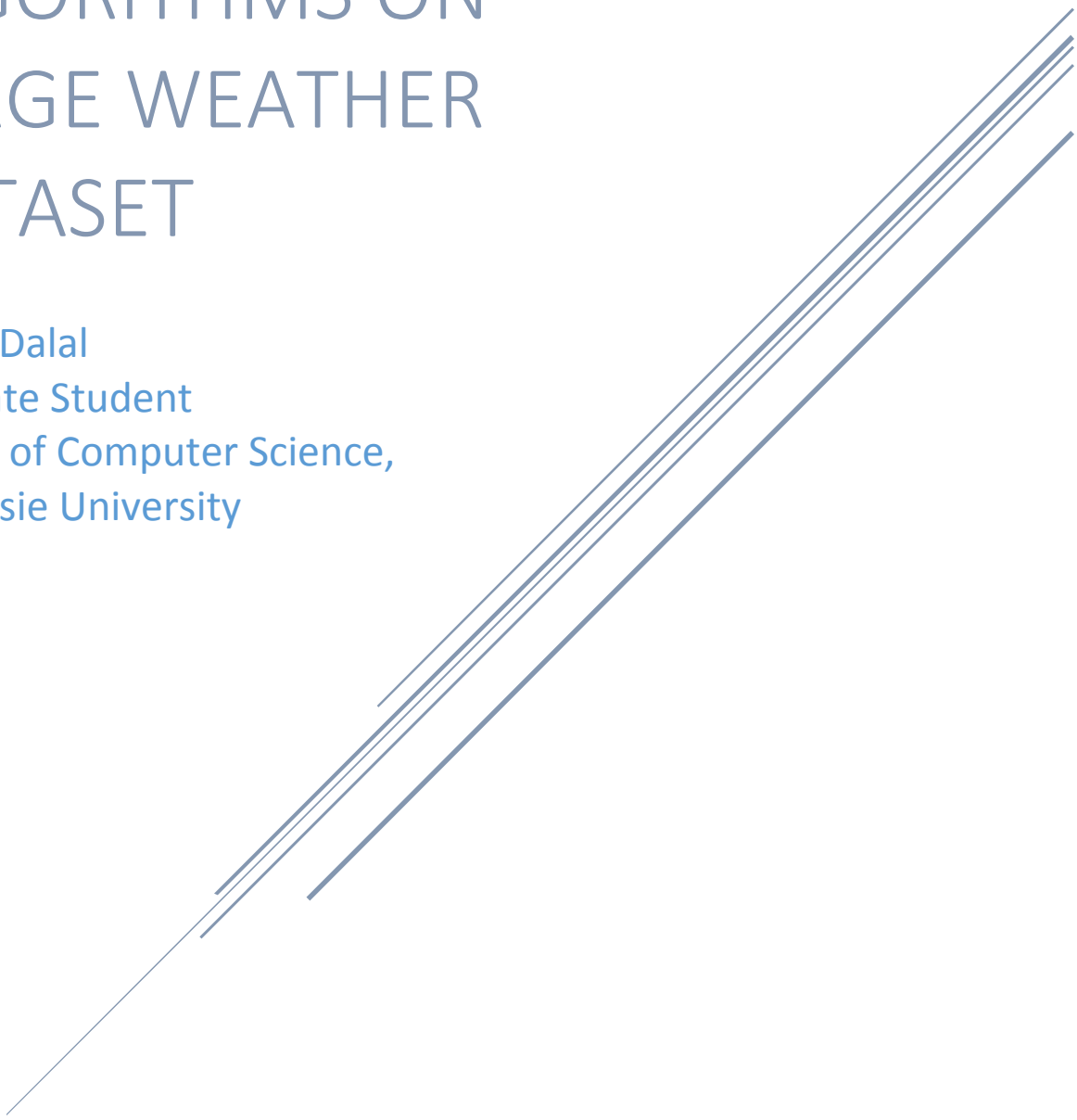
PERFORMANCE ANALYSIS OF CLASSIFICATION ALGORITHMS ON LARGE WEATHER DATASET

Hardik Dalal
Graduate Student
Faculty of Computer Science,
Dalhousie University



Contents

Summary 2
Data 2
Data Description 2
Data Cleaning 2
Algorithms 3
Naïve Bayes 3
Decision Tree 3
Experimental Results 3
Performance Measures 4
Performance Analysis 4
Performance of Naïve Bayes before feature selection and re-sampling: 4
Performance of Decision Tree before feature selection and re-sampling: 4
Performance of Naïve Bayes after feature selection and re-sampling 5
Performance of Decision Tree after feature selection and re-sampling 5
Performance of Naïve Bayes with 10-fold cross validation 6
Performance of Decision Tree with 10-fold cross validation 6
Conclusion 7

Summary

The project presents analysis on weather of Halifax using Weka GUI. There are total of 23 variables/features which represent the weather condition recorded every hour of the day. The data is cleaned before further use. The report presents two classification models namely Naïve Bayes and Decision Trees and the classifier build using the models. The classifiers are evaluated using test data and performance measure are recorded. Feature selection and re-sampling are two important steps in improving performance metrics. At the end best results of both the classifier are described.

Data

Data Description

Data is collected from climate.weather.gc.ca. It is made publicly available by Environment Canada (EC) via RSS, url and direct download. To gather and pre-process the data, I am using the direct download option. The data is formatted as CSV (comma-separated values).

The data is available in 3 possible intervals: hourly, daily and monthly. I have chosen hourly interval data from **December 2012 to November 2014**. The total instances prior to cleaning were close to 19,000. The data has **24 attributes** namely:

Date/Time	Year	Month	Day	Time
Data Quality	Temp	Temp Flag	Dew Point Temp	Dew Point Temp Flag
Rel Hum	Rel Hum Flag	Wind Dir (10s deg)	Wind Dir Flag	Wind Spd (km/h)
Wind Spd Flag	Visibility (km)	Visibility Flag	Stn Press Flag	Hmdx
Hmdx Flag	Wind Chill	Wind Chill Flag		

The class labels are:

Tornado	Waterspout	Funnel Cloud	Thunderstorms	Heavy Thunderstorms
Rain*	Rain Showers*	Drizzle*	Freezing Rain*	Freezing Drizzle*
Snow*	Snow Grains*	Ice Crystals	Ice Pellets*	Ice Pellet Showers*
Snow Showers*	Snow Pellets*	Hail*	Fog	Ice Fog
Smoke	Haze	Blowing Snow	Blowing Sand	Blowing Dust
Dust	Freezing Fog	Virga		

The classes marked with an asterisk (*) can have one the three intensities: light, moderate and heavy. If such class does not have any intensity it is considered as light, otherwise as stated.

Data Cleaning

The CSV files for each month are combined using **Google Refine 2**. Google Refine 2 is very efficient tool to work with semi-structured and cluttered data. I am using the same for cleaning the data set.

The following steps are taken to ensure integrity and form of data set:

Step – 1: Inconsistent and missing values; several instances have class label 'NA' (nearly 40% of total) instances. The reason for such observation is due to the limitation of the instruments to determine the correct class label. Besides some instances have missing attribute values. All such instances are ignored.

Step – 2: Ambiguous class labels; class labels such as 'Rain, Snow' and 'Windy, Showers' are ignored due to presence of multiple labels. Moreover, such labels cannot be classified into any one of the class, and to accommodate such class, I will require to handle more classes.

Step – 3: Unnecessary details; each CSV file has legends and other information of the data. This information is not relevant for mining and analysis, hence removed.

After data cleaning and pre-processing the available instances for learning are **10,267**.

Algorithms

Naïve Bayes

To classify the data set I used Naïve Bayes Classifier. I choose Naïve Bayes due its simplicity, speed and the sound base of **Probability** theory. For weather data, NB classifier can prove to be very efficient as the attributes are tightly coupled with the weather condition. Initially, I trained the NB classifier which result in 64% accuracy. To enhance the accuracy, I applied **Chi-square test** (explained in later section) and eliminate the attributes that affects the classifier least. After applying feature selection method, the accuracy did not improve significantly. On observing the data closely, I found that the data was imbalanced. I performed well-known technique in statistics to smooth the distribution; called **resampling with replacement**. This technique essentially produces a random subsample of the dataset using sampling with replacement. After smoothing the imbalance distribution, the accuracy improved to 69%. To evaluate the performance, I used 10-fold cross validation with stratification which gave accuracy of 73%.

Decision Tree

Later I used **C4.5** decision tree to train classifier. Decision tree is one of the most important model in meteorology. C4.5 works on the basic concept of **Information Entropy**. The algorithm in essence selects nodes based on information gain; the attribute with highest information gain will be the root node of tree. Besides this, C4.5 creates pruned tree which essentially tries to remove branches that are not statistically significant and replace those branches with leaf nodes. It also handles missing values by excluding them from information gain calculations.

In the first trial, the accuracy was close to 68%. After applying feature selection technique namely '**Information Gain Ratio**' (explained in later section) and re-sampling the data to normalize the distribution, the accuracy was close to 80%. With 10-fold cross validation technique, the accuracy was as high as 85%.

Experimental Results

For both the classifiers the data was divided into two sets; **training set-66% and testing set-34%**.

Performance Measures

The following measures are used to judge the performance of the classifiers:

Precision – how well the classifier has performed in keeping the number of false positives less. Or in other words the number of instances selected are relevant.

Recall – the **sensitivity** measure to see how well the classifier reduces the false negatives. Or the number of relevant instances selected.

F-measure – the weighted harmonic mean of precision and recall.

ROC Area – the area under the curve (AUC) plotted on two dimensional graph for True Positive Rate (or **Sensitivity**) against False Positive Rate (or **1-Specificity**). To be precise ROC area is the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

Note: For the purpose of analysis, I choose four most occurring classes namely 'Mainly clear', 'Mostly cloudy', 'Cloudy', 'Fog' along with weighted average of all classes.

Performance Analysis

Performance of Naïve Bayes before feature selection and re-sampling:

Summary	
Correctly Classified Instances	2239 (64.1364 %)
Incorrectly Classified Instances	1252 (35.8636 %)
Total Number of Instances	3491

Detail Accuracy by Class							
		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Classes	Mainly Clear	0.562	0.072	0.619	0.562	0.589	0.891
	Mostly Cloudy	0.53	0.082	0.54	0.53	0.535	0.852
	Cloudy	0.592	0.093	0.532	0.592	0.56	0.881
	Fog	0.927	0.019	0.932	0.927	0.93	0.992
Weighted Avg. of all classes		0.641	0.051	0.645	0.641	0.64	0.921

Performance of Decision Tree before feature selection and re-sampling:

Summary	
Correctly Classified Instances	2377 (68.0894 %)
Incorrectly Classified Instances	1114 (31.9106 %)
Total Number of Instances	3491

Detail Accuracy by Class							
		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Classes	Mainly Clear	0.993	0.3	0.409	0.993	0.579	0.917
	Mostly Cloudy	0.384	0	1	0.384	0.555	0.904
	Cloudy	0.425	0.003	0.961	0.425	0.589	0.898

	Fog	0.975	0.018	0.94	0.975	0.957	0.988
Weighted Avg. of all classes		0.681	0.059	0.826	0.681	0.68	0.932

For Naïve Bayes, I used **Chi-square test** for independence which help me to test how fit each attribute is for the labels. **Merit and rank** determines the statistical relationship between the attributes and select features to improve the performance. However, the selection of right attributes that optimize the classifier is not possible.

As a feature selection technique for Decision Tree, I used the **Information Gain Ratio** ranking to test the entropy of each of the attributes. The technique helps decision tree to bias towards considering attributes with more number of distinct values. Hence the tree is more immune to problem of over fitting the data.

To smooth the imbalanced data, I used **re-sampling** with replacement for both the classifier which effectively reduce the inconsistencies in the distribution. Re-sampling yields better results due to the fact that it corrects the standard errors and confidence intervals.

Performance of Naïve Bayes after feature selection and re-sampling

Summary	
Correctly Classified Instances	2438 (69.8367 %)
Incorrectly Classified Instances	1053 (30.1633 %)
Total Number of Instances	3491

Detail Accuracy by Class							
		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Classes	Mainly Clear	0.722	0.089	0.631	0.722	0.673	0.912
	Mostly Cloudy	0.56	0.03	0.768	0.56	0.648	0.916
	Cloudy	0.559	0.063	0.584	0.559	0.571	0.888
	Fog	0.924	0.014	0.956	0.924	0.939	0.993
Weighted Avg. of all classes		0.698	0.039	0.716	0.698	0.701	0.94

Performance of Decision Tree after feature selection and re-sampling

Summary	
Correctly Classified Instances	2807 (80.4068 %)
Incorrectly Classified Instances	684 (19.5932 %)
Total Number of Instances	3491

Detail Accuracy by Class							
		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Classes	Mainly Clear	0.995	0.186	0.529	0.995	0.691	0.965
	Mostly Cloudy	0.636	0	1	0.636	0.777	0.965
	Cloudy	0.594	0	1	0.594	0.745	0.947

	Fog	0.983	0.006	0.982	0.983	0.983	0.992
Weighted Avg. of all classes		0.804	0.036	0.874	0.804	0.809	0.969

Even though computationally expensive, for experimental purpose I am using **10-fold cross validation with stratification** to evaluate the accuracy of both the classifiers. With stratified version of cross validation, the folds are selected such that the mean response value of each fold is same. The value K=0 is selected from its wide usage and acceptance. Hence K is 10 for this project. Besides larger values of K means lesser number of sample combination available. In our case, if K=10 and total instances equals 10267, the possible combinations will be nCr = 3.5 X 10^{33}, which is justified to validate the classifier.

Performance of Naïve Bayes with 10-fold cross validation

Summary	
Correctly Classified Instances	7543 (73.4684 %)
Incorrectly Classified Instances	2724 (26.5316 %)
Total Number of Instances	10267

Detail Accuracy by Class							
		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Classes	Mainly Clear	0.754	0.067	0.713	0.754	0.733	0.93
	Mostly Cloudy	0.666	0.043	0.728	0.666	0.696	0.917
	Cloudy	0.632	0.046	0.7	0.632	0.664	0.916
	Fog	0.932	0.015	0.95	0.932	0.941	0.993
Weighted Avg. of all classes		0.735	0.035	0.745	0.735	0.736	0.949

Performance of Decision Tree with 10-fold cross validation

Summary	
Correctly Classified Instances	8817 (85.8771 %)
Incorrectly Classified Instances	1450 (14.1229 %)
Total Number of Instances	10267

Detail Accuracy by Class							
		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Classes	Mainly Clear	0.996	0.134	0.621	0.996	0.765	0.983
	Mostly Cloudy	0.738	0	0.999	0.738	0.849	0.98
	Cloudy	0.733	0.001	0.989	0.733	0.842	0.978
	Fog	0.986	0.004	0.987	0.986	0.986	0.991
Weighted Avg. of all classes		0.859	0.027	0.899	0.859	0.863	0.984

Conclusion

The project as a whole helped me understand Naïve Bayes and Decision Tree in depth. After going through the results I understood that Decision tree outperforms Naïve Bayes in time-series analysis.

Apart from the classification technique, I learned gathering, cleaning and mining big data problems. If more time was available, I would like to include the instances with multiple classes, which can be used to train a model. I would like to **forecast** the weather using better machine learning algorithms. I would like to try neural network algorithms which is a buzz word in field of meteorology. The **Self-Organizing Map** is very useful neural network model and it is ideal for high dimensional data visualization and modeling.