A GREEDY PERFORMANCE DRIVEN ALGORITHM FOR DECISION FUSION LEARNING

Dhiraj Joshi

Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802

Milind Naphade, Apostol Natsev

IBM Thomas J. Watson Research Center
Hawthorne, NY 10532

ABSTRACT

We propose a greedy performance driven algorithm for learning how to fuse across multiple classification and search systems. We assume a scenario when many such systems need to be fused to generate the final ranking. The algorithm is inspired from *Ensemble Learning* [2] but takes that idea further for improving generalization capability. Fusion learning is applied to leverage text, visual and model based modalities for 2005 TRECVID query retrieval task. Experiments using the well established retrieval effectiveness measure of mean average precision reveal that our proposed algorithm improves over naive baseline (fusion with equal weights) as well as over Caruana's original algorithm (NACHOS) by 36 % and 46 % respectively.

Index Terms— TRECVID, mean average precision, late fusion, hill climbing

1. INTRODUCTION

In most machine learning, classification and search tasks, it is often the case that consulting multiple algorithms, or systems and somehow combining their outputs (whether they are ranked lists or confidences) always tends to perform better than even the best individual algorithm or system. Ample evidence of this exists in benchmarks [1]. Fusing such outputs with little or no knowledge of the individual classifiers or search engines is a very productive approach for leveraging an ensemble of such systems or algorithms. Such late stage combination of decisions is often termed late fusion or decision fusion and generally gives robust improvements in several domains including the video classification and retrieval domain [7]. When the individual systems whose output decisions are being fused have nearly comparable performances, naive strategies such as simply averaging their decisions perform adequately and beat several supposedly smarter fusion approaches [1]. Here we assume that given a set of entities that need to be classified or ranked, all individual systems and/or algorithms are able to process these entities and provide to the late fusion algorithm a ranking or confidence where higher values indicate that the document is being found relevant to the query and ranked towards the top and lower values indicate that the document is not considered relevant and is being pushed to the bottom of the ranked list. The goal is to then learn a strategy which leverages all available decision inputs optimally. Since fusion learning is usually performed using a hill-climb set, the learned fusion strategy could easily overfit the data used. Hence, avoiding overfitting poses a major challenge.

Classifier combination methods have been studied for a long time. A classic treatise on combination of classifiers can be found in [4]. The cited work presents a statistical framework which encompasses many existing methods of compound classifier combinations. In [5], a theoretical study of certain basic classifier combination strategies has been performed. Methods combining classifiers based on their performance in an unknown test sample's local neighborhood have been proposed [11]. Boosting methods which produce complex composite hypothesis using multiple weak classifiers are very popular [3, 9]. In [2], Caruana et. al. proposed an ensemble learning algorithm, NACHOS, which performs a greedy forward selection on a hillclimb set and learns weights across multiple classifiers. We applied NACHOS to the TRECVID 2005 search task [10] with the aim of fusing three independently designed search sub-systems, one based on text retrieval, another on visual similarity based retrieval and a third based on detecting relevant semantic concepts in the query videos and the target data set. Our experiments revealed that NACHOS [2] performed worse than naive fusion that combines these three sub-systems with equal weights. Fusion with equal weights is in general shown to perform reasonably well when all individual decision streams exhibit performance in the same ball park [1]. In our quest, we designed a novel algorithm for ensemble fusion that was still in the same spirit as the NACHOS algorithm but was robust and had the ability to generalize better. Experiments using the mean average precision measure revealed that our technique improves over the baseline by 36 % and over NACHOS by 46 % on the TRECVID 2005 query retrieval task.

This work was performed when Dhiraj Joshi was a summer intern at the IBM T. J. Watson Research Center. The email contacts of the authors are djoshi@cse.psu.edu, naphade@us.ibm.com, and natsev@us.ibm.com.

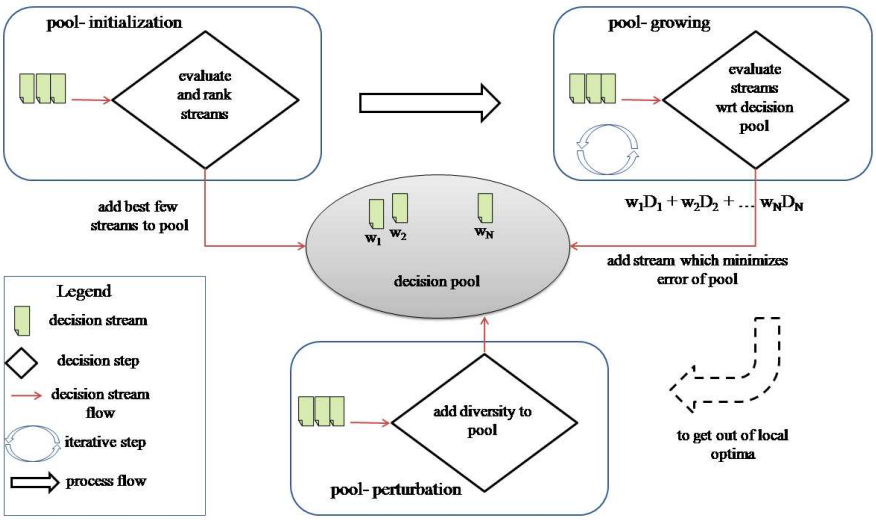

Fig. 1. A graphical representation of the component processes in the fusion learning algorithm.

2. DECISION FUSION LEARNING ALGORITHM

At its core, our fusion learning algorithm involves learning a linear combination over decision streams where the weights learnt for particular decision streams are representative of their authority in making a decision in a particular scenario. As is obvious, the weights learnt or authority of making decision could differ from problem, decision criteria, and the hillclimb data used for judging decision validities. Alternatively, the problem of learning a linear combination can be rephrased as one of search for an optimal pool of decision streams for a given task where decision streams are included into the pool with replacement. In the end, the proportion of different streams present is equivalent to their weights. As shown in Figure 1, the three most important steps involved are 1). *Pool Initialization* - preferably beginning with the best decision makers, 2). *Pool Growing* - an inclusion criteria which enhances the overall performance of the pool, and 3). *Pool Perturbation* - a technique to jump out of local optimas by adding diversity to the pool. Several other ideas were brought into our greedy performance driven ensemble learning algorithm including utilization of history for rollback during a hillclimb search. Usually some kind of resampling of data is considered beneficial during learning. Our learning algorithm is compounded with bagging and cross-validation-based resampling to improve model generalization and robustness. The detailed steps and their individual motivations are outlined next.

1. The pool is initialized with the top K decision streams each assigned a pre-determined weight. At this step, the best decision stream could be triggered with a higher weight. This can be useful in the presence of some a-priori confidence in the best decision stream. If all streams are believed to be equally authoritative for the problem, it is a good strategy to begin with equal weights. The parameter K can be varied to control performance and avoid overfitting.

2. Each decision stream is evaluated against some pre-decided error metric for the problem and the stream which minimizes the error of the pool is added to the pool with a weight. However, in order to be included, a decision stream must decrease the error of the pool by a certain percentage (usually set between 4% and 7%). This wards against overfitting.

3. If addition of no decision stream helps decrease the error of the pool, the algorithm allows for two strategies to perturb the solution and allow for more hillclimbing.

   - The weight of the highest-weight stream can be increased. This scheme is believed to be beneficial when one decision stream is apriori believed to be more authoritative than the rest.
   - A random decision stream can be included to add diversity to the pool. This scheme is found to be a good way to get out of local optimas.

4. Steps 2 and 3 are repeated for a fixed number of iterations or till convergence. Additionally, we maintain history about the best set of decision stream weights obtained during the hillclimb. The algorithm rolls back to these weights in the end.

In order to ensure generalization of decision weights to different and potentially larger sets of data, we allow for the following kind of resampling methods to be used with fusion learning. (1) Bagging: Only a proportion of data points in

the hillclimb set are used to perform decision fusion learning. This is repeated for a fixed number of iterations and the final weights are obtained as the average of weights in each run. (2) Cross-Validation: In a fashion similar to bagging, we split the hillclimb set into training and test partitions. The algorithm is trained on the training partition and fusion weights obtained are tested on the test partition. This is repeated for a fixed number of iterations and fusion weights which give the least cross-validation error form the final solution. The algorithm parameters used in the current experiments are mentioned in Section 3.

The error calculation step determines the complexity of the algorithm as it is performed for each decision stream at each iteration. This involves performing a sort on the dataset. Suppose the size of the hillclimb set is denoted as V and the number of decision streams is N, the computational complexity of each iteration of the algorithm is $\mathcal{O}\ (NV \log V)$. If the number of iterations is T, the complexity becomes $\mathcal{O}\ (TNV \log V)$. An alternative to the proposed approach could be to perform an exhaustive search for fusion weights. Assuming that there are K decision streams, as earlier, and one allows each stream to take integer weights between 0 and $M-1$, this approach would involve searching M^N weight combinations for a hillclimb set of size V. The complexity for this would come out to be $\mathcal{O}\ (M^N V \log V)$. Moreover, this approach is not scalable when the number of decision streams N is very large. Hence the proposed approach is a *polynomial time* greedy search as opposed to *exponential time* global grid search.

3. EXPERIMENTS - FUSING TEXT, VISUAL AND MODEL BASED MODALITIES

Our experiments focus around a practical problem in the TRECVID search task. TRECVID [10] is an annual semantic video retrieval benchmark run by the National Institute of Standards and Technology (NIST) in which a common corpus and a common set of queries are made available to participants every summer. Participants then analyze the corpus and the queries and try to answer the queries by ranking the video shots in the test corpus based on their relevance to the queries. Such ranking systems leverage text retrieval (as the video documents provided in the test corpus are accompanied by automatic speech recognition transcripts), and visual similarity based retrieval (as the queries contain a textual description as well as exemplary images and videos). We have also pioneered [6, 7] the approach of detecting a large number of semantic concepts such as objects, locations and events in videos and then using those to retrieve video shots that are relevant to the query text and the concepts in the accompanying query images and videos

The dataset of 80 hours of broadcast news video comprising 45765 shots was split into a hillclimb set of 21175 shots and test set of 24590 shots. Care was taken to ensure that all shots from a particular video end up in only one of those two sets without being split across the training and testing set. The fusion algorithm was trained on the hillclimb set and tested on the test set. For each query we had the following set of outputs, 1) T - retrieval using text transcripts obtained with the IBM TRECVID text retrieval system, 2) V - visual retrieval obtained with the IBM TRECVID visual similarity retrieval system, and 3) M - model based retrieval using the IBM semantic concept detection and retrieval system [8]. For details of these individual retrieval sub-systems please refer to [1]. To measure retrieval effectiveness, we used the standard NIST measure of non-interpolated average precision for each query and the mean average precision for an overall evaluation of the system. Let R be the number of true relevant documents in a set of size S; L the ranked list of documents returned. At any given index j let R_j be the number of relevant documents in the top j documents. Let $I_j = 1$ if the j^{th} document is relevant and 0 otherwise. Assuming $R < S$, the non-interpolated average precision **(AP)** is then defined as

$$\frac{1}{R} \sum_{j=1}^{S} \frac{R_j}{j} * I_j \tag{1}$$

The goal was to learn weights for individual streams T, V, and M with an intent to improve overall retrieval performance. Weights were learnt for each query independently and the error-criteria used in the stream selection step was average precision at a depth of 1000. The initial parameter K as in step 1 in Section 2 was set as 1. Since we did not have an apriori knowledge about the authoritativeness of any of the three decision streams, a random perturbation of the pool was used at step 3, as described in Section 2. The percentage used as inclusion criteria in step 2 was set as 5%. Experiments were performed using both bagging and cross-validation-based resampling methods. In Table 1, we present the fusion results. The numbers presented are average precision at a depth 1000 for the test set. For the fusion, we present the bagged-average and best cross validated results across 10 runs respectively. We also show performance of individual T, V, and M streams for each query. The two baselines shown are the naive fusion across T, V, and M decision streams and performance using NACHOS algorithm. Performance of the *oracle selection*- the best classifier taken across each query independently is also shown. As is evident from the results, both $Fusion - bag$ and $Fusion - CV$ beat the baselines, the oracle and the individual streams. The percentage gain in performance with $Fusion - bag$ and $Fusion - CV$ over baseline fusion is 36% and 26% respectively. The lower overall performance of $Fusion - CV$ could perhaps be attributed to overfitting. Moreover, $Fusion - bag$ performs 46% better than NACHOS baseline.

$Query$	V	M	T	$Oracle$	$Naive$	$NACHOS$	$Fusion-Bag$	$Fusion-CV$
149	0.0011	0.0002	0.0796	0.0796	0.1382	0.0593	0.1266	0.1251
150	0.0004	0.001	0.0331	0.0331	0.2682	0.0331	0.1281	0.0004
151	0.2225	0.0754	0.346	0.346	0.5057	0.4881	0.5041	0.5015
152	0.084	0.0097	0.0284	0.084	0.0633	0.0981	0.0938	0.0797
153	0.0259	0.0041	0.4029	0.4029	0.4156	0.401	0.4191	0.4194
154	0.0043	0.0233	0.1108	0.1108	0.1512	0.1512	0.1488	0.1282
155	0.2728	0.008	0.00	0.2728	0.1605	0.008	0.2861	0.2728
156	0.9044	0.0433	0.128	0.9044	0.2737	0.2737	0.8657	0.7076
157	0.0178	0.0082	0.0022	0.0178	0.0136	0.0202	0.0196	0.0204
158	0.0461	0.0258	0.0516	0.0516	0.1428	0.1082	0.1526	0.1538
159	0.004	0.0001	0.0009	0.004	0.004	0.004	0.0047	0.0037
160	0.0227	0.0075	0.00	0.0227	0.0029	0.0021	0.0078	0.0046
161	0.0552	0.0922	0.0165	0.0922	0.0646	0.093	0.1007	0.0763
162	0.031	0.0074	0.0019	0.031	0.0074	0.0316	0.0327	0.0292
163	0.0172	0.0129	0.0067	0.0172	0.0215	0.0282	0.0279	0.0201
164	0.2028	0.222	0.1654	0.222	0.3522	0.3098	0.3493	0.3485
165	0.365	0.0434	0.0572	0.365	0.1488	0.2598	0.3765	0.3584
166	0.0468	0.0044	0.0045	0.0468	0.0313	0.0568	0.0514	0.0551
167	0.0361	0.0102	0.00	0.0361	0.0062	0.0361	0.0369	0.0361
168	0.102	0.2439	0.041	0.2439	0.1331	0.259	0.2531	0.2629
169	0.087	0.0414	0.0544	0.087	0.1285	0.1396	0.1423	0.135
170	0.0431	0.0664	0.0006	0.0664	0.0223	0.0746	0.0749	0.0771
171	0.5032	0.249	0.1651	0.5032	0.4431	0.249	0.5206	0.5510
172	0.1386	0.0352	0.0128	0.1386	0.0641	0.1133	0.1382	0.1391
MAP@1000	0.1348	0.0515	0.0712	0.1741	0.1483	0.1374	0.2026	0.1878

Table 1. Compare the performance of the proposed approach on the 24 queries from TRECVID 2005 search task. The two baselines are naive fusion across T, V, and M streams and performance with NACHOS algorithm.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed and described a greedy performance driven algorithm for learning how to perform decision fusion across multiple classification and search systems. The algorithm is inspired from the *Ensemble Learning* idea proposed by Caruana et.al. [2] but has added features in order to make this learning procedure robust with improved generalization capability. The learning algorithm is compounded with bagging and cross-validation-based resampling to improve model generalization and robustness. We applied the fusion learning to leverage text, visual and model based modalities for 2005 TRECVID query retrieval task. Experiments using the well established retrieval effectiveness measure of mean average precision reveal that our proposed algorithm improves over the naive fusion baseline as well as over Caruana's original algorithm, NACHOS by 36 % and 46 % respectively. Future directions include applying this algorithm to other TRECVID tasks such as decision fusion for semantic concept detection, as well as pseudo-relevance feedback for further improving performance of the search task.

5. REFERENCES

[1] A. Amir, J. Argillander, A. Haubold, F. Kang, M. Naphade, A. Natsev, J. Smith, and J. Tesic, "IBM Research TRECVID-2005 Video Retrieval System," Gaithersburg, 2005.

[2] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble Selection from Library of Models," Proc. Int. Conference on Machine Learning, 2004.

[3] Y. Freund, and R. E. Schapire, "Experiments with a New Boosting Algorithm," Proc. Int. Conference on Machine Learning, 1996.

[4] J. Kitler, M. Hatef, R. P. W. Duin, and J. Matas, "On Combining Classifiers," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226–239, 1998.

[5] L. I. Kuncheva, "A Theoretical Study on Six Classifier Fusion Strategies," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 40, no. 2, pp. 281–286, 2002.

[6] M. R. Naphade, I. Kozintsev, and T. S. Huang, "A Factor Graph Framework for Semantic Video Indexing", IEEE Trans. on Circuits and Systems for Video Technology, vol. 12, no. 1, pp. 40–52, 2002.

[7] M. Naphade, J. Smith, and F. Souvannavong, "On the Semantic Detection of Concepts at TRECVID," Proc. ACM Multimedia, 2004.

[8] A Natsev, M Naphade, and J Smith, "Semantic Representation Searching and Mining of Multimedia Content," Proc. Knowledge Discovery and Data Mining, 2004

[9] J. O. Sullivan, J. Langford, R. Caruana, and A. Blum, "FeatureBoost: A MetaLearning Algorithm that Improves Model Robustness," Proc. Int. Conference on Machine Learning, 2000.

[10] TREC
Video Retrieval, "National Institute of Standards and Technology," `http://www-nlpir.nist.gov/projects/t01v`, 2005.

[11] K. Woods, W. P. Kegelmeyer, and K. Bower, "Combination of Multiple Classifiers Using Local Accuracy Estimates," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 4, pp. 405–410, 1997.