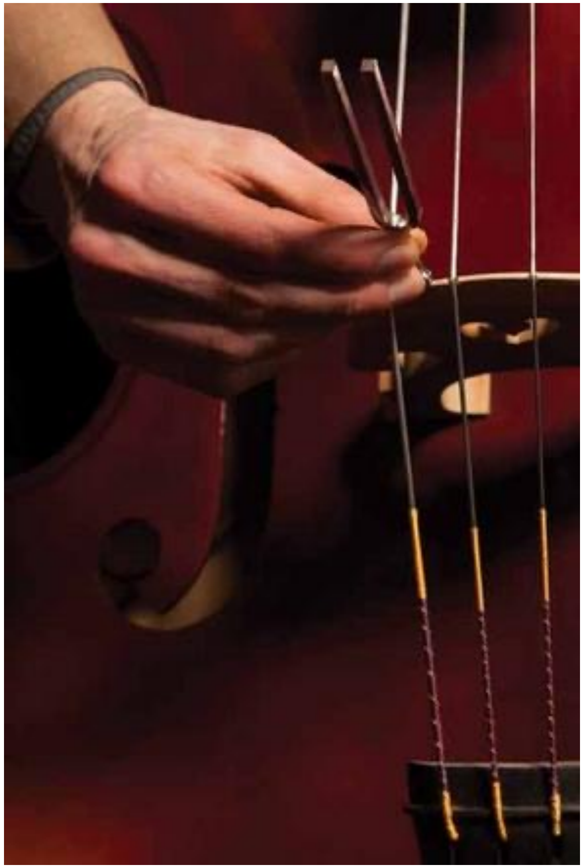
Collaborative Hyperparameter Tuning



Summer Internship Project
Cloud Machine Learning Group
Microsoft, Redmond, WA
June 2014 – Oct 2014

Yamuna Krishnamurthy
Technische Universität Dortmund
Germany

Mentors

Misha Bilenko
Rich Caruana
Ofer Dekel
Yael Dekel

Hyperparameter Tuning

- State of the Art
- **Collaborative Hyperparameter Tuning** – Our Approach
  - Featurize Datasets
  - Extensive Experiments on Different Datasets
  - Create historical knowledgebase of results
  - Generate smart sweeps on demand
- Performance Results

State of the Art

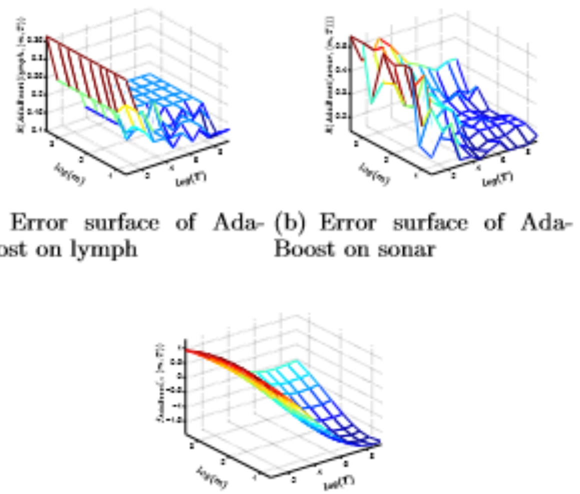
- ParamILS, local-search based methods, *Hutter et al* [1]
- REVAC, estimation of distribution methods, *Nannen and Eiben* [2]
- Spearmint, *Snoek et al* [3]
- Surrogate optimization approach in Weka platform *Thorton et al* [4], in deep belief networks, *Bergstra et al* [5], using assessments from similar problems, *Bardenet et al* [6]

Collaborative Hyperparameter Tuning

- Generalize across similar learning problems, in other words similar datasets
-
- So we sort of built one ☺

No sorting hat ☹

Collaborative Hyperparameter Tuning



(a) Error surface of AdaBoost on lymph (b) Error surface of AdaBoost on sonar

(c) The common latent ranker

Figure 1. (a,b) Error surfaces on two similar datasets have similar shapes although the errors are quantitatively different. (c) The similar shapes can be captured by a latent ranker.

- Similar datasets have similar hyperparameter correlation
- Figure 1 from [6] shows how error surface for similar datasets look similar

Featurize Datasets

- By Dimensions of the dataset
  1. Number of Instances, N_I
  2. Number of Features, N_F
  3. Number of Instances Squared, N_I^2
  4. Number of Features Squared, N_F^2
  5. Number of Instances*Number of Features, $\mathrm{N}_I * N_F$
  6. Number of Instances/Number of Features, $\frac{N_I}{N_F}$
  7. Fraction of Sparse Features
     - features where at most 10% of the instances have a non-zero value

- By feature data type
  1. Fraction of Binary Features
  2. Fraction of Integral Features
  3. Fraction of NonNegative Features
  4. Fraction of Categorical Features

Featurize Datasets

- By distribution of values of the features
  1. Number of Instances with Missing Features
  2. Fraction One Value Features
     – features that have no information in them – that is all have the same values
  3. Fraction of Features with 2 Different Values (Not binary values)
  4. Fraction of Features with 3-10 Different Values
  5. Fraction of Features with 11-20 Different Values

Featurize Dataset Example

- For Example, for the following Breast-cancer dataset

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
1	5	1	1	1	2	1	3	1	1
2	5	4	4	5	7	10	3	2	1
3	3	1	1	1	2	2	3	1	1
4	6	8	8	1	3	4	3	7	1
5	4	1	1	3	2	1	3	1	1
6	8	10	10	8	7	10	9	7	1
7	1	1	1	1	2	10	3	1	1
8	2	1	2	1	2	1	3	1	1
9	2	1	1	1	2	1	1	1	5
10	4	2	1	1	2	1	2	1	1

Number of Instances	Number of Features	Number of Instances Squared	Number of Features Squared	Number of Instances*Number of Features	Number of Instances/Number of Features	Fraction of Sparse Features
10	9	100	81	8100	1.11	0

Binary Features Fraction	Integral Features Fraction	Non-Negative Features Fraction	Categorical Features Fraction
0	1	1	0

Instances with Missing Features	Fraction One Value Features	Fraction of Features with 2 Different Values	Fraction of Features with 3-10 Different Values	Fraction of Features with 11-20 Different Values
0	0	0.11	0.89	0

Extensive Experiments

- Create $\boldsymbol{M}$, a set of models generated with different values of hyperparameters,$\boldsymbol{h}$, for each learner

- These $\boldsymbol{h}$ values are obtained by discretizing the space of the hyperparameters and choosing a finite set.

- Execute models in $\boldsymbol{M}$ for a set of datasets $\boldsymbol{D} = \{\boldsymbol{D_1}, \dots, \boldsymbol{D_n}\}$

- Currently for linear learners

- For example, for Fast Tree Binary Classification
  - $h = \{iter, nl, mil, lr\}$
    - iter -> Number of Trees
    - nl -> Number of Leaves
    - mil -> Minimum documents in leaf
    - lr -> learning rate
  - Discretized hyperparameter space
    - Iter = 20,100,500
    - nl = 2-128;log;inc:4
    - mil = 1,10,50
    - lr = 0.025-0.4;log
  - $$M = \{\{100,4,10,0.3\} \\ \vdots \\ \{20,8,100,0.025\}\}$$

Create Historical Knowledge Base of Results

- Record the results of the experiments
  - AUC
- Generate $\mathrm{A} = MxD$ matrices with
  - Log-normal AUC
  - Log-normal (1-AUC)
  - Order Statistics
    - $\forall M_i \in M$ rank by AUC for each dataset

Log-normal AUC

$$A_{i,j} = \frac{\log(AUC_{i,j}) - mean(\{\log(AUC_{1j}), \ldots, \log(AUC_{|M|,j}\})}{var(\log(AUC_{\cdot j}))}$$

$AUC\ is\ the\ MxD\ matrix\ with\ auc$

$i \in [1, |M|], j \in [1, |D|]$

Log-normal (1-AUC)

$$A_{i,j} = \frac{\log(AUC_{i,j}) - mean(\{\log(AUC_{1j}), \ldots, \log(AUC_{|M|,j}\})}{var(\log(AUC_{\cdot j}))}$$

$AUC\ is\ the\ MxD\ matrix\ with\ auc$

$i \in [1, |M|], j \in [1, |D|]$

Order Statistics

M/D	D_1	D_2	D_3	D_4
M_1	0.9075	0.9174	0.8796	0.9820
M_2	0.8883	0.8776	0.9058	0.8806
M_3	0.9914	0.9894	0.9933	0.9737

Table 1 AUC from experiments

M/D	D_1	D_2	D_3	D_4
M_1	2	2	3	1
M_2	3	3	2	3
M_3	1	1	1	2

Table 2 Order Statistics

Generate smart sweeps on demand

1. Featurize new dataset D_{new}
2. Find the K most similar datasets of D_{new} in D
   - Using KNN
   - Using Bayesian Sets [7]
3. For each model in M compute the average of the AUC for the K most similar datasets, $A_{avg} = \frac{\sum_{D_j \in D_k} A(M_i, D_j)}{k}$
4. Rank the models in M by the above average
5. Choose top s ranked models or sweeps
   - With Diversity
   - Without diversity

Diversity Coefficient d

- Introduces diversity in the models selected by filtering out similar models

- The number of models to be filtered around each s required sweeps is computed as:

$$f = \frac{|M| * d}{s}$$

  $|M|$-> number of models
  n -> number of datasets
  s -> number of sweeps
  d -> diversity coefficient $d \in [0,1]$

- Pseudo code for model filtering

```
for(int i = 0;i < s;i++){
    select M_i ∈ M with highest ranking
    ∀M_j ∈ M, j ≠ i compute Distance(M_i, M_j)
    filter out f most similar models
        M = M − M_i − f models similar to M_i
}
```

- For example
  - $M = 10, d = 0.8, s = 4$
  - $f = \frac{10*0.8}{4} = 2$

M/D	D_1	D_2	D_3	D_4	Ranking
M_1					6
M_2					8
M_3					2
M_4					3
M_5		Log-normal AUC or Log-normal (1-AUC) or Order Statistics			10
M_6					7
M_7					1
M_8					4
M_9					9
M_{10}					5

$$s = \{M_1, M_4, M_7, M_{10}\}$$

Experiment Setup

- Models
  - Currently for Binary Classifiers Only

Learners	Logistic Regression	Fast Tree	Fast Forest	Average Perceptron	LDSVM	Linear SVM	Binary Neural Net
Hyper Params	**l1**=0-2;steps:10 **l2**=0-2;steps:10 **ot**=1e-4,1e-5,1e-6 **m**=5-50;inc:15 **norm**=Gaussian,MinMax{zero+},MinMax,Bin	**iter**=2-16384;log;inc:2 **nl**=2-1024;log;inc:2 **mil**: 10-10e4;log;steps:8 **lr**=0.1-0.5;inc:0.1 **ff**=0-1;inc:0.2	**Iter**=2-16384 ;log;inc:2 **nl**=2-1024;log;inc:2 **mil**=10-12e4;log;inc:4 **bagfrac**=0.5,0.7,0.9 **ff**=0.5,0.7,0.9 sf=0.5,0.7,0.9	**loss**=HingeLoss **lr**=0.01,0.05,0.1,0.5,1 **l2**=0-2;steps:5 **iter**=2-16384;log;inc:2 **decreaselr**=+|- **initwts**:0-1;inc:0.2 **norm**=Gaussian,MinMax{zero+},MinMax,Bin	**depth**=0,3,5,7 **lw**=0.1,0.01,0.001 **lt**=0.1,0.01,0.001 **lp**=0.1,0.01,0.001 **s**=1.0,0.1,0.01 **iter**=10000,15000,20000 **norm**=Gaussian, MinMax{zero+}, MinMax,Bin	**lambda**=1e-5-1;log;inc:5 **iter**=1-10e4;log;inc:2 **initwts**=0.0-1.0;inc:0.1 **norm**=Gaussian,MinMax{zero+},MinMax ,Bin	**hidden**=20,100,1000 **iter**=20-160;log;inc:2 **lr**=0.001-1.0;log;inc:10 **initwts**=0.001-1.0;log;inc:10
Number of Models	8397	12600	22680	21000	4860	6160	1920

Table 3 Learners and their hyperparameter values for which experiments were run

Experiment Setup

- Datasets (\\tspace09\data\BinaryClassification)

Dataset	Instances	Features	Numeric (Int/Real)	Binary	Categorical	Text	Sparse	Missing Values
CCSChallenge	10000	100	Yes	No	No	No	Yes	No
CoptTest	12111	116	No	Yes	No	No	Yes	No
Father	7128	2527	No	Yes	No	No	Yes	Yes
Hyperonym	12837	50035	No	Yes	No	No	Yes	No
NewsHardware	959	35213	No	Yes	No	No	Yes	No
Sentiment	1400	24531	No	Yes	No	No	Yes	No

Table 4 MLComp Datasets (http://mlcomp.org/)

Dataset	Instances	Features	Numeric (Int/Real)	Binary	Categorical	Text	Sparse	Missing Values
BreastCancer	476	9	Yes	No	No	No	No	Yes
InternetAd	1634	1558	Yes	Yes	No	No	No	Yes
Ionosphere	351	34	Yes	Yes	No	No	No	No
SeismicBumps	2584	24	Yes	Yes	Yes	No	No	No
Adult	32561	108	Yes	Yes	Yes	No	No	No
Census_KDD	199523	515	Yes	Yes	Yes	No	No	No

Table 5 UCI Datasets (https://archive.ics.uci.edu/ml/datasets.html)

Experiment Setup

- More Datasets

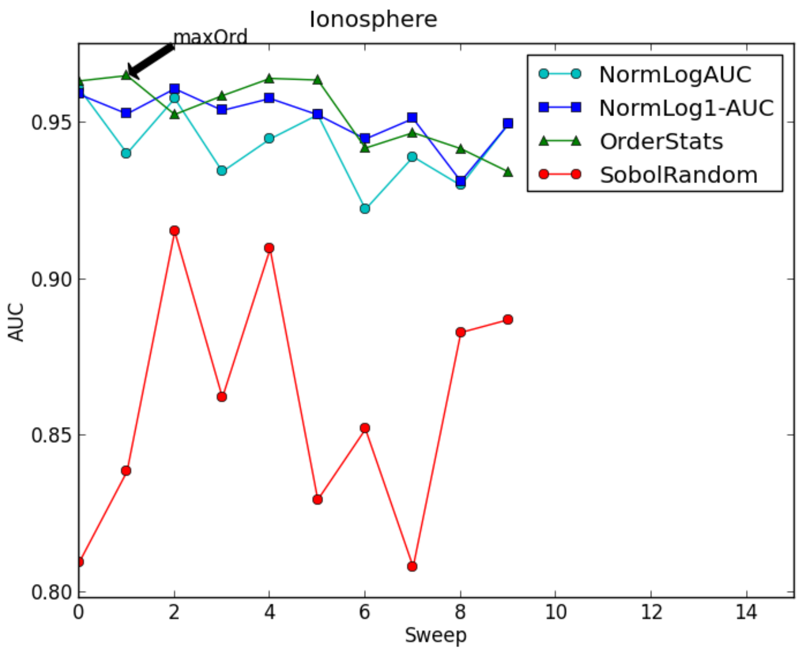
Dataset	Instances	Features	Numeric (Int/Real)	Binary	Categorical	Text	Sparse	Missing Values
Enron	57607	91397	Yes	No	No	No	Yes	Yes
RCV1	781265	47153	Yes	No	No	No	Yes	No
YearPrediction	463715	90	Yes	No	No	No	No	No

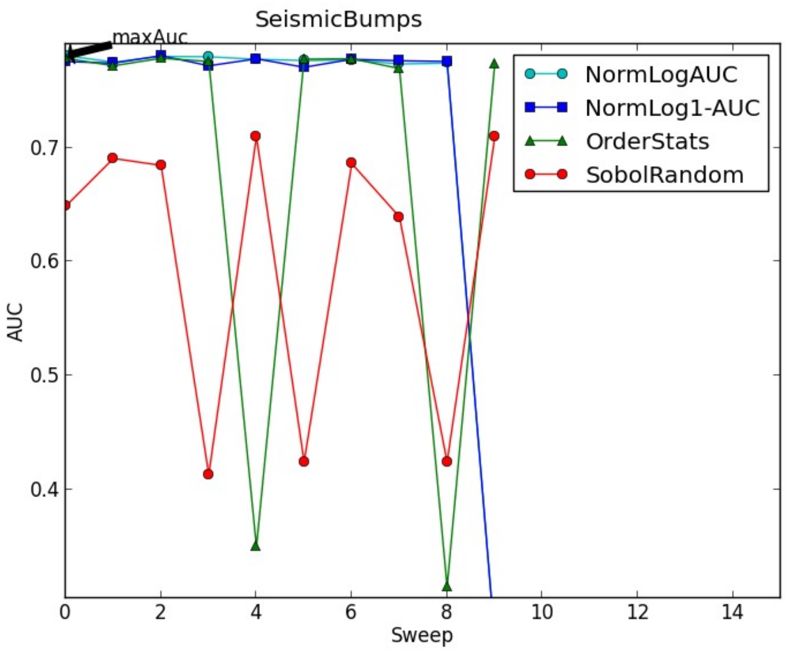
Table 6 Other Datasets

- Compute Resources for running experiments
  - MSR Cluster
  - TLCHPCK
  - MLC

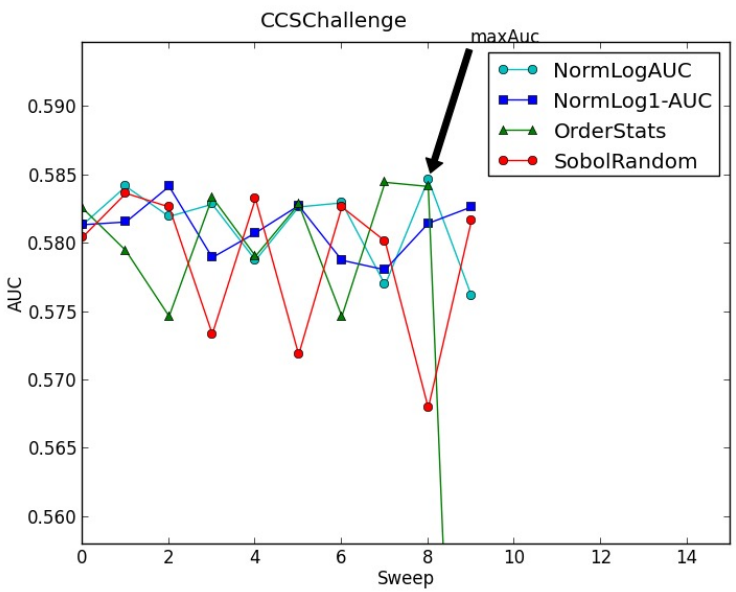
Results - Leave One Out Cross Validation

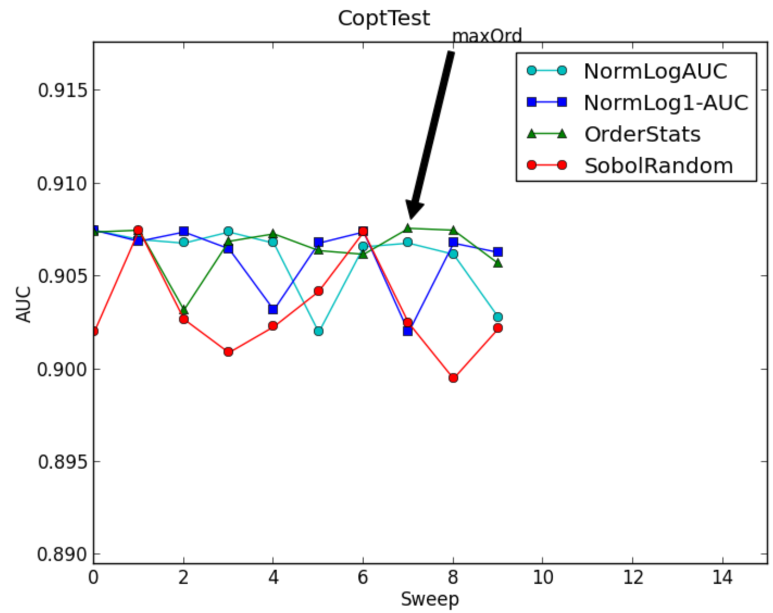
- Learner = Logisitic Regression, K for KNN = 3, Number of sweeps S = 10, Diversity Coefficient = 0.8

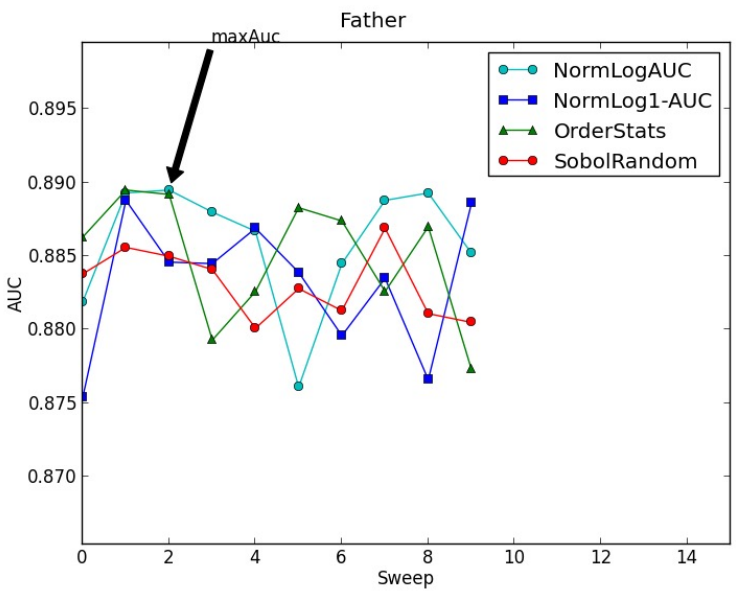


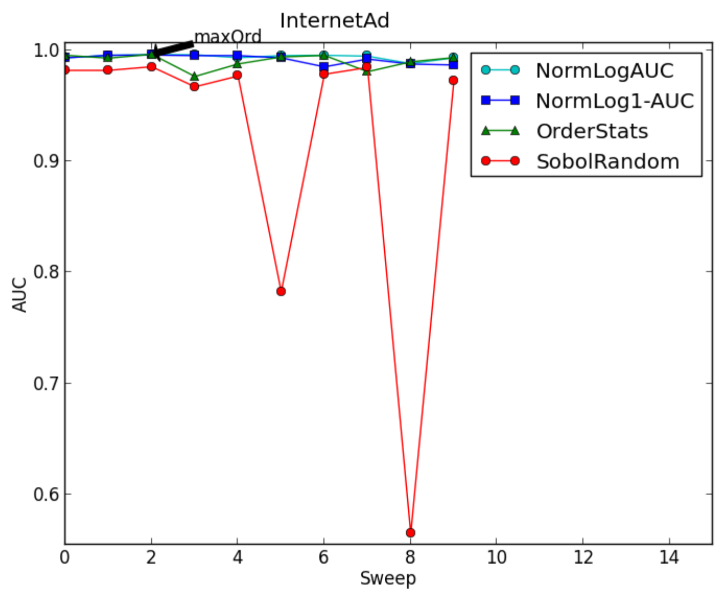


Results - Leave One Out Cross Validation



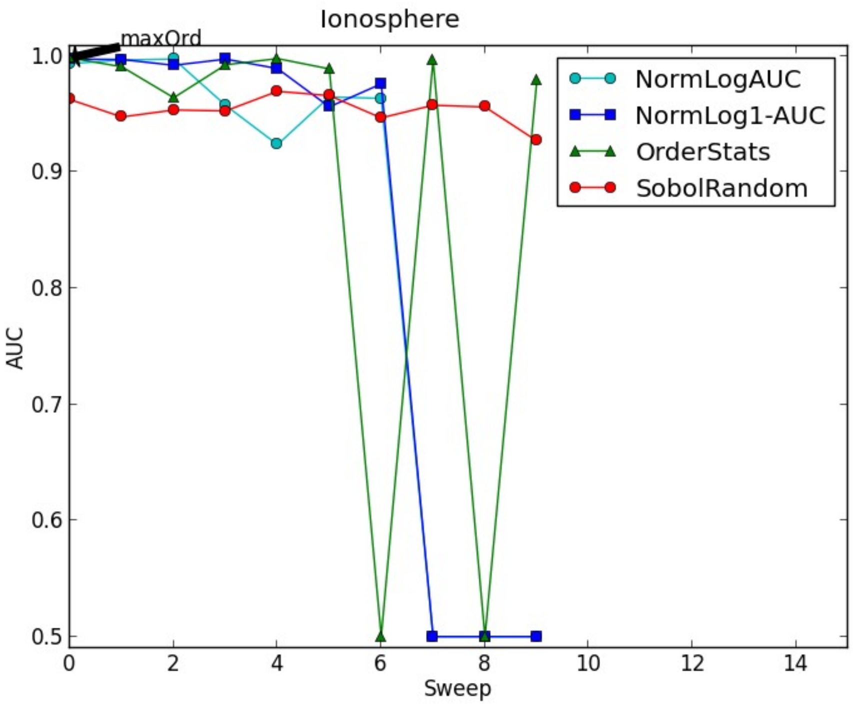


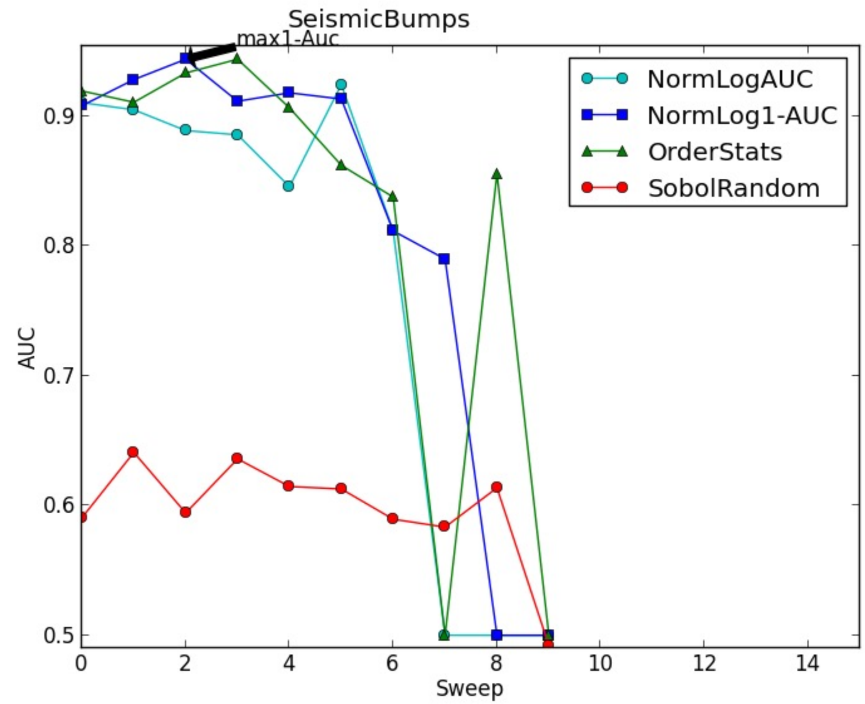




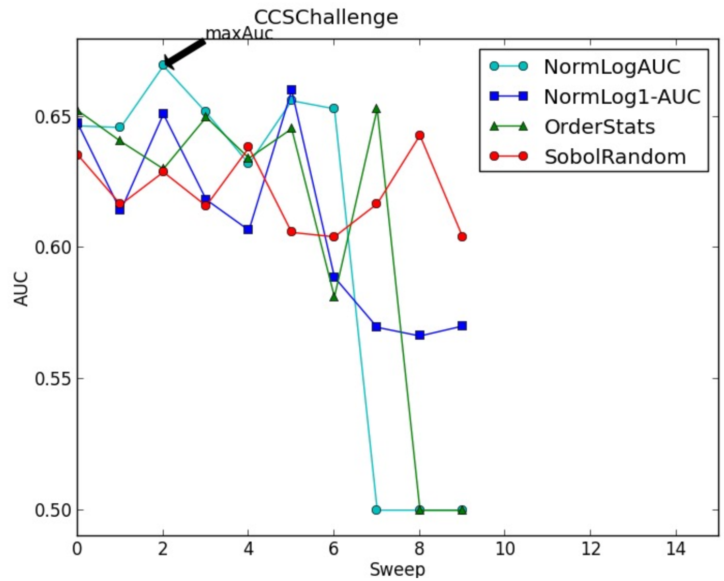
Results - Leave One Out Cross Validation

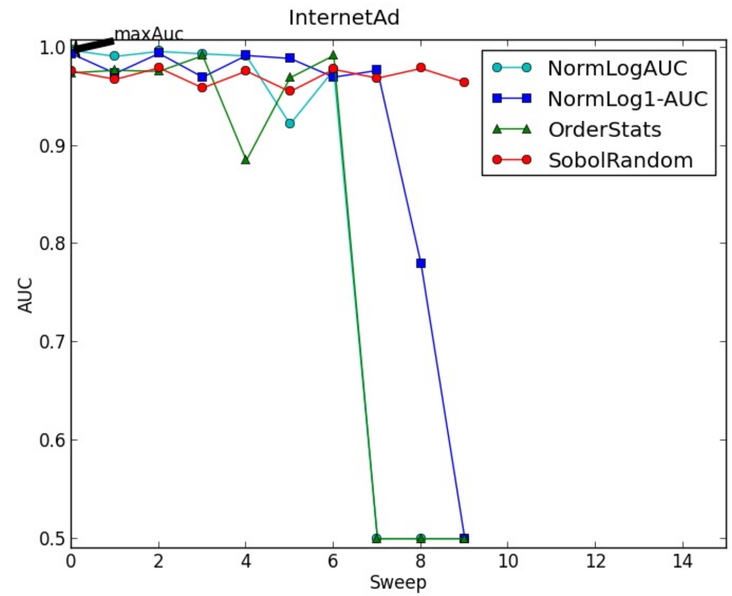
- Learner = Fast Tree, K for KNN = 3, Number of sweeps S = 10, Diversity Coefficient = 0.8

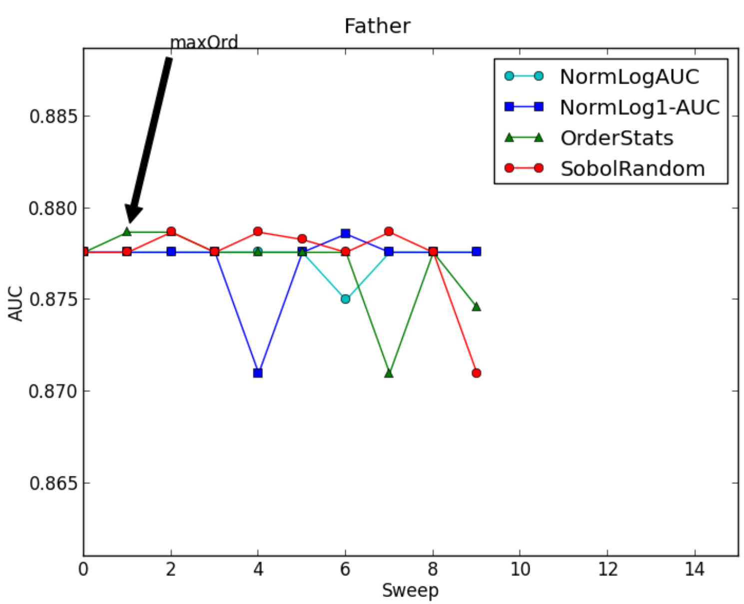


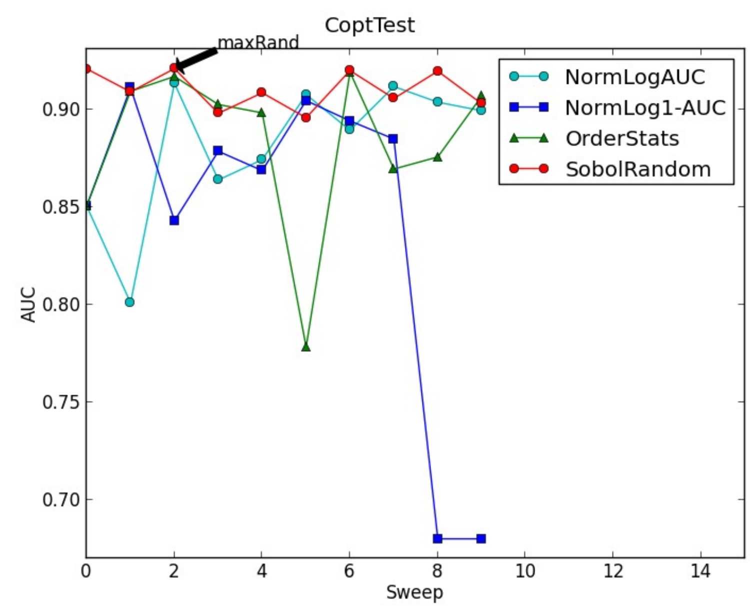


Results - Leave One Out Cross Validation

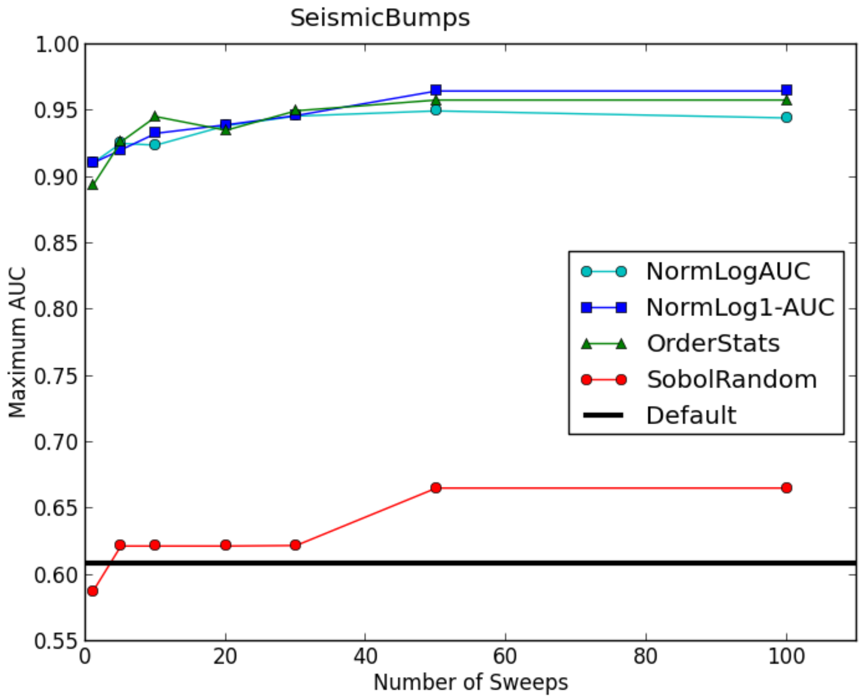


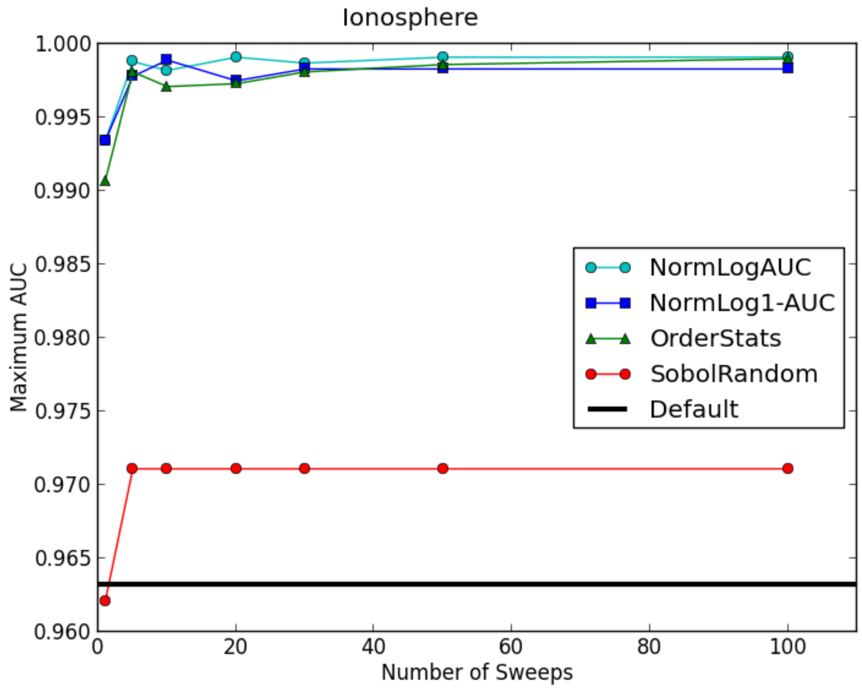




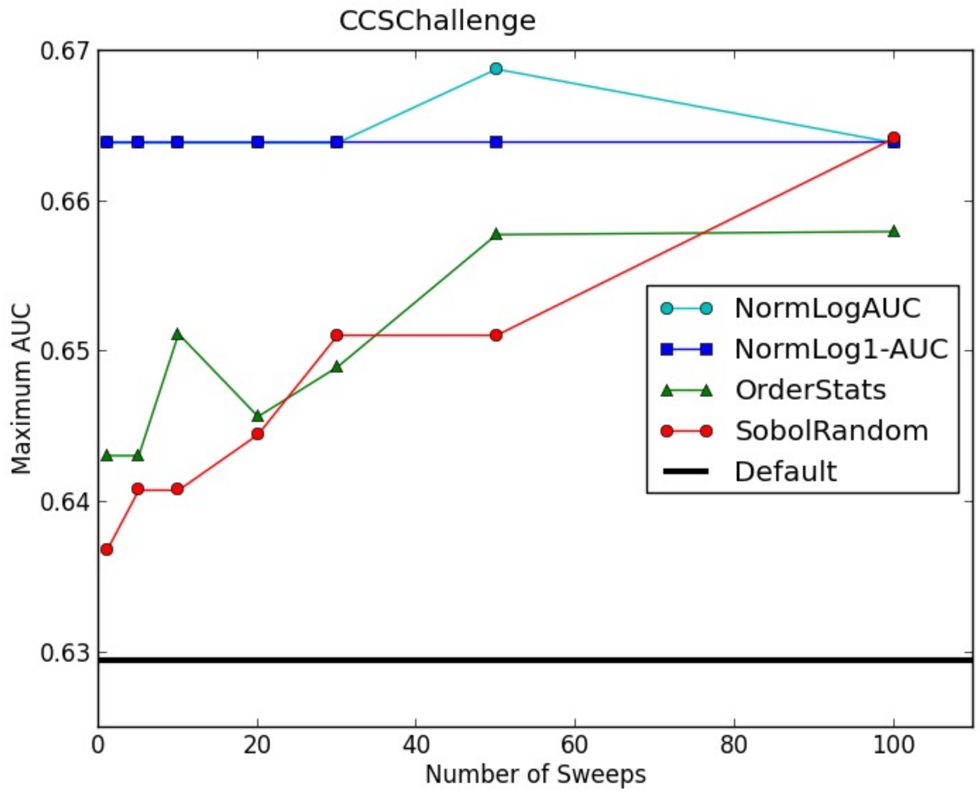


Results – Comparison over Sweeps for Fast Tree

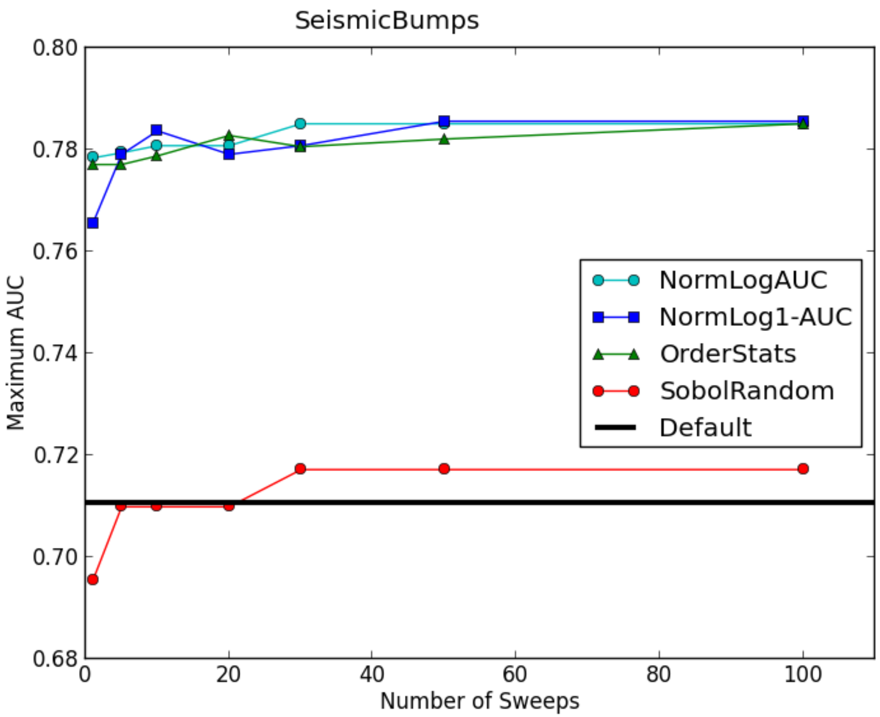


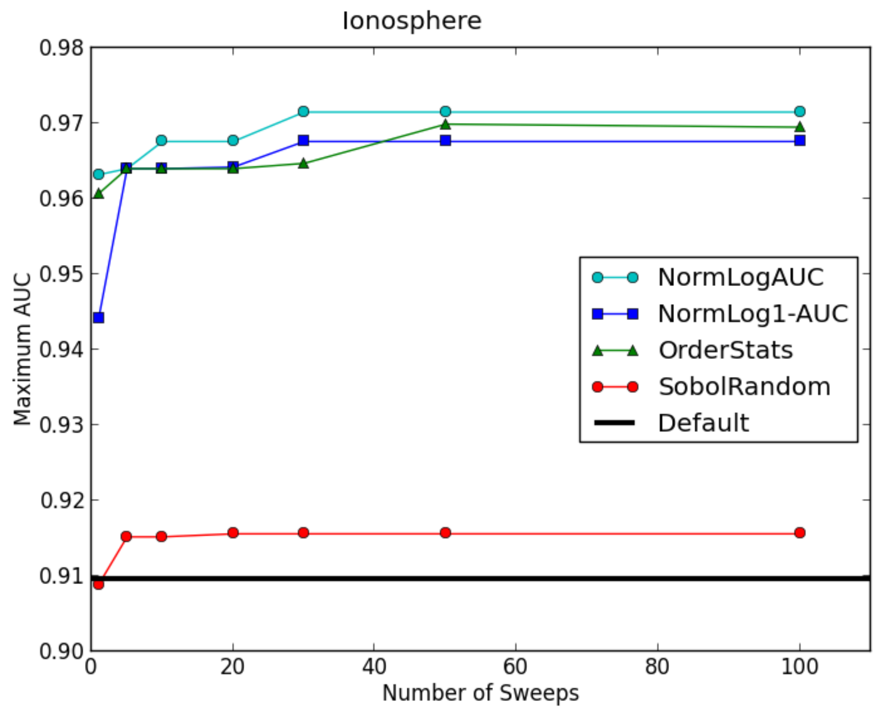


Results – Comparison over Sweeps for Fast Tree

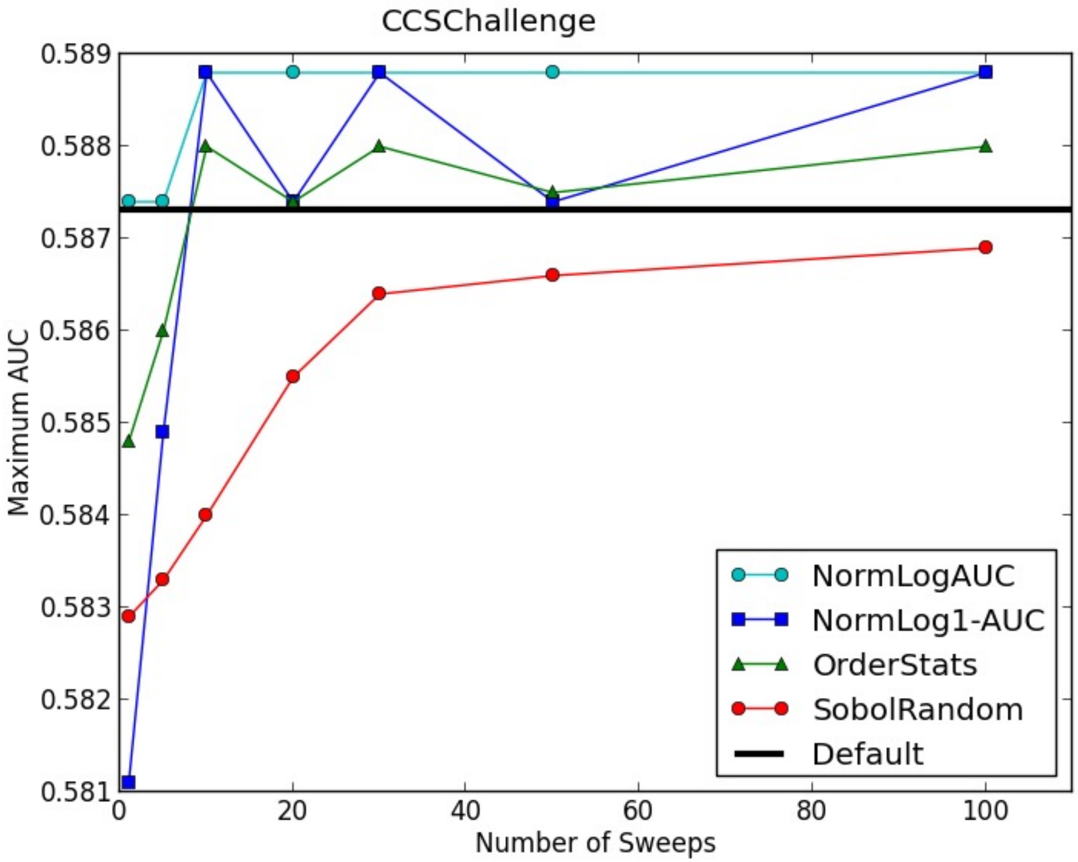


Results – Comparison over Sweeps for Logistic Regression





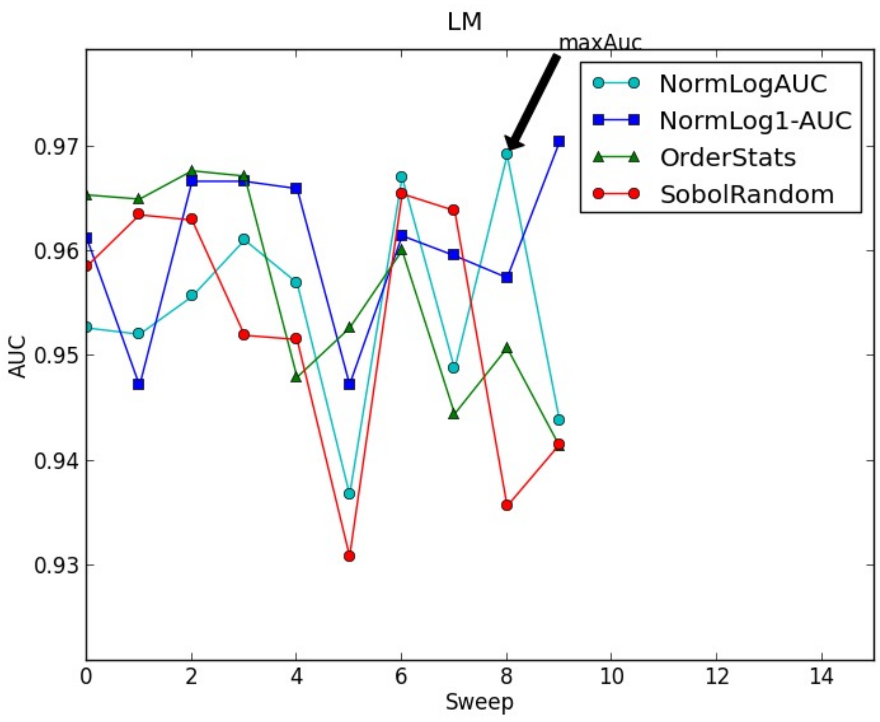
Results – Comparison over Sweeps for Logistic Regression



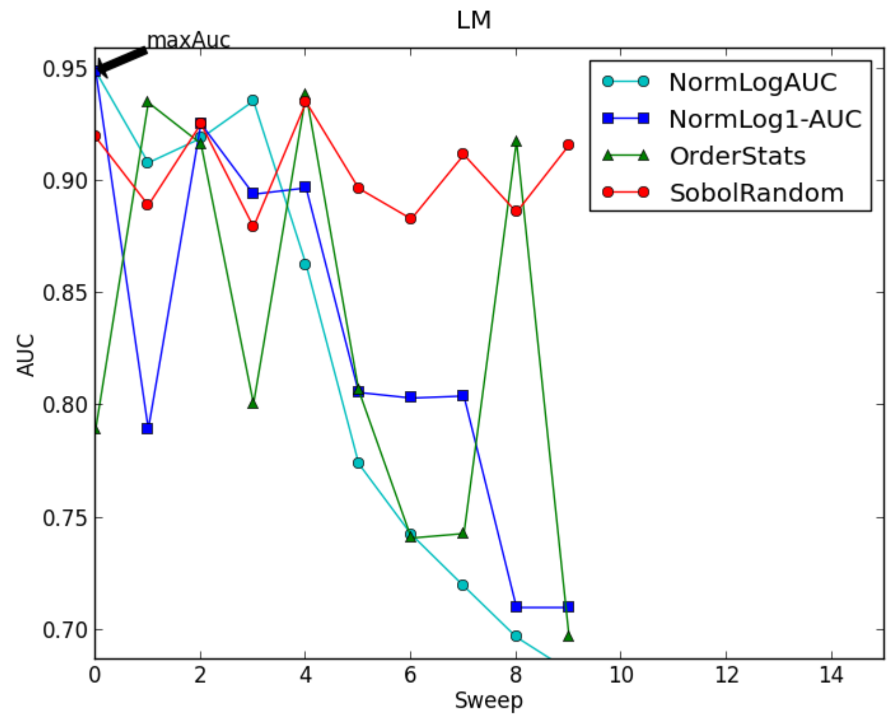
Results – New Dataset

Dataset	Instances	Features	Numeric (Int/Real)	Binary	Categorical	Text	Sparse	Missing Values
LM	4512	65536	No	No	No	Yes	No	No

Table 7 New Dataset



Logistic Regression

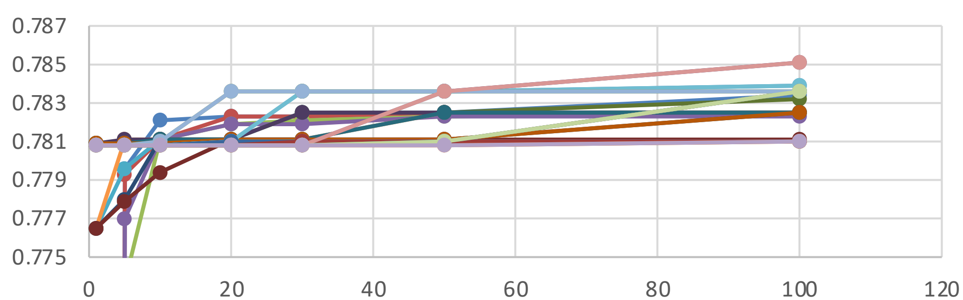


Fast Tree

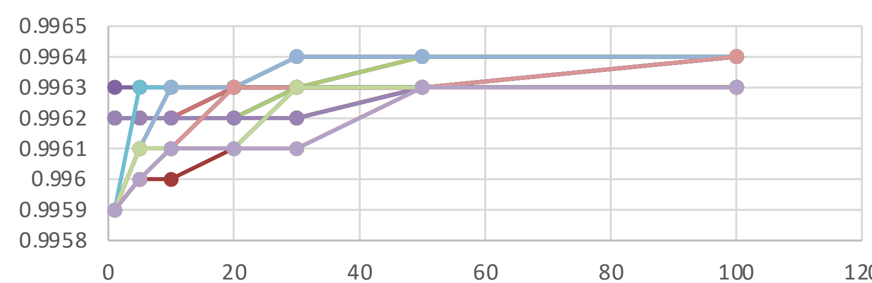
Accuracy and Number of Sweeps

- x-axis is number of sweeps and y-axis is AUC
- Each line corresponds to the max AUC for each sweep for a particular value of K and diversity co-efficient
- Accuracy increases with number of sweeps

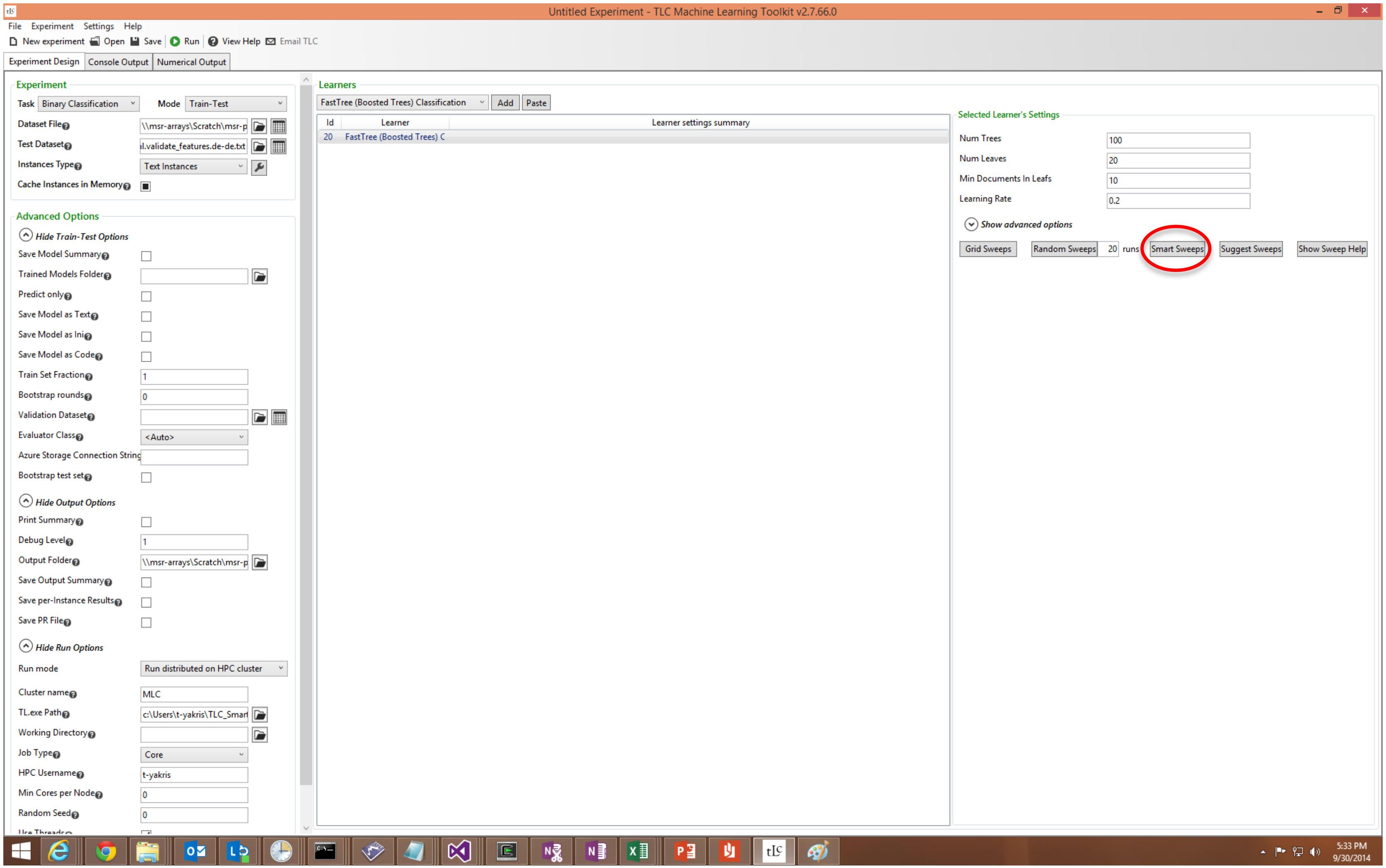
LR, dataset 6 (SeismicBumps)
random: 5 sweeps 0.7029, 100 sweeps 0.7247



LR, dataset 3 (Breast-Cancer)
random: 5 sweeps 0.9952, 10 sweeps 0.9955, 20 sweeps 0.996



Incorporating Smart Sweep in AzureML



Future Directions

- Create a more comprehensive past experiments knowledge base with more datasets
- Determine additional dataset features
- Measure model execution time accurately and use it for model selection
- Extend to other learners
- Use experimental results as prior for Bayesian Inference and other optimization techniques
- Algorithm recommendation

References

1. Hutter, F., Hoos, H. H., Leyton-Brown, K., and Stützle, T. ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, October 2009.
2. Nannen, V. and Eiben, A. E. Relevance estimation and value calibration of evolutionary algorithm parameters. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 975–980, 2007.
3. Snoek, J., Larochelle, H., and Adams, R. P. Practi-cal Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Process- ing Systems*, volume 25, 2012.
4. Thornton, C., Hutter, F., Hoos, H. H., and Leyton- Brown, K. Auto-WEKA: Automated selection and hyper-parameter optimization of classification algorithms. Technical report, http://arxiv.org/abs/ 1208.3719, 2012.
5. Bergstra, J. and Bengio, Y. Random search for hyperparameter optimization. *Journal of Machine Learning Research*, 2012.
6. Bardenet R., Brendel M., Kégl B., and Sebag M. Collaborative hyperparameter tuning. In *Proceedings of ICML-13*, 2013.
7. Ghahramani Z. and Heller K. A. Bayesian sets. In *NIPS*, 2005.