COMPX521: Assignment 2

Connor Welham

June 16, 2022

1 Introduction

Random Ferns are a form of semi-naive Bayesian classifier which uses a simple generative model for each class. It uses Bayes' theorem to estimate the class probabilities for each class model. However, Random Ferns has one significant difference from Naive Bayes in that the features are not assumed to be conditionally independent.

This classifier is suggested in a 2010 paper [1] which explains the methodology behind Random Ferns. The paper talks about using Random Ferns for object detection and aiming to reduce the computational complexity in handling perspective distortion.

The aims of this project is to implement Random Ferns in WEKA [2] and compare the classification accuracy of the classifier to other algorithms to evaluate how well Random Ferns does perform.

2 Method

The method being used in my implementation of Random Ferns is based on the 2010 paper [1] as follows.

The set of N predictors are randomly shuffled before it is split by a specific size S. This gives M disjoint subsets of attributes. Only the predictor attributes in these subsets are assumed to be conditionally independent, unlike Naive Bayes which assumes all predictor attributes are conditionally independent from each other.

For each subset M from 1 to k, Random Ferns estimates a joint probability distribution table from the training data. This is then stored in a hash table which does not included and estimates with a value of 0. The estimates are then adjusted using the Laplace estimator to avoid the zero-frequency problem.

A hash table will be created for each class value. In these hash tables, each combination of attributes, according to the subsets will be recorded as an instance along with its count. If a combination does not occur in the training data, it will not be recorded, which saves a lot of time and computational complexity. The hash table keys will be formatted:

$(attributeValue_1)/(attributeIndex_1)--(attributeValue_2)/(attributeIndex_2)--...$

$$--(attributeValue_s)/(attributeIndex_s)$$

where the order comes from the order of indices in the subset generated, with number of attributes in the subset is s.

By assuming conditional independence for each subset F_k, we can write the conditional probability of observing an instance X given class c_i as:

$$P(X|C = c_i) = \Pi_{k=1}^{M} P(F_k|C = c_i)$$

This is where $P(F_k|C = c_i)$ is the class conditional joint probability distribution for the kth subset of attributes for class c_i.

By using the estimates of $P(X|C = c_i)$ and Bayes' Theorem using the class prior probabilities $P(C = c_i)$ we can calculate the posterior probabilities $P(C|X = x)$ for a given instance x. However, the probability estimates here are dependent on how the attribute subsets are shuffled. To eliminate this dependency, Random Ferns needs to be run multiple times by training multiple Random Ferns models and averaging the class probabilities. This will be done using Random Committee and setting different numbers of ensemble members. We will also aim to test various subset sizes in order to eliminate a reduced accuracy for a poor subset size and to also test which is generally the best subset size to use.

To check the accuracy of Random Ferns, it will need to be compared to other algorithms that it does a similar job to, such as Naive Bayes and Random Forest. The baseline algorithm will be Random Committee with 10 ensemble members with Random Ferns of subset size 2. This is done so that we can compare not only Naive Bayes and Random Forest with Random Ferns, but also compare small ensemble members to large and small subset size to large.

To gather the experiment results, we will use WEKA's [2] experimenter to apply 10 runs of cross-validation on each of the algorithms to estimate predictive performance in terms of the percentage of correct classification on each of the available datasets. Close attention will be payed to statistical difference at 0.05 significance using a paired t-test between two algorithms in order to evaluate if there is a statistically significant difference between accuracy of algoriothms.

3 Experimental results

To conduct this experiment, Random Ferns was implemented with Random Committee and compared to Naive Bayes and Random Forest by testing on different datasets. The experiment was run using WEKA [2] to estimate predictive performance in terms of the percentage of correct classification(s). Each test will be done with different subset sizes of 2, 3 and 4 and different ensemble members of 10, 50 and 100.

The results of the experiment are shown in table 1.

Table 1: Percentage of Correct Classifications

Dataset	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
balance-scale	90.50	90.53	90.58	87.95 •	88.22 •	88.27 •	45.76 •	45.76 •	45.76 •	91.44	79.23 •
car	88.78	89.24	89.21	89.78	90.48 ◦	90.63 ◦	89.25	89.94	90.09	85.46 •	94.67 ◦
hayes-roth	70.83	71.23	71.37	70.38	72.13	72.88	55.69 •	55.69 •	55.69 •	82.66 ◦	77.74 ◦
lymphography	85.83	86.18	86.58	85.10	85.50	85.38	83.50	84.58	84.65	85.10	82.29
nursery	91.55	91.63	91.66	92.57 ◦	92.74 ◦	92.74 ◦	94.23 ◦	94.44 ◦	94.47 ◦	90.30 •	99.06 ◦
promoters	89.06	90.42	90.05	84.45	85.47	85.53	82.45	84.20	84.87	90.14	91.83
solar-flare-C-class	80.61	80.59	80.68	81.95 ◦	81.93 ◦	81.92 ◦	82.91 ◦	82.96 ◦	82.98 ◦	78.86 •	81.80
solar-flare-M-class	92.65	92.63	92.61	93.46 ◦	93.59 ◦	93.55 ◦	94.18 ◦	94.17 ◦	94.23 ◦	90.87 •	94.10 ◦
solar-flare-X-class	97.17	97.21	97.23	98.01 ◦	98.04 ◦	98.06 ◦	98.78 ◦	98.80 ◦	98.79 ◦	96.03 •	98.86 ◦
spect	79.43	79.77	79.58	80.41	80.52	80.41	81.61	81.88	81.80	78.68	81.81
splice	95.53	95.50	95.53	95.23	95.39	95.44	93.83 •	94.35 •	94.49 •	95.42	95.88

◦, • statistically significant improvement or degradation

(1) meta.RandomCommittee '-S 1 -num-slots 1 -I 10 -W trees.RandomFerns – -S 1 -size 2'
(2) meta.RandomCommittee '-S 1 -num-slots 1 -I 50 -W trees.RandomFerns – -S 1 -size 2'
(3) meta.RandomCommittee '-S 1 -num-slots 1 -I 100 -W trees.RandomFerns – -S 1 -size 2'
(4) meta.RandomCommittee '-S 1 -num-slots 1 -I 10 -W trees.RandomFerns – -S 1 -size 3'
(5) meta.RandomCommittee '-S 1 -num-slots 1 -I 50 -W trees.RandomFerns – -S 1 -size 3'
(6) meta.RandomCommittee '-S 1 -num-slots 1 -I 100 -W trees.RandomFerns – -S 1 -size 3'
(7) meta.RandomCommittee '-S 1 -num-slots 1 -I 10 -W trees.RandomFerns – -S 1 -size 4'
(8) meta.RandomCommittee '-S 1 -num-slots 1 -I 50 -W trees.RandomFerns – -S 1 -size 4'
(9) meta.RandomCommittee '-S 1 -num-slots 1 -I 100 -W trees.RandomFerns – -S 1 -size 4'
(10) bayes.NaiveBayes "
(11) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1'

4 Evaluation

Table 1 shows the prediction accuracy for each algorithm.

Looking at algorithms 1 through 9, which are Random Ferns implemented with Random Committee of various sizes and ensemble members, we can see that it seems to only show a significant improvement or degradation in respect to subset size. There is no significant improvement or degradation between different ensemble members. For datasets such as balance-scale, hayes-roth and splice, it is shown that there is a statistically significant degradation using a higher subset size (3/4). This is most likely due to the nature of the data that works better with a smaller subset size. Other datasets worked better with a larger subset size, such as car, which worked best with a size of 3, or the solar-flare datasets which had a significant improvement for subset sizes 3 and 4, irrespective of the ensemble members.

Looking at algorithm 10, Naive Bayes, we can see that for most datasets the algorithm had a statistically significant degradation when compared to the baseline Random Ferns. This suggests that the Random Ferns had a higher percentage of correct classifications. However, looking at algorithm 11, Random Forest, the opposite could be said as for most datasets there was a significant improvement in accuracy. There are outliers for both of these observations such as hayes-roth which was a significant improvement under Naive Bayes or balance-scale which was a significant degradation under Random Forest.

5 Conclusions

To conclude, there appears to be no statistically significant improvement or degradation when changing the number of ensemble members of a Random Committee using Random Ferns, which suggests that our baseline of 10 members is sufficient for the tested datasets. There does appear to be a significant improvement/degradation for some datasets when increasing the subset size, which suggests that subset size needs to be chosen by analyzing the data before creating a model for maximum accuracy. Subset size should be chosen by the number of predictor variables in the data. Finally, it appears that Random Ferns is a significant improvement over Naive Bayes in terms of the percentage of correct classification for most algorithms. However, Random Ferns is a significant degradation over Random Forest in terms of the percentage of correct classification for most algorithms.

To further explore this, it might be worthwhile testing Naive Bayes and Random Forest against the best performing Random Ferns of a suitable subset size and ensemble members to confirm this conclusion. However, I was unable to check this due to the computational power of my laptop as well as the deadline of this assignment.

References

[1] Lepetit V Fua P Ozuysal M, Calonder M. *Fast keypoint recognition using random ferns.* IEEE Trans Pattern Anal Mach Intell, 2010.

[2] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, Burlington, MA, 3 edition, 2011.