COMMUNITY-DRIVEN HIERARCHICAL FUSION OF NUMEROUS CLASSIFIERS: APPLICATION TO VIDEO SEMANTIC INDEXING

Hervé Bredin

Spoken Language Processing Group
LIMSI-CNRS, BP 133
91403 Orsay Cedex, France
bredin@limsi.fr

ABSTRACT

We deal with the issue of combining dozens of classifiers into a better one. Our first contribution is the introduction of the notion of *communities of classifiers*. We build a complete graph with one node per classifier and edges weighted by a measure of similarity between connected classifiers. The resulting community structure is uncovered from this graph using the state-of-the-art *Louvain* algorithm. Our second contribution is a hierarchical fusion approach driven by these communities. First, intra-community fusion results in one classifier per community. Then, inter-community fusion takes advantage of their complementarity to achieve much better classification performance. Application to the combination of 90 classifiers in the framework of *TRECVid 2010 Semantic Indexing* task shows a 30% increase in performance relative to a baseline flat fusion.

Index Terms— community detection, late fusion, hierarchical fusion, semantic indexing

1. INTRODUCTION

Semantic indexing, as defined in the TRECVid evaluation campaign, consists in automatically detecting the presence of visual concepts in pre-segmented video shots [1] and returning a ranked list of shots the most likely to contain a given concept. Judging from the performance obtained by the best system in 2010 (with a mean inferred average precision on 30 concepts of 0.090), there is still a long way to go to solve this problem [2]. Some concepts appear to be much easier to detect than others and no single classifier emerges as *the one* that systematically (for any concept) outperforms the others. Therefore, for the sake of universality, most systems rely on the combination of a large (100+) set of classifiers. They usually differ in the type of descriptors (color, texture, or bag of visual words, etc.) or the machine learning algorithm (support vector machine or k nearest neighbors, for instance) they rely on.

This paper focuses on the last step of this common semantic indexing pipeline: the late fusion of available classifiers. Let $\mathtt{K}$ be their number and $\mathtt{N}$ the number of video shots. Each classifier $k \in \{1 \dots \mathtt{K}\}$ provides scores $\mathbf{x}_k = [x_{k1}, \dots, x_{k\mathtt{N}}]$ indicating the likelihood for each shot $n \in \{1 \dots \mathtt{N}\}$ to contain the requested concept. The objective is to find a combination function $\mathbf{f}$ so that the resulting classifier $\mathbf{x} = \mathbf{f}(\mathbf{x}_1, \dots, \mathbf{x}_\mathtt{K})$ is better than any of its components, and as good as possible.

1.1. Motivations

When looking for an effective combination of classifiers, several interrogations arise. Should we use them all in the fusion process, or just the best ones? Does combining two classifiers always yield better results than the two of them taken separately? Should we weigh them differently in case one is much better than the other? Tackling a similar problem, *Ng and Kantor* [3] proposed a method to predict the effectiveness of their fusion approach and concluded:

> [...] *schemes with dissimilar outputs but comparable performance are more likely to give rise to effective naive data fusion.*

where the *similarity* between two classifier *outputs* can be measured as the Spearman rank correlation coefficient (see paragraph 2.1 for details) – and *naive data fusion* should be understood as fusion by sum of normalized scores.

We went one step further and drove a simple experiment whose outcome is summarized in Figure 1. Given a set of $\mathtt{K} = 50$ classifiers and an estimation of their performance (average precision) α_k on the *TRECVid 2010 Semantic Indexing* task, we considered all pairs (i, j) of classifiers and evaluated the performance of their fusion by weighted sum of normalized scores:

$$\mathbf{x} = \alpha_i \cdot \mathbf{x}_i + \alpha_j \cdot \mathbf{x}_j \tag{1}$$

Each circle corresponds to one of those pairs. The x-axis corresponds to $\max(\alpha_i, \alpha_j)$, which is the performance of the best of the two classifiers. The y-axis indicates α_{i+j}, the performance of the classifier resulting from their combination. Dark (resp. bright) grey circles indicate that classifiers i and j strongly agree (resp. disagree) in their rankings. The circle diameter is directly proportional to the ratio of their performance α_i / α_j (where $\alpha_i < \alpha_j$).

As most circles are above the $x = y$ line (i.e. $\alpha_{i+j} > \max(\alpha_i, \alpha_j)$), Figure 1 clearly shows that, when classifiers performance is accurately predicted, the weighted sum fusion approach described in Equation 1 is almost always beneficial. Moreover, it confirms that we should combine classifiers that tend to disagree on their rankings and have similar performance (bright large circles) to achieve the best performance.

1.2. Outline

Based on this observation, we propose a novel approach to hierarchical combination of multiple classifiers. First, Section 2 describes how meaningful communities of classifiers are automatically detected. Then, Section 3 presents our novel approach to hierarchical fusion, based on these communities. Experiments driven on the *TRECVid 2010 Semantic Indexing* task are summarized in Section 4. Finally, Section 5 concludes the paper.

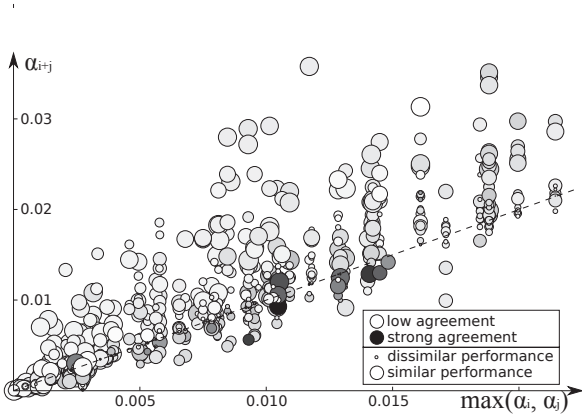

Fig. 1. Which classifiers should we combine?

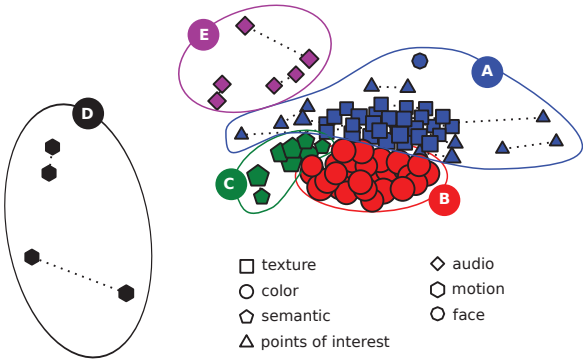

Fig. 2. Communities of classifiers for concept *Computers*

2. COMMUNITIES OF CLASSIFIERS

This section introduces one of the main contributions of this paper: the notion of communities of classifiers.

2.1. Classifiers agreement

For each classifier k, the raw scores $\mathbf{x}_k$ are sorted and converted into ranks $\mathbf{r}_k \in \{1 \dots \mathtt{N}\}$, so that $r_{kn} = 1$ (resp. $\mathtt{N}$) for the shot whose value x_{kn} is the maximum (resp. minimum). Let us denote ρ_{ij} the Spearman rank correlation coefficient of two classifiers i and j:

$$\rho_{ij} = \frac{\sum_{n=1}^{n=\mathtt{N}} (r_{in} - \overline{r_i})(r_{jn} - \overline{r_j})}{\sqrt{\sum_{n=1}^{n=\mathtt{N}} (r_{in} - \overline{r_i})^2 \sum_{n=1}^{n=\mathtt{N}} (r_{jn} - \overline{r_j})^2}} \quad (2)$$

ρ_{ij} ranges from -1 (one ranking is the exact opposite of the other one) to 1 (rankings are identical). $\rho_{ij} = 0$ can be understood as classifiers being independent from each other. We then define the agreement A_{ij} between two classifiers i and j:

$$A_{ij} = \max(0, \rho_{ij}) \quad (3)$$

2.2. Classifiers (social) network

A complete undirected graph $\mathcal{G}$ is constructed with one node per classifier. Each pair of classifiers (i, j) is connected by an undirected edge, whose weight is directly proportional to A_{ij}. An instance of such a graph is drawn in Figure 2 for the TRECVid 2010 concept *Computers*. For the sake of clarity, most edges are not drawn here – but do remember that the graph is complete. It is represented using the so-called *spring layout*. Therefore, classifiers with higher A_{ij} tend to be positioned closer to each other.

Looking at the relative position of classifiers, it appears that some kind of community structure naturally emerges. As in a social network, several groups of classifiers are more strongly connected internally than with the rest of the network. One can partially explain this structure by the low-level descriptors used internally by the classifiers. This information is denoted by the shape of the nodes in Figure 2. For instance, classifiers based on color descriptors (circles) seem to agglutinate, as do classifiers based on audio features (diamonds). Finally, the size of nodes in Figure 2 is directly proportional to the performance of the corresponding classifier. Therefore, best performing classifiers (i.e. larger nodes) also tend to agglutinate as they provide rankings that are closer to the true ranking than bad ones – therefore closer to each other...

2.3. Automatic community detection

In order to make use of this unique property, we rely on the so-called *Louvain* approach for automatic community detection proposed by *Blondel et al.*, and apply it on graph $\mathcal{G}$ previously defined in paragraph 2.2. It is a heuristic method that is based on the maximization of modularity $\mathcal{Q}$:

$$\mathcal{Q} = \frac{1}{\sum_{i,j} A_{ij}} \sum_{i,j} \left[A_{ij} - \frac{\sum_k A_{ik} \sum_k A_{kj}}{\sum_{i,j} A_{ij}} \right] \delta_{ij} \quad (4)$$

where $\delta_{ij} = 1$ if classifiers i and j are members of the same community, 0 otherwise. $\mathcal{Q}$ can be seen as a measure of the quality of the detected communities. It increases when communities have stronger intra-community and weaker inter-community edges [4].

Starting with as many communities as there are nodes, the *Louvain* approach looks at all nodes for a potential change of community resulting in a higher modularity. Once modularity can no longer be improved, a new graph is built – in which every community is a node and edges are weighted by the sum of the corresponding edges in the original graph. This process is repeated until the maximum of modularity is attained. For a more detailed description and analysis of the algorithm, the interested reader might want to have a look at reference [5].

With no objective groundtruth to compare with, it is difficult to evaluate the detected communities. However, looking at Figure 2 and the five detected communities (A to E), it seems the *Louvain* algorithm did a good job finding communities related to the nature of the low-level descriptors on which classifiers are based. In particular, a dotted edge between a pair of classifiers indicates that they are based on the very same descriptors and they only differ in the machine learning algorithm they rely on. None of these pairs is split into two different communities.

3. COMMUNITY-DRIVEN HIERARCHICAL FUSION

Figure 3 summarizes our second main contribution: a novel approach to hierarchical fusion. It can be divided into three consecutive steps.

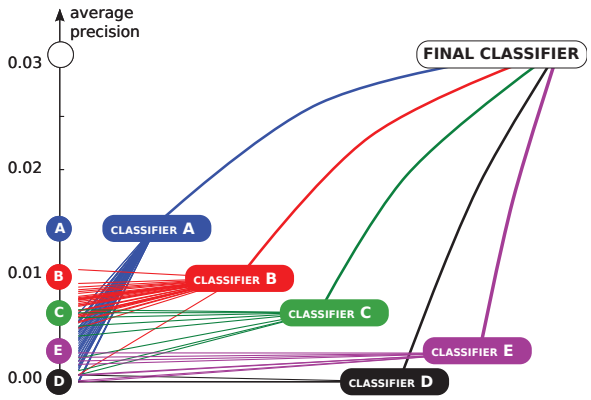

Fig. 3. Community-driven hierarchical fusion

Step 1: community detection. Classifiers are automatically grouped into $\mathtt{C}$ communities using the *Louvain* method described in Section 2. $\mathtt{C} = 5$ communities (A to E) are detected in Figure 2. This step is completely unsupervised as it is only based on the scores $\mathbf{x}_k$ provided by the classifiers on the test set, $k \in \{1 \dots \mathtt{K}\}$.

Step 2: intra-community fusion. Classifiers from each community are combined by simple sum of normalized scores, in order to obtain one new classifier per community (classifiers A to E in Figure 3):

$$\mathbf{x}_c = \sum_{k=1}^{k=\mathtt{K}} \delta_c(k)\, \widehat{\mathbf{x}_k} \tag{5}$$

with $\delta_c(k) = 1$ if classifier k is part of community c (and 0 otherwise). Those new classifiers are expected to be at least as good as the best of their components (like classifiers B to E in Figure 3) and can sometimes lead to much better performance (classifier A). This step is also completely unsupervised.

Step 3: inter-community fusion. Since they come from different communities, these new *community classifiers* are very likely to output very dissimilar scores and rankings. Therefore, as proposed in paragraph 1.1, they are combined using weighted sum fusion of normalized scores:

$$\mathbf{x} = \sum_{c=1}^{c=\mathtt{C}} \alpha_c \widehat{\mathbf{x}_c} \tag{6}$$

To this end, the performance α_c of each of these new *community classifiers* needs to be estimated using a development set: this makes this step supervised.

Steps 2 and 3 both rely on normalized scores. We investigated multiple normalization techniques (min/max, σ/μ, TanH) but only report on the one that proved to be the best, TanH normalization [6]:

$$\widehat{x_{kn}} = \frac{1}{2}\left\{\tanh\left[0.01\left(\frac{x_{kn}-\mu_k}{\sigma_k}\right)\right]+1\right\} \tag{7}$$

with μ_k (resp. σ_k) is the mean (resp. standard deviation) of scores provided by classifier k on test set.

4. EXPERIMENTS

In the framework of *TRECVid 2010 Semantic Indexing* task [1], NIST and Quaero provided participants with two corpora annotated with 50 concepts: 120k shots for the development set, $\mathtt{N} = 150$k shots for the test set. In order to evaluate the performance of our proposed approach for fusion, we gathered the scores on development and test sets of $\mathtt{K} = 90$ classifiers donated by the IRIM consortium [7]. We compared our community-driven hierarchical fusion approach with several others:

- two flat fusion baselines (all classifiers belong to the same community) with or without score normalization:

$$\mathbf{x} = \sum_{k=1}^{k=\mathtt{K}} \alpha_k \widehat{\mathbf{x}_k} \tag{8}$$

- one hierarchical fusion approach with random communities (results are averaged on 50 different runs),
- two community-driven hierarchical approaches where communities are obtained using complete-link or single-link agglomerative clustering [8] based on the similarity matrix A. The agglomerative clustering stops when the similarity between current clusters (communities) is lower than a tuned threshold θ.

We report in Table 1 both the arithmetic and geometric means of extended inferred average precision (xinfAP) over all 50 concepts [9]. However, since some concepts are much easier to detect than others, we claim that geometric mean makes more sense as a given relative improvement for one concept will have the same impact on the overall value, whatever that concept is. Focusing on the arithmetic mean will tend to bias the results in favor of methods performing well on easy concepts (xinfAP varies from less than 0.001 up to more than 0.700 depending on the concept).

Fusion	**Ari. mean xinfAP**	**Geo. mean xinfAP**
Flat (no norm.)	0.0595 (−3%)	0.0186 (−9%)
Flat (TanH)	**0.0614**	**0.0204**
Random (50×)	0.0618 (+1%)	0.0214 (+5%)
Complete-link*	0.0679 (+11%)	**0.0266** (+31%)
Single-link*	**0.0686** (+12%)	0.0258 (+27%)
Louvain	0.0634 (+3%)	**0.0264** (+30%)

Table 1. Results for 50 concepts on TRECVid 2010 test set. Improvement over Flat (TanH) is shown between brackets.

Results obtained by the flat fusion approaches show that score normalization is very beneficial, especially for sum-based fusion techniques like ours: it prevents one bad classifier from outweighing all others simply because it outputs scores in a higher range. The +5% improvement brought by the *Random* approach also needs to be highlighted. It shows that, however classifiers are grouped together, hierarchical fusion appears to be a good practice. Results marked with a star * are biased, in that they depend on the threshold θ that is optimized on the test set. Figure 4 shows how the choice of θ may affect the performance.

All in all, Table 1 shows that our proposed approach and its variant based on complete-link clustering yield the best performance with a 30% relative improvement over flat fusion baseline. However, Figure 4 brings to light that our preferred approach has the strong advantage not to depend on an additional threshold for which bad tuning could be catastrophic.

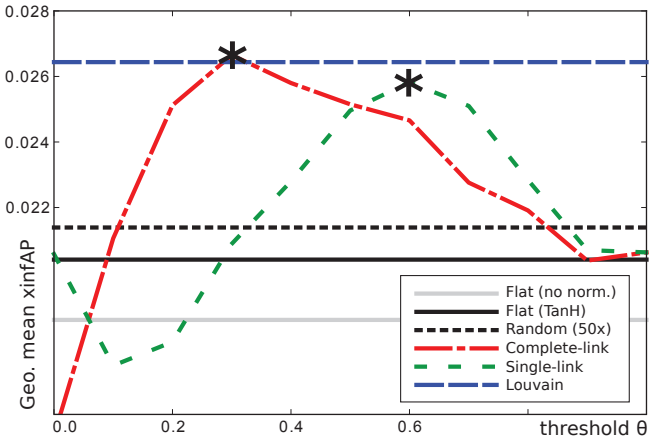

Fig. 4. Effect of threshold θ

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed an efficient way to combine dozens of classifiers into a better one, and applied it successfully in the context of the *TRECVid 2010 Semantic Indexing* task: $+30\%$ relative increase of performance over a flat fusion baseline.

Our first contribution is the introduction of the notion of *communities of classifiers*. It results from the application of a state-of-the-art community detection algorithm on a complete graph where each classifier is a node and each pair of classifiers is connected by an edge weighted according to their similarity. In the near future, two points will be the object of a more detailed study:

Similarity measures So far, classifiers similarity is computed as the Spearman rank correlation coefficient on the whole set of scores. However, in an information retrieval paradigm, only top ranked documents are really of interest. Similarity measures taking this observation into account will be investigated.

Community detection algorithms Community detection in graphs is a very active field of research [10]. *Louvain* approach is only one among many others and we should make sure that it is optimal for our own application.

Our second contribution is a three-steps hierarchical fusion approach driven by these communities. Although it was found to be very successful in our experiments on the *TRECVid 2010 Semantic Indexing* task, there is lot more to be done in this direction. In particular, the following issues will be addressed in the future:

Fully unsupervised fusion The third step of our proposed fusion approach is the only one that needs supervision. We are currently looking for ways to predict the relative performance of classifiers in a unsupervised manner – so that we can use this information to make the whole process completely unsupervised.

Other baseline fusion techniques Linear combination of scores is not the only solution to the late fusion problem [11]. There is no reason our community-driven hierarchical approach cannot be applied to other (possibly better) techniques such as logistic regression or support vector machine in the score space, for instance.

6. ACKNOWLEDGMENT

This work was partly realized as part of the Quaero Program and the QCompere project, respectively funded by OSEO (French State agency for innovation) and ANR (French national research agency). The author would also like to thank the members of the IRIM consortium [7] for the classifier scores used throughout the experiments described in this paper.

7. REFERENCES

[1] Alan F. Smeaton, Paul Over, and Wessel Kraaij, "High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements," in *Multimedia Content Analysis, Theory and Applications*, Ajay Divakaran, Ed., pp. 151–174. Springer Verlag, Berlin, 2009.

[2] Cees G. M. Snoek, Koen E. A. van de Sande, Ork de Rooij, Bouke Huurnink, Efstratios Gavves, Daan Odijk, Maarten de Rijke, Theo Gevers, Marcel Worring, Dennis C. Koelma, and Arnold W. M. Smeulders, "The MediaMill TRECVID 2010 Semantic Video Search Engine," in *Proceedings of the 8th TRECVID Workshop*, Gaithersburg, USA, November 2010.

[3] Kwong Bor Ng and Paul B. Kantor, "Predicting the Effectiveness of Naive Data Fusion on the Basis of System Characteristics," *Journal of the American Society for Information Science*, vol. 51, pp. 1177–1189, November 2000.

[4] Mark E. J. Newman, "Modularity and Community Structure in Networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, June 2006.

[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, "Fast Unfolding of Communities in Large Networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. P10008, 2008.

[6] Arun A. Ross, Karthik Nandakumar, and Anil K. Jain, *Handbook of Multibiometrics (International Series on Biometrics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[7] David Gorisse, Frédéric Precioso, Philippe Gosselin, Lionel Granjon, Denis Pellerin, Michéle Rombaut, Hervé Bredin, Lionel Koenig, Hélène Lachambre, Elie El Khoury, Rémi Vieux, Boris Mansencal, Yifan Zhou, Jenny Benois-Pineau, Hervé Jégou, Stéphane Ayache, Bahjat Safadi, Georges Quénot, Alexandre Benoît, and Patrick Lambert, "IRIM at TRECVID 2010: Semantic Indexing and Instance Search," in *Proceedings of the 8th TRECVID Workshop*, 2010.

[8] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn, "Data Clustering: a Review," *ACM Computing Surveys*, vol. 31, pp. 264–323, September 1999.

[9] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam, "A Simple and Efficient Sampling Method for Estimating AP and NDCG," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2008, SIGIR '08, pp. 603–610, ACM.

[10] Santo Fortunato, "Community Detection in Graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75 – 174, 2010.

[11] Pradeep Atrey, M. Answar Hossain, Abdulmotaleb El Saddik, and Mohan Kankanhalli, "Multimodal Fusion for Multimedia Analysis: a Survey," *Multimedia Systems*, pp. 1–35, 2010.