A NOVEL RANKING METHOD FOR MULTIPLE CLASSIFIER SYSTEMS

Anurag Kumar and Bhiksha Raj

Carnegie Mellon University, Pittsburgh,PA- 15213,USA

ABSTRACT

We introduce an unsupervised optimization method for optimal fusion of multiple classifiers in retrieval problems. The method is based on a ranking loss called the "clarity" index, which does not depend on the label of the test instances. The technique optimizes the weights with which individual classifier scores must be combined to maximize this clarity. Our method is *instance-specific*; the weights are optimized individually for each test instance. The proposed schema can also be used for instance-specific ranking of classifiers. We also show that the method is highly tolerant to the introduction of noise in classifier outputs.

Index Terms— Ranking, Retrieval, Classifier Fusion

1. INTRODUCTION

Classification of multimedia data is frequently cast as a *retrieval* problem, *i.e.* that of generating a rank ordering of the entries in a corpus, such that positive instances of the target class in the corpus are ranked ahead of negative instances [1]. Ranking is by performed by assigning a numeric "membership score" to the instances, by which they are subsequently sorted.

The score used to rank the instances is often derived from an ensemble of classifiers. Multimedia recordings comprise multiple sources of evidence – video, audio, various forms of embedded text, meta data, etc., all of which potentially carry information about their class membership. Each of the classifiers in the ensemble classifies the instances in the corpus using one or more of these sources of evidence, and generates a confidence score for each instance. This score may be viewed as an estimate of the probability of membership to the class that is assigned to the instance by the classifier. The scores generated by the individual classifiers must now somehow be combined to derive an overall membership score to rank the instances.

The most common method is simply to compute the membership score as a weighted average of the scores produced by the individual classifiers [2]. This "classifier fusion" approach has been found to be surprisingly effective, and even a simple *average* weighting scheme, which assigns equal weights to all classifiers, can result in remarkably robust performance that can be hard to beat. A number of methods have been proposed in the literature to arrive at superior estimates of the weights assigned to each classifier [2] [3] [4] [5]. These schemes, however, make a common assumption: they assume that improved rank ordering can be achieved by improving classification accuracy. Thus, the optimization of the weights assigned to the classifiers in the ensemble is generally performed to maximize expected classification accuracy.

This assumption is largely valid, in that a perfect classifier would rank order all positive instances ahead of all negative instances, however it is not entirely true. From the perspective of optimizing classification accuracy, a misclassified positive instance that lies at a given distance from the decision boundary on the wrong side and scores below only a small fraction of negative instances is no different from one that lies at the same distance, but scores worse than all of the negative instances. In a ranking scenario, on the other hand, the first case, where the positive instance is outscored by only a small fraction of negative instances, is greatly preferable to the latter. Thus, it is not sufficient to only consider the distance of positive instances from the decision boundary when optimizing the weights, one must also consider their ranking relative to the negative instances.

In this paper we propose an alternate mechanism for computing membership scores from ensembles of classifiers, that explicitly optimizes an objective related to the expected rank of the instances. Specifically, we define the match of any instance to the class through a *bipartite ranking loss* [6][7], which we term *clarity*. The weights are optimized to minimize this loss, and the corresponding loss is used as the membership score of the instance.

In addition to explicitly considering the optimal ranking as a criterion to assign class scores, the proposed method has some additional benefits. One direct benefit is in terms of supervision. Conventional mechanisms for learning fusion weights are usually supervised: the weights are learned to optimize the classification accuracy of a validation set for which true labels are known. Weights learned on validation sets are not guaranteed to also be effective for actual test data [8]. Our proposed method, on the other hand, is *unsupervised*; the computation of clarity does not require ground-truth labels, and we optimize the weights directly on the test data in the corpus. A second benefit lies in the specificity of the weights to the instances. Once learned from validation data, conventional fusion methods apply the same set of weights to all test instances in the corpus, ignoring the variations between the test instances. Our approach optimizes the weights individually for every test instance.

The latter feature – the fact that we learn instance-specific weights, also lends itself to a third unexpected benefit, which we also evaluate in this paper: it automatically provides us with a mechanism for *classifier selection*.

Experiments on a multimedia retrieval task show that the proposed membership scoring method can result in better retrieval, as measured by standard retrieval metrics. It is also able to effectively identify and deselect noisy classifers when employed for classifier selection.

The rest of the paper is organized as follows: Section 2 briefly discusses related work, Section 3 explains our proposed algorithm and Section 4 shows our experimental results. We present our conclusions in Section 5.

2. RELATED WORK

Approaches to classifier combination in multi-classifier systems broadly fall into two categories. The first combines the *scores* output by the individual classifiers, while the second aggregates *ranks* of instances from individual classifiers. Both of these approaches have been studied in several works such as [9] [10] [11] [12] [13].

Of the two the more common approach is that of combining *scores*, although the combined score itself may be used to obtain a final ranking. A generic formalism for combining classifier scores is Stacking [14]. Stacking involves the idea of actually *learning* to combine classifiers optimally, rather than simply taking votes or the average of classifier outputs. The outputs of individual classifiers are treated as inputs to a next classification stage which is in turn optimized for the best overall classification [5] [15] [16] [17]. However, stacking methods which use the outputs of individual classifiers as features for a next-level algorithm such as a support vector machine do not explicitly consider *ranking* as the final objective, a desirable aspect in retrieval situations.

On the other hand, rank aggregation methods require individual models to be reasonable; otherwise there could be large variations in the rankings produced by the individual models resulting in meaningless aggregated rank.

This suggests that both the scores and the relative position of an instance with respect to other instances, *i.e.* its rank, must be considered in the final combination strategy. In our work we achieve this through the *clarity index* which we optimize. Another important constraint of methods that rely on held-out or training data for learning how to combine classifiers is that the learned weights may not generalize to actual test instances[8]. Again our method solves this by optimizing on the actual test data itself, resulting in no generalization constraints.

3. PROPOSED ALGORITHM

3.1. Problem Setting

Consider a retrieval problem in which the task is to rank order the instances in a set Y by their membership to some target class. To assist us, we possess a set of m classifiers C_i, $i = 1 \cdots m$. To any sample instance p, each classifier C_i assigns a score $x_i(p) = C_i(p)$ representing the confidence with which C_i assigns p to the target class. We aim to estimate a set of weights $w_i(p)$ such that a weighted combination $s(p, \{w_i(p), i = 1 \cdots m\}) = \sum_{i=1}^{m} w_i(p)x_i(p)$ can be used to optimally rank order the instances in Y. For brevity of notation, we will represent the set of m scores assigned to any instance as an m-dimensional score vector $\vec{x}(p) = [x_1(p)\, x_2(p)\, x_3(p) \cdots x_m(p)]^T$. We will represent any set of weights $w_i, i = 1 \cdots m$ as a weights vector $\vec{w} = [w_1\, w_2\, w_3 \cdots w_m]^T$. Using this notation, $s(p, \vec{w}) = \vec{w}^T\vec{x}(p)$. For further brevity, we also drop the "(p)" in our notation. $\vec{x}$ will now represent both the instances and their score vector.

We will also assume the availability of a set of labeled training instances X that can be used to help us estimate $\vec{w}$. X may or may not be same as the actual training data used to train the m classifiers. In this paper, we set them to be the same. We separate X into X_+ representing the set of *positive* instances in X, *i.e.* the instances that belong to the target class and X_- representing the set of *negative* instances, which do not belong to the target class.

3.2. Relevance, Irrelevance and Clarity Index

Our objective is to compute a *membership* score $r(\vec{x})$, such that ordering the instances in Y by $r(\vec{x})$ may be expected to result in accurate retrieval of Y, such that most positive instances rank higher than most negative instances. We define the weighted combination $s(\vec{x}, \vec{w}) = \vec{w}^T\vec{x}$ obtained with a weight vector $\vec{w}$ as the fused *confidence* score of that instance. Most classifier fusion techniques for retrieval optimize $\vec{w}$ and simply set $r(\vec{x}) = s(\vec{x}, \vec{w})$; however we will keep the two separate and compute $r(\vec{x})$ *from* $s(\vec{x}, \vec{w})$. We will optimize $\vec{w}$ such that $s(\vec{x}, \vec{w})$ maximizes a *clarity* index, which we define below, which relates to the ranking of the instances. $r(\vec{x})$ will subsequently be set to the clarity value itself.

For any instance $\vec{x}$ it is expected that the optimally weighted score $\vec{w}^T\vec{x}$ will be high if the instance is actually positive and low if it is negative. This can be understood by contrasting the test instance $\vec{x}$ to the labeled training instances in X. The weight vector $\vec{w}$ should be chosen such that the weighted score should outscore as many positives in $X+$ as possible if $\vec{x}$ is actually a positive instance. On the other hand, if $\vec{x}$ is actually negative the weight vector should put $\vec{x}$ below as many negatives as possible. This intuition is formalized mathematically as follows.

We define two losses, the *relevance loss* and the *irrelevance loss*, and an index based on these losses [7]. The *relevance loss* $RL(\vec{x}, \vec{w})$ for an unlabeled point p^u with score vector $\vec{x}$ and weight vector $\vec{w}$ is defined as the fraction of negatively labeled training instances from X_- that score *more* than $\vec{x}$, when the scores are combined using $\vec{w}$:

$$RL(\vec{x}, \vec{w}) = \frac{1}{|X_-|} \sum_{x_- \in X_-} I\left(\vec{w}^T\vec{x}_- > \vec{w}^T\vec{x}\right) \tag{1}$$

where $I(z)$ is an indicator function that takes the value 1 if its Boolean argument z is true, and 0 otherwise.

Similarly, the *irrelevance loss* $IL(\vec{x}, \vec{w})$ is defined as the fraction of positively labeled training instances from X_+ that score *less* than $\vec{x}$, when $\vec{w}$ is used as the weight vector.

$$IL(\vec{x}, \vec{w}) = \frac{1}{|X_+|} \sum_{x_+ \in X_+} I\left(\vec{w}^T\vec{x} > \vec{w}^T\vec{x}_+\right) \tag{2}$$

If the unlabeled instance $\vec{x}$ is actually a positive instance, it is desired that its *relevance loss* be low (0 in the ideal case). Also, the higher the *irrelevance loss* the more certain we are that $\vec{x}$ is positive. However, if $\vec{x}$ is actually a negative instance, then the *irrelevance loss* should be low, and the *relevance loss* must be high. These two factors can be combined into a single index termed as *Clarity Index*, which we define as the absolute value of the difference between the *relevance loss* and *irrelevance loss*.

$$CL(\vec{x}, \vec{w}) = |RL(\vec{x}, \vec{w}) - IL(\vec{x}, \vec{w})| \tag{3}$$

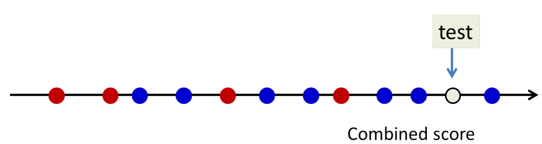

Fig. 1: The axis represents combined scores of instances. The red dots and blue dots represent negative and positive labelled instances, respectively. The grey dot is a test instance. 6 of the 7 positive instances score less than the test instance, hence the irrelevance loss is $6/7$. None of the negative instances score more than the test instance, hence the relevance loss is 0. The clarity is $|0 - 6/7| = 6/7$.

Figure 1 illustrates the relevance and irrelevance losses and the clarity index. It is obvious that the higher the *clarity index*, the easier it is to make a decision for $\vec{x}$. The range of the clarity index is $[0, 1]$ and it is desired for it to be high for any unlabeled instance.

We also define the *Raw Clarity Index (RCL)* as the difference between *RL* and *IL*. Thus $RCL(\vec{x}, \vec{w}) = RL(\vec{x}, \vec{w}) - IL(\vec{x}, \vec{w})$ and the range of RCL is $[-1, 1]$. CL is the absolute value of RCL. For a positive instance we expect the *raw clarity index* to be negative; the closer it is to -1 better it is. Similarly for a negative instance the desired value of RCL is to be positive and high. In all cases, the CL value should be high.

We optimize $\vec{w}$ for each instance $\vec{x}$, by maximizing CL with respect to w.

$$\vec{w}_x = \arg\max_{\vec{w}} CL(\vec{x}, \vec{w}) \tag{4}$$

We set the membership score of $\vec{x}$ as the raw clarity score at $\vec{w}_x$:

$$r(\vec{x}) = RCL(\vec{x}, \vec{w}_x)$$

Instances are now ranked according to $r(\vec{x})$.

3.3. Estimating $\vec{w}_x$

Direct optimization of CL with respect to $\vec{w}$ is intractable in general, however, because the indicator function I in the definitions of RL and IL is not differentiable. We approximate it instead by a smooth, differentiable sigmoid function:

$$I(t) \approx I_s(t) = \frac{1}{1 + e^{-\alpha t}} \tag{5}$$

By choosing the correct α this function can be made arbitrarily close to the indicator function I.

Using this approximation, the *relevance loss (RL)* and *irrelevance loss*(IL) are redefined as

$$\hat{RL}(\vec{x}, \vec{w}) = \frac{1}{|X_-|} \sum_{x_- \in X_-} \frac{1}{1 + e^{-\alpha \vec{w}^T (\vec{x}_- - \vec{x})}} \tag{6}$$

$$\hat{IL}(\vec{x}, \vec{w}) = \frac{1}{|X_+|} \sum_{x_+ \in X_+} \frac{1}{1 + e^{-\alpha \vec{w}^T (\vec{x} - \vec{x}_+)}} \tag{7}$$

The raw clarity index is now approximated as $\hat{RCL}(\vec{x}, \vec{w}) = \hat{RL}(\vec{x}, \vec{w}) - \hat{RL}(\vec{x}, \vec{w})$.

The *clarity* is the absolute value of the raw clarity. The absolute value function, however, is again non-differentiable and hence unsuitable for optimization. We optimize $\hat{CL}(\vec{x}, \vec{w}) = |\hat{RCL}(\vec{x}, \vec{w})|$ instead, using the following approach.

$$\begin{aligned} \vec{w}_{max} &= \arg\max_{\vec{w}} \hat{RCL}(\vec{x}, \vec{w}) \\ \vec{w}_{min} &= \arg\min_{\vec{w}} \hat{RCL}(\vec{x}, \vec{w}) \end{aligned}$$

$$\vec{w}_x = \begin{cases} \vec{w}_{max}, & if \;\; \hat{RCL}(\vec{x}, \vec{w}_{max}) > |\hat{RCL}(\vec{x}, \vec{w}_{min})| \\ \vec{w}_{min} & otherwise \end{cases} \tag{8}$$

Here we simultaneously find find both the maximum and minimum value of RCL, and choose the weight for which the absolute value of RCL is higher. Maximization of RCL is performed with a simple gradient *ascent* procedure. Minimization is performed using gradient *descent* starting from the same initial location used for the maximization. This simplifies an otherwise complicated optimization procedure for an absolute value function. The weights are also subjected to constraints of $\vec{w} > 0$, since the classifiers are assumed to be no worse than random. Moreover, to keep weights from exploding we put the additional constraint $|\vec{w}^T \vec{w}| = 1$. This essentially means that we project the weights on to a hypersphere in the positive orthant during each update step in the optimization. This overall procedure can be also be thought of as margin maximization over clarity measure. We note that the overall algorithm might get stuck in local optima.

The final membership score $r(\vec{x}) = RCL(\vec{x}, \vec{w}_x)$. Note that we use the raw clarity RCL value in $r(\vec{x})$. The optimized weight $\vec{w}_x$ can also be used to compute the *confidence* score $s(\vec{x}, \vec{w}_x) = \vec{w}_x^T \vec{x}$. The confidence value $s(\vec{x}, \vec{w}_x)$ too can be used to rank instances. We also evaluate this in our experiments.

3.4. Ranking and N-best Selection

The learned weights $\vec{w}_x$ can be thought of as importance values assigned to the primary classifiers. This allows us to choose the N-best classifiers using the learned weights for each instance. This can be beneficial when there are a few poor or noisy classifiers, as these classifiers can in fact be detrimental to overall performance. Our method, which is instance specific, can be used to discard poor classifiers for each instance using the estimated weights.

In the experiments below we also evaluate the utility of the proposed method for classifier selection.

4. EXPERIMENTS AND RESULTS

We performed our experiments on a subset of the TRECVID-MED13 database [1]. The task here is to retrieve videos for different event categories. The primary classifiers were trained using 13 different features including different variations of SIFT, CSIFT, MOSIFT and STIP features [18][19] for the video components of the recordings, and Acoustic Unit Descriptors (AUDS) [20] [21] for the audio component. The primary classifiers are χ^2 kernel based SVMs trained on these features. Thus overall we have $m = 13$ classifier outputs for each instance. Our training data for optimizing the classifier fusion was the same as the instances with which the SVMs were trained. X thus contained outputs of these instances on the trained SVMs. Overall we used 9232 videos: 5142 in training, 2041 in development and 2049 in test. These videos belong to 10 different *event* categories *namely*; “Birthday Party”, “Changing Vehicle Tire”, “Grooming an Animal”, “Parkour”, “Attempting a bike trick”, “Cleaning an appliance”, “Dog Show”, “Giving directions to a location”, “Marriage Proposal”, and “Working on a metal crafts project”. Since this is a retrieval task, several videos in the corpus do not belong to any of these 10 categories. These videos are thus negative example instances for all 10 classes.

We report results in terms of *Average Precision*(AP) for each category and *Mean Average Precision*(MAP) over all categories. The AP for any event is the average of the precision values for the classifier, when the retrieved ranked list is terminated at any true positive [22][23][24]. The MAP is the mean of the AP values for all events.

We show results obtained using both, the proposed *membership score* $r(\vec{x})$, and the confidence score obtained with the optimized weights: $s(\vec{x}, \vec{w}_x)$. We set the learning rate η in the gradient ascent updates used to learn $\vec{w}_x$ to 0.1. As a comparator, we also show results obtained with possibly the most common conventional fusion technique in these scenarios, which performs confidence-based ranking based on average weighting. The score used to rank instances is computed as $avg(\vec{x}) = m^{-1} \sum_i x_i$. Average weighting is a highly effective and robust technique and is known to outperform other, much more sophisticated methods in many problems. On this particular data set, average weighting was consistently observed to outperform weighting schemes where weights were learned by SVMs and logistic regression classifiers; consequently the performance obtained with these latter, nominally superior conventional weighting schemes is not shown.

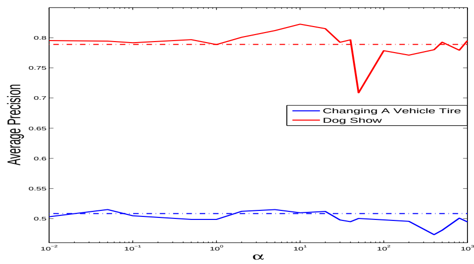

Fig. 2: AP as a function of α for 2 event classes. The horizontal dotted lines show the AP with average fusion and bold line shows AP using weighted scores (s^u). α in log scale.

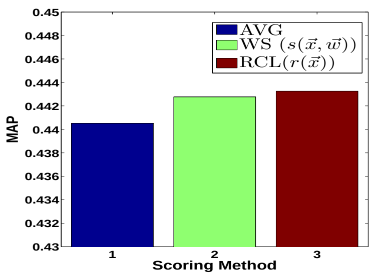

Fig. 3: Mean Average Precision (MAP) over all event classes for different ranking methods

α in the sigmoid function in Equation 5 is a key parameter. A high α value makes the sigmoid arbitrarily close to the true indicator function $I(t)$, but creates several local optima in the objective function, resulting in increased variance of the estimator. A low α value on the other hand will result in lower variance but a higher bias. This parameter can thus have a significant effect on outcome of algorithm. We show this dependence on the test set for 2 events in Figure 2.

Figure 2 shows that there is considerable variation in performance with α. Also, the performance obtained with the best α is significantly higher than that obtained with average fusion. In subsequent results, we choose α for each event class based on the performance on the development set. The results in Figure 3 show the MAP values over all event categories. 'AVG' corresponds to average weighing, 'WS' to weighted score $s(\vec{x}, \vec{w}_x)$ and RCL, standing for *raw clarity*, represents retrieval based on the proposed score $r(\vec{x})$. We see that ranking based on the proposed clarity-based membership score outperforms other methods.

As described in Section 3.4 the learned weights can be used to select classifiers. We can now evaluate retrieval based on the membership score, confidence measure and average weighting scores, but computed now only on the selected N-best classifiers. Performance (MAP) in this N-best selection schema is shown in Figure 4(a). It is clearly visible that it is possible to find an N for which the MAP value is significantly higher than when all classifiers are considered. This aspect is more pronounced if we look at individual events. One such example for event *Changing A Vehicle Tire* is shown in Figure 4(b). In the plot above N-best RCL-AVG corresponds to $r(\vec{x})$ computed with N-best classifiers but weighed by average weight. N-best RCL-WS correspond to $r(\vec{x})$ computed with N-best classifiers weighed by estimated weight($\vec{w}_x$). We also test the robustness of our method when some of the classifiers are noisy. We do this

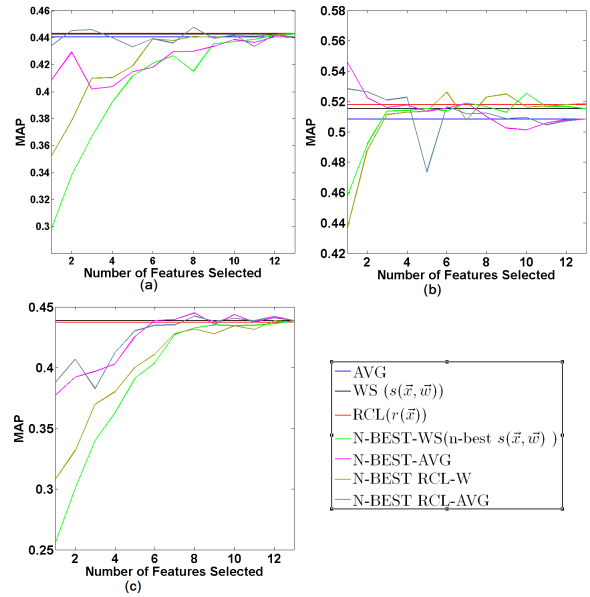

Fig. 4: Results on average fusion of N-best scores (N-BEST AVG) and weighted fusion of N-best scores (N-BEST WS), RCL obtained on the N-best classifiers with average weights (N-BEST RCL-AVG), RCL on N-best classifiers with estimated weights (N-BEST RCL-W) as a function of N. (a) Left: MAP of all 10 events (b) Right: AP for Event *Changing Vehicle Tyre* (c) MAP values as a function of N when noise is added to classifier outputs

by introducing noise in the outputs of the classifiers. The outputs of 20% randomly chosen test points for 3 of the classifiers are corrupted by adding random noise. This obviously results in a drop in the MAP value. However, using N-best selection methods, much better results can be obtained as illustrated in Figure 4(c).

5. CONCLUSIONS

Our results show that the proposed clarity-based ranking can outperform conventional retrieval based on weighted average scores. Raw clarity based scoring consistently outperforms others, in both take all and N-best terms. The overall idea is to maximize *clarity* measure and achieved that using *raw clarity measure*. This works well in the current set of experiments. The results on individual events are not shown because of space constraints. The relative improvement for individual events ranged from 0.5%-4% using weighted scores $s(\vec{x}, \vec{w}_x)$ and 1.80%-11.57% when ranking is performed using $r(\vec{x})$. However, we note that for $2-3$ of the events there is a slight decrease in performance. This might be attributed to imperfect selection of α. Nonetheless, we saw an overall improvement in MAP using both the weighted score and the proposed index, $r(\vec{x})$. We also showed the utility of the method in selecting N-best classifiers for each instance. The variation with N is expected. In practice N may be optimized using a development set. Our method also demonstrates a significant robustness to noisy classifiers. We note that classifiers can give noisy results on newer test instances and an ideal method should be somehow able to use these noisy scores to achieve good performance. Our method is able to do that as is evident in the N-best plots in Figure 4(c). A more theoretical analysis might be able to answer the question of selecting the best α which still needs to investigated.

6. REFERENCES

[1] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quenot, "Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[2] F. Moreno-Seco, José M Inesta, Pedro J P De León, and Luisa Micó, "Comparison of classifier fusion methods for classification in pattern recognition tasks," in *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 705–713. Springer, 2006.

[3] O. L Junior, D. Delgado, V Gonçalves, and U Nunes, "Trainable classifier-fusion schemes: an application to pedestrian detection," in *Intelligent Transportation Systems*, 2009.

[4] J Kittler, "Combining classifiers: A theoretical framework," *Pattern analysis and Applications*, vol. 1, no. 1, pp. 18–27, 1998.

[5] Christopher J Merz, "Using correspondence analysis to combine classifiers," *Machine Learning*, vol. 36, no. 1-2, pp. 33–58, 1999.

[6] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *The Journal of machine learning research*, vol. 4, pp. 933–969, 2003.

[7] T S Huang, C K Dagli, S Rajaram, E Y Chang, M I Mandel, G E Poliner, and D PW Ellis, "Active learning for interactive multimedia retrieval," *Proc. of the IEEE*, vol. 96, no. 4, pp. 648–667, 2008.

[8] J Kittler, "A framework for classifier fusion: Is it still needed?," in *Advances in Pattern Recognition*, pp. 45–56. Springer, 2000.

[9] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar, "Rank aggregation methods for the web," in *Proc. of the 10th Intl. conference on World Wide Web*. ACM, 2001, pp. 613–622.

[10] Merijn Van Erp and Lambert Schomaker, "Variants of the borda count method for combining ranked classifier hypotheses," in *In The 7th Intl. Workshop On Frontiers In Handwriting Recognition 2000. Amsterdam Learning Methodology Inspired By Human's Intelligence Bo Zhang, Dayong Ding, And Ling Zhang*. Citeseer, 2000.

[11] D Sculley, "Rank aggregation for similar items.," SIAM.

[12] G. Lebanon and J. Lafferty, "Cranking: Combining rankings using conditional probability models on permutations," in *ICML*. Citeseer, 2002, vol. 2, pp. 363–370.

[13] Anton Leuski and James Allan, "Improving interactive retrieval by combining ranked list and clustering.," in *RIAO*, 2000, pp. 665–681.

[14] David H Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[15] Ljupčo Todorovski and Sašo Džeroski, "Combining classifiers with meta decision trees," *Machine learning*, vol. 50, no. 3, pp. 223–249, 2003.

[16] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 399–402.

[17] Rachid Benmokhtar and Benoit Huet, "Classifier fusion: combination methods for semantic indexing in video content," in *Artificial Neural Networks–ICANN 2006*, pp. 65–74. Springer, 2006.

[18] Zhen-Zhong Lan, Lu Jiang, Shoou-I Yu, Shourabh Rawat, Yang Cai, Chenqiang Gao, Shicheng Xu, Haoquan Shen, Xuanchong Li, Yipei Wang, et al., "Cmu-informedia at trecvid 2013 multimedia event detection," in *Proc. TRECVID 2013 Workshop, Gaithersburg, MD, USA*, 2013.

[19] Wei Tong, Yi Yang, Lu Jiang, Shoou-I Yu, ZhenZhong Lan, Zhigang Ma, Waito Sze, Ehsan Younessian, and Alexander G Hauptmann, "E-lamp: integration of innovative ideas for multimedia event detection," *Machine vision and applications*, vol. 25, no. 1, pp. 5–15, 2014.

[20] Sourish Chaudhuri, Mark Harvilla, and Bhiksha Raj, "Unsupervised learning of acoustic unit descriptors for audio content representation and classification.," .

[21] Anurag Kumar, Pranay Dighe, Rita Singh, Sourish Chaudhuri, and Bhiksha Raj, "Audio event detection from acoustic unit occurrence patterns.," in *ICASSP*, 2012, pp. 489–492.

[22] Stephen Robertson, "A new interpretation of average precision," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 689–690.

[23] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims, "A support vector method for optimizing average precision," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 271–278.

[24] Emine Yilmaz and Javed A Aslam, "Estimating average precision with incomplete and imperfect judgments," in *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 102–111.