Homework Assignment 3
CSE 291 Spring 2011
Classifier Parameter Optimization by AUC Maximization

Jihoon Kim
UCSD Division of Biomedical Informatics
j5kim@ucsd.edu

Hari Damineni
UCSD Computer Enginering & Science
hdaminen@cs.ucsd.edu

André Christoffer Andersen
NTNU Industrial Economics
andrecan@stud.ntnu.no

April 26, 2011

1 Introduction

In this assignment we compare the classification performances of Support Vector Machine (SVM) and Neural Network (NN) using area under the receiver operating characteristic curve (AUC) as a score function. Also, we will compare two different methods for handling class imbalance problems; biases cost per class and oversampling.

The assignment divides naturally in to the following phases

1. Pre-processing

2. Feature selection through logistical regression

3. Search for an AUC maximizing parameter

The main results are displayed in Table 1.

Model	Biased costs	Oversampling
Support Vector Machine	0.607	0.631
Neural Network	0.633	0.659

Table 1: The Best AUC results

2 Dataset

The dataset in question is the `KDD-CUP-98` dataset form the Second International Knowledge Discovery and Data Mining Tools Competition. It is a collection of data on people who have donated to a national veterans organization. Data mining task is to predict who will donate money to the participating organization. In this assignment we will be using the entire original dataset of 95412 instances, with a transformed subset of the original 481 features.

3 Pre-processing

3.1 Data cleaning

The pre-processing phase goes through add, discard, dummify, impute, normalize and split. R is used for these task as programming is needed.

Add: introduces a binary missingness feature (1 if missing, otherwise 0) for each existing feature that has at least one missing value will make sure that latent information is not discarded.

Discard: removes instances whose outcome value is missing, because these obviously have no predictive value.

Dummify: introduces $k-1$ new binary features for each k-valued categorical feature. One is dropped in order to prevent linear dependence.

Impute: the missing value are replaced with the column mean for numeric feature and the mode for binary feature.

Normalize: performs min-max scaling to numeric feature to have the range $[0, 1]$. This ensures SVM model does not get biased toward a feature with extreme values

Split: replaces multi-modal features into separate categorical features then dummify each categorical features.

3.2 Feature selection

We use a logistical regression in order to identify the most predictive features. We select features whose P-values are less than 0.05. This leaves 30 features for modeling.

3.3 Oversampling

The `KDD-CUP-98` dataset is class-imbalanced in the sense that there are many more negative labeled instances than positive labeled instances. Because of this we have generated two datasets:

- *Intact dataset (D_1):* This dataset is the pre-processed original data containing all 90,569 negative and 4,843 positive instances.
- *Oversampled dataset (D_2):* We oversampled the 87,174 positive instances and added to the intact data set D_1.

4 Experiment

4.1 Comparison of SVM and NN

Our goal is to compare the performances of a linear classifier support vector machines (SVM) and a non-linear classifier Neural Network (NN) using Area

Under the ROC Curve (AUC). As a part of our modeling process, we construct two models based on a linear SVM and a non-linear NN.RapidMiner processes are illustrated in Figure 1.

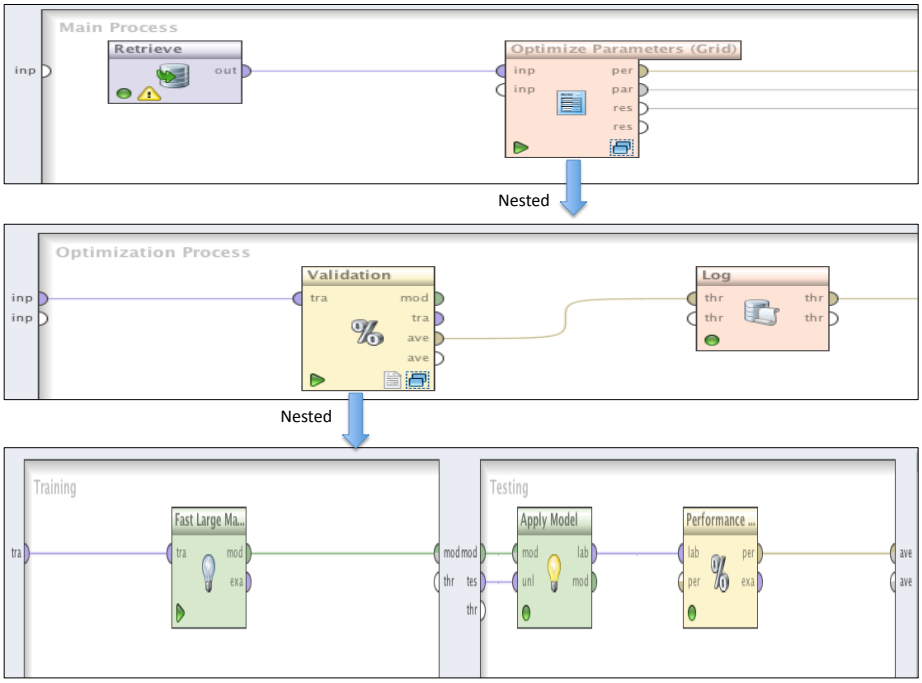

Figure 1: Screen shot of 3-level nested processes in RapidMiner.

We model RapidMiner processes in 3-level hierarchy. In level 1, *Optimize Parameters (Grid)* is used to find the optimal values of the parameters. In level 2, nested in *Optimize Parameters*, *Validation* is used for cross-validation then the performance score is recorded using *Log* process. Lastly, in level 3, nested in *Validation*, *Fast Large Margin SVM* is trained on the trainiing-set then *Apply Model* and *Performance* is used for testing. Similarly, we experiment with NN by replacing *Fast large Margin* with *Neural Net* process. We use the same random seed '1234' to have the same training sets between SVM and NN. After obtaining model parameters after cross-validation, RapidMiner *ROC Comparator* operator is used to visually illustrate AUCs of SVM

and NN models.

4.2 Comparison of Biased Costs and Oversampling

For each dataset D_1 and D_2, we develop two classifiers using `FastLargeMargin` SVM and NN in RapidMiner. The cost parameters, C, and learning rates are calibrated using grid-search 10-fold cross-validation for SVM and NN models respectively.

Biased Cost:

$$c(f) + C_{-} \sum_{i:y=-1} l(f(x_i), -1) + C_{+} \sum_{i:y=1} l(f(x_i), 1)$$

We apply biased costs to the intact dataset D_1. Here we use two cost parameters, C_{-} and C_{+} for the negative and positive examples which serve as emphasis coefficients for the relevant classes in the following loss minimization equation according to equation 1. C_{+} parameter enables us to compensate for the imbalance of the positive examples in the whole dataset.

Oversamping:

$$c(f) + C \sum_{i} l(f(x_i), y_i)$$

We use oversampled dataset D_2 to emulate the biased costs. Here, only a single parameter C is tuned as the oversampling of positive examples renders the dataset balanced in between positive and negative examples. Equation 2 describes the SVM loss minimization equation for D_2. We choose to oversample thepositivelabeled examples rather thanundersamplingthe negative labeled examples in order to avoid information loss

5 Results

5.1 Results of SVM *vs.* NN

Table 2 and 3 illustrate the results obtained during modeling of D_1 after 10-fold cross-validation using Fast Large Margin SVM. We can see that the AUC remains 0.607 across the grid. We choose C- and C+ values of 1 and 10 as they seem to produce both precision and recall values over 10%.

C-	C+	Prediction	Outcome 0	Outcome 1
1	1	0	90569	4843
		1	0	0
1	5	0	90493	4815
		1	76	28
1	10	0	85210	4164
		1	5359	679
1	15	0	69166	2889
		1	21403	1954
1	20	0	51672	1863
		1	38897	2980

Table 2: SVM confusion matrix on $D1$

C-	C+	Accuracy	Precision	Recall	AUC	Lift	F1 Score
1	1	94.92	Unknown	0	0.607	Unknown	Unknown
1	5	94.87	28.55	0.58	0.607	562.53	1.13
1	10	90.02	11.25	14.02	0.607	221.57	12.48
1	15	74.54	8.37	40.35	0.607	164.84	12.86
1	20	57.28	7.12	61.53	0.607	140.19	12.76

Table 3: SVM performance matrix on $D1$

Learning Rate	Prediction	Outcome 0	Outcome 1
0.01	0	90569	4843
	1	0	0
0.05	0	90534	4832
	1	35	11
0.125	0	90511	4827
	1	58	16
0.3	0	90478	4829
	1	91	14
0.75	0	90532	4828
	1	37	15
1	0	90495	90495
	1	74	11

Table 4: NN confusion matrix on $D1$

In table 4 and 5 you can see the results for the NN model after 10-fold cross-validation on D_1. A learning rate of 0.05 is the best one since it produces the highest AUC of 0.633. Other models with higher true-positive rates are not chosen as our objective measure is highest AUC. For D1, NN model performs better with an overall AUC of 0.633 versus an AUC of 0.607 for SVM model. The ROC curve for SVM ($C_- = 1, C_+ = 10$) and NN (learning rate = 0.05), in Figure 2, visually illustrates the AUCs and the performance difference between SVM and NN for D_1. From this ROC curve, it is clear that NN model performs better than SVM model in terms of AUC. However, it should be noted that some thresholds do exist where SVM model fares equal to or better than the NN model in the range of false-positive rate (x-axis) [0.4, 0.65].

Learning Rate	Accuracy	Precision	Recall	AUC	Lift	F1 Score
0.01	94.92	Unknown	0	0.624	Unknown	Unknown
0.05	94.9	29.66	0.23	0.633	584.22	0.45
0.125	94.88	21.65	0.33	0.592	426.6	0.65
0.30	94.84	13.33	0.29	0.559	251.02	0.57
0.75	94.9	28.85	0.31	0.557	568.3	0.61
1.00	94.86	13.07	0.23	0.549	257.43	0.45

Table 5: NN performance matrix on $D1$

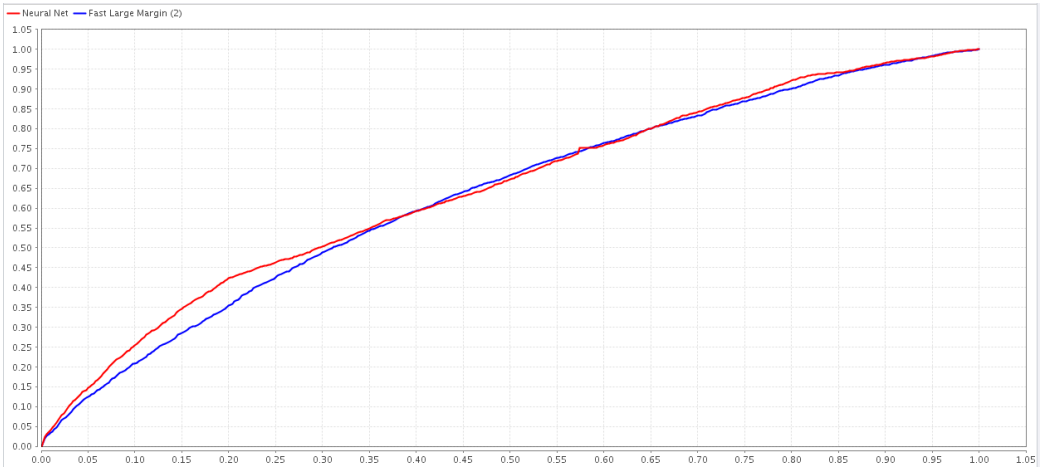

Figure 2: ROC curve for NN and SVM from D_1

5.2 Results of Biased Costs *vs.* Oversampling

Table 6 and 7 shows the results of SVM, and Tables 8 and 9 shows the results NN modeling, both after 10-fold cross-validation on D_2. In table 7 it is clear that the SVM model from D_2 yields an AUC of 0.631 across C values. $C = 2$ is chosen as the final SVM model parameter as it represents higher true-rates without a big increase in the false-rates.

From Table 9 we can see that the highest AUC value, 0.659, for the NN model occurs at learning rate 0.75. For D_2, NN model performs better with an overall AUC of 0.653 versus an AUC of 0.631 for SVM model. The ROC

C	prediction	outcome 0	outcome 1
0.5	0	58640	39825
	1	31929	47349
1	0	58609	39771
	1	31960	47403
2	0	58697	39769
	1	31972	47405
4	0	58610	58610
	1	31959	47392

Table 6: SVM confusion matrix using 10-fold cross-validation on D_2

C	Accuracy	Precision	Recall	AUC	Lift	F1 Score
0.5	59.63	59.73	54.32	0.631	121.78	56.89
1	59.64	59.73	54.38	0.631	121.78	56.93
2	59.64	59.72	54.38	0.631	121.77	56.92
4	59.64	59.72	54.36	0.631	121.77	56.92

Table 7: SVM performance matrix on D_2

Learning Rate	Prediction	Outcome 0	Outcome 1
0.3	0	68326	45350
	1	22244	41824
0.75	0	70566	48401
	1	20003	38773
1	0	77774	60903
	1	12795	26271

Table 8: NN confusion matrix using 10-fold cross-validation on D_2, the over-sampled dataset

Learning Rate	Accuracy	Precision	Recall	AUC	Lift	F1 Score
0.30	61.97	65.28	47.98	0.657	133.10	55.31
0.75	61.52	66.04	44.48	0.659	134.64	53.05
1.00	58.54	69.69	30.14	0.645	142.10	39.76

Table 9: NN performance matrix on D_2, the oversampled dataset.

curve for SVM ($C = 2$) and NN (learning rate = 0.75), in Figure 3, visually illustrates the AUCs and the performance difference between SVM and NN for D_1.

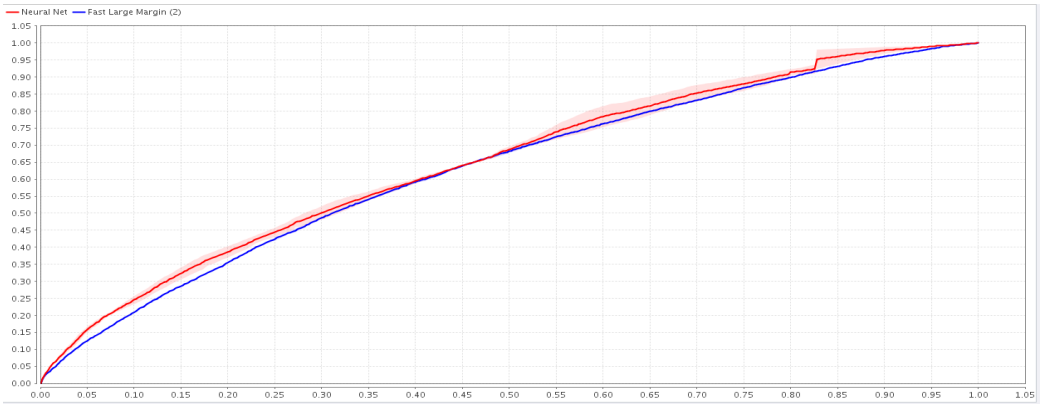

Figure 3: ROC curves for NN and SVM from D_2, the oversampled dataset

From the ROC curve in Figure 3, it is clear that NN model performs better than SVM model in terms of AUC. However, it should be noted that some thresholds do exist where SVM model fares equal to the NN model in the range of false-positive rate (x-axis) [0.4, 0.5].

6 Conclusion

Non-linear NN model produced better AUC values than SVM model on the same dataset. This leads us to believe that there are some attributes which bear a non-linear relation to the output label and cause the linear-model to under-perform compared to the non-linear models.

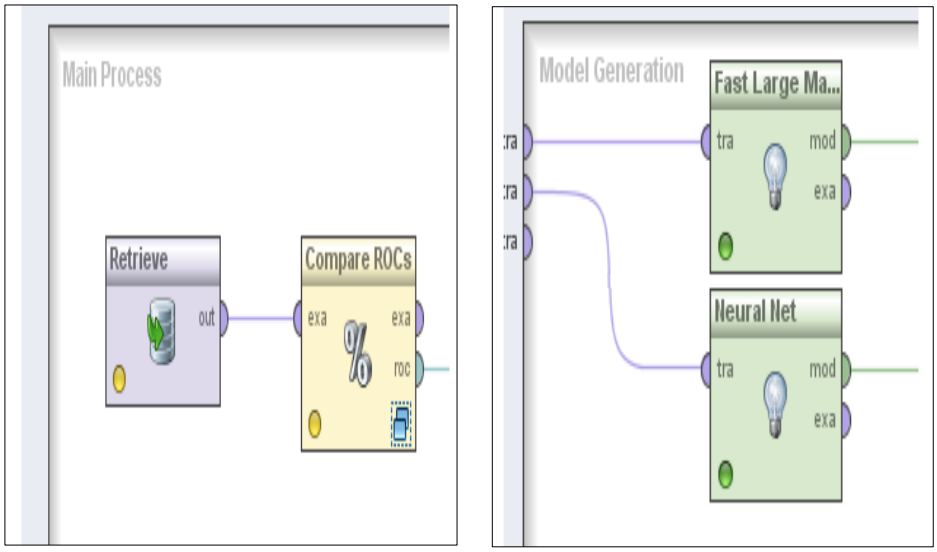

Figure 4: ROC process. Left: main process and Right: nested process within Compare ROCs

Though NN model leads the SVM model with a better AUC, it should be noted that there are certain thresholds at which SVM model performs as good as or better than the NN model. If those thresholds are used as decision criteria, then an SVM model can be preferred to the NN model for modeling purposes.

We prefer biased costs over oversampling as the latter often produces misleading results. SVM, for example, is similar to edited nearest neighbor classifier. Then all newly added the duplicated instances will result in "zero" distances thereby "zero" errors, which is misleading. Also, in sampling scheme, how to sample is also challenging and might lead to contamination of original data, making the derived classifier invalid.