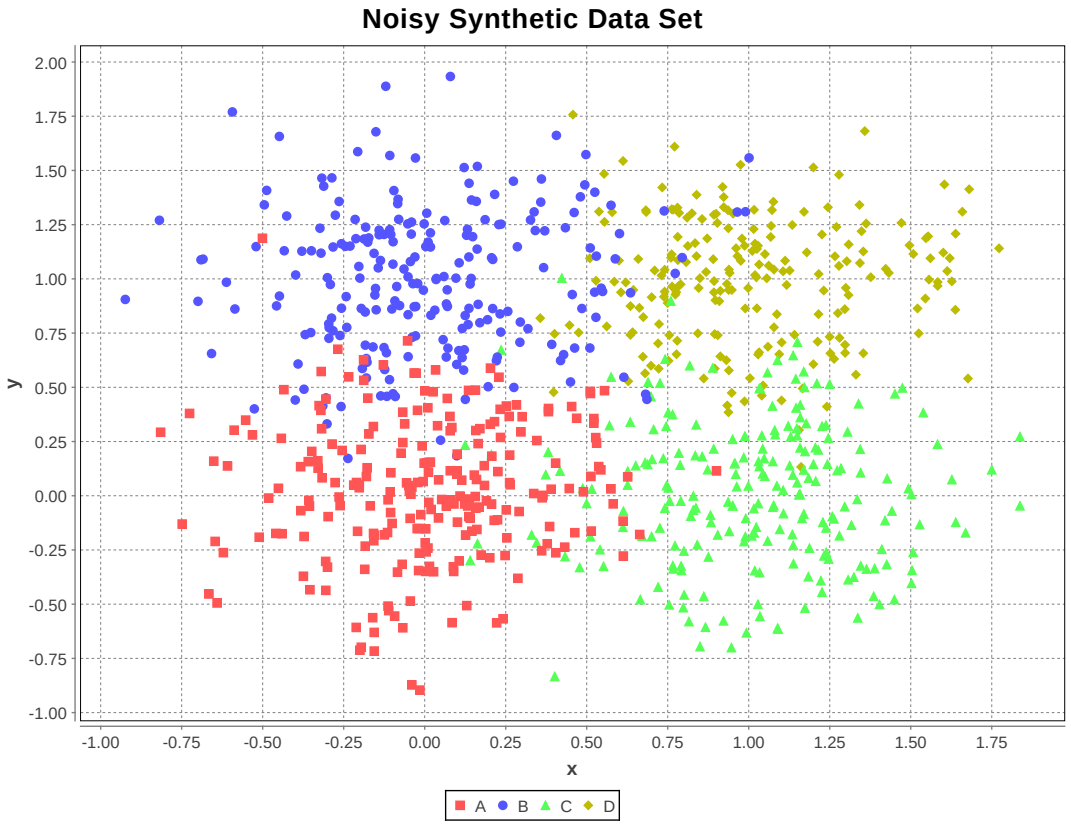
- Hypothesis: Direct multiclass method will outperform PPC when decision boundaries are shared
- Construct a synthetic data set meant to favor multi-j48
- Decision boundaries are shared

Duplicate Decision Boundaries: Noiseless Synthetic Data



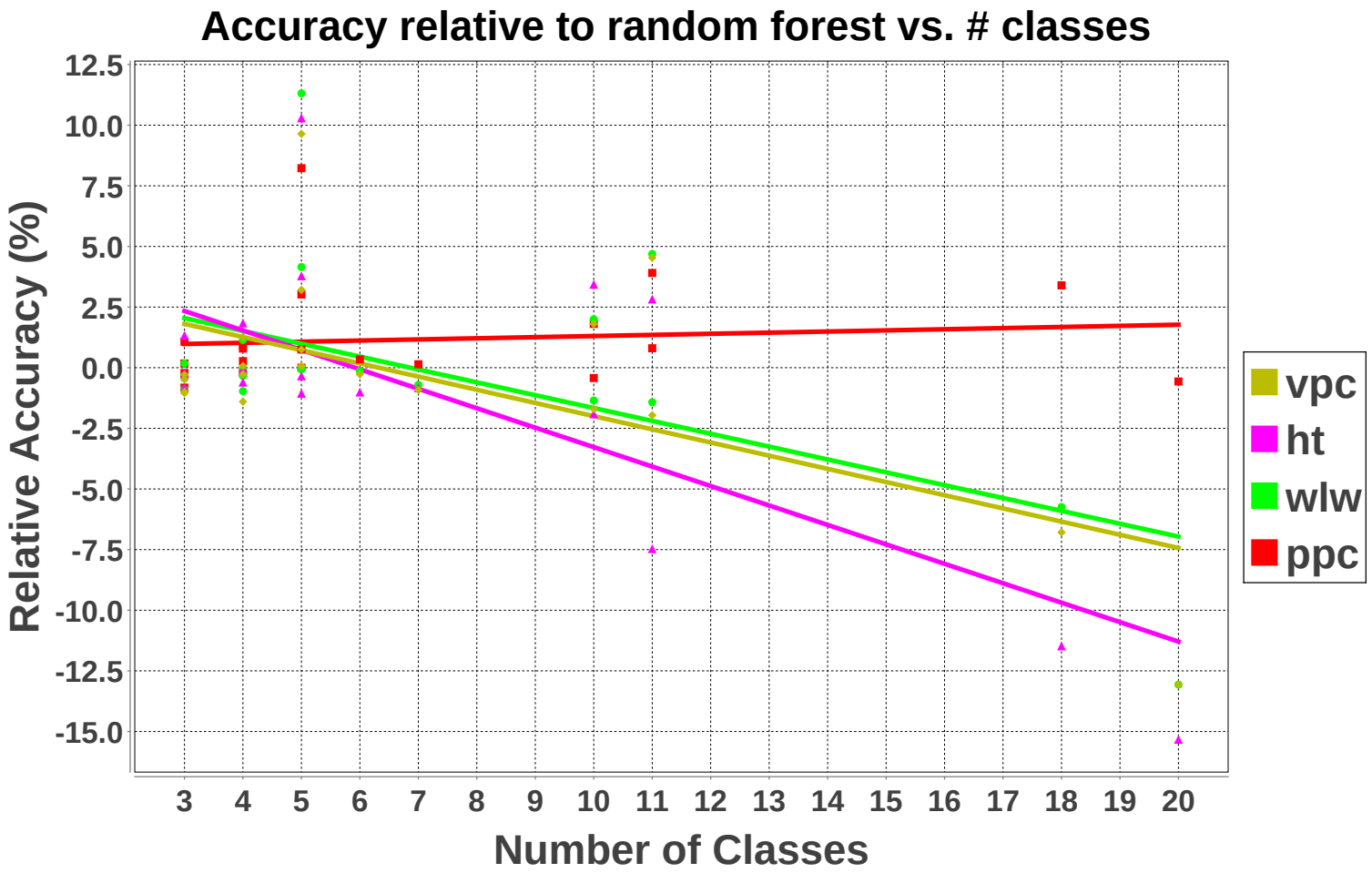
multi-j48	ppc-j48
99.2 (0.08)	98.7 (0.10)

Duplicate Decision Boundaries: Noisy Synthetic Data



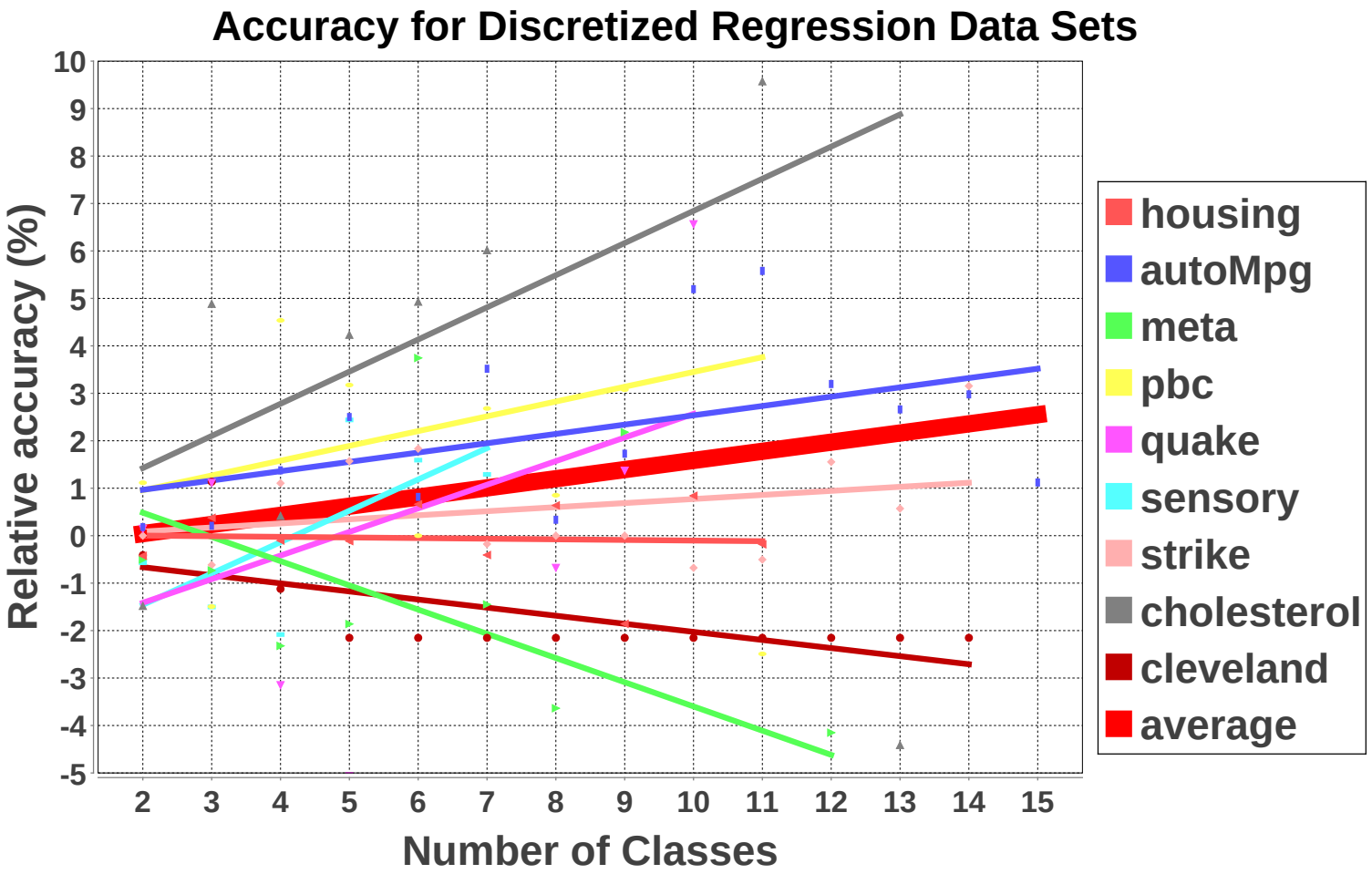
multi-j48	ppc-j48
84.5 (0.34)	**86.0** (0.31)

PPC More Accurate at Large Number of Classes 1/2



Method accuracy relative to RF-100

PPC More Accurate at Large Number of Classes 2/2



PPC relative to RF-100 for discretized regression data sets

Terms in PPC estimate equally important

- Hypothesis: Both terms in the PPC estimate are equally important

$$\hat{p}(c_i|L,\vec{x}) = \frac{1}{k-1}\sum_{j\neq i}\hat{p}(c_i|c_i \cup c_j,\vec{x})\hat{p}(c_i \cup c_j|L,\vec{x})$$

- Pairwise term: $\hat{p}(c_i|c_i \cup c_j,\vec{x})$
- Weight (pair-vs-rest) term: $\hat{p}(c_i \cup c_j|L,\vec{x})$
- Use J48 decision trees, 100 replications, 20 data sets.

Adjusted p-values under various degradations.

hypothesis	p_{Holm}
both vs. none	2.25E-10
no-pair vs. none	6.87E-05
no-weight vs. none	7.49E-04
both vs. no-weight	0.012
both vs. no-pair	0.04693
no-weight vs. no-pair	0.540291

PPC Summary & Conclusions

- Introduced new pairwise classification algorithm, PPC
- Based on Theorem of Total Probability
- Explicitly models $p(c_i \cup c_j | L, \vec{x})$
- Outperforms or ties related methods
  - For several base classifiers, metrics, data sets
- Some data sets benefit from direct multiclass methods
- PPC works well at large # classes
- Future Work
  - Faster but less accurate pair-vs-rest classifier?
  - Independent vs. shared in PPC?

Thesis Statement

Multiclass classification problems can be productively solved by combining multiple classifiers. Specifically:

- In linear combinations of multiclass classifiers, regularization significantly improves performance.
- When solving a multiclass problem with a set of binary classifiers, it is more effective to constrain subproblems to use the same hyperparameters than to optimize each independently.
- When solving a multiclass problem with a set of pairwise binary classifiers, incorporation of the probability of membership in each pair improves performance.

Acknowledgments

- PhET Interactive Simulations
- NSF Grants
  - SBE-0542013
    - Science of Learning Center (Garrison Cottrell, PI)
  - BCS-0339103
  - BCS-720375
  - SBE-0518699
- Mike Mozer, Greg Grudic
- Dissertation Support Group/CAPS
- Turing Institute
- UCI Repository

Questions?

- Questions?