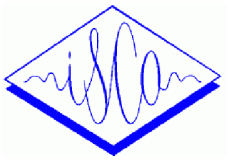
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Applying multiview learning algorithms to human-human conversation classification

Sokol Koço, Cécile Capponi, Frédéric Béchet

Aix Marseille Université, LIF-CNRS, Marseille, France
firstname.lastname@lif.univ-mrs.fr

Abstract

We propose in this paper to use a novel multiview boosting-like algorithm called *Mumbo* for processing human-human conversations recorded in a call-senter setting. We present how dialog classification can be seen as a multiview classification problem and we compare the performance of *Mumbo* and the one of a standard boosting algorithm. The first results obtained on a subset of the DECODA corpus show that a significant improvement in classification performance can be achieved, especially on high Word Error Rate transcriptions.

Index Terms: human-human conversation, multiview learning, boosting, spoken language understanding.

10.21437/Interspeech.2012-609

1. Introduction

Multiview classification algorithms have been used in many multimedia processing tasks, involving audio and image processing. Each view is supposed to carry some information that the other views would not embed, and each view can be more or less noisy. Multiview algorithms are designed to select the most informative set of features, that either best discriminate data concepts or best describe one concept among others [4] [11].

Even when processing audio alone, speech processing systems always use several sets of features that can be seen as several views. The aim of this paper is to investigate how some multiview learning algorithms could deal with the specificities of human-human spoken dialog processing. More specifically, we aim at illustrating that the use of several views (sets of features) extracted from the original signals, can *together* enhance the overall performances for a given classification task.

We propose to use a recent multiview boosting-like algorithm called *MuMBo* [8] which is applied for the first time to speech data in this paper. The experimental set-up is a dialog classification task using the DECODA corpus [1]. This corpus contains human-human conversations recorded in a call-center setting. We present in this paper how dialog classification can be seen as a multiview classification problem and we compare the performance of *MuMBo* and the one of a standard boosting algorithm.

2. Related works

Up to now, several approaches of multiview learning have been developed in the machine learning community, mostly in the semi-supervised setting. The first of them was the well-known *Co-Training* algorithm [2], which was based on far too much restrictive assumptions [3]. In addition, in the supervised setting, leveraging the performances of classifiers learned on different views has mainly been performed through fusion-based methods, either early or late fusion [7] [12]. Early fusion consists in grouping (selected) features of the different views into a large vector, and then learning a classifier on this resulting view. On the opposite, late fusion allows one classifier per view to be learned, while the final classifier is a combination of them. Unless data are quite noisy, late fusion usually performs better than early fusion. Yet, none of them leads to good performances when the views are of unbalanced informative content, for weaker views tend to reduce the final performances. An empirical comparison of these methods applied on multimedia problems is presented in [10].

Whatever the fusion-based approach is, it relies on a weighted combination of classifiers (or selected descriptions), where classifiers are learned independently. One drawback of these methods is that the classifier learned on one view does not communicate its failures to the other views. Besides, views must be independent in order for the combined classifier to be most accurate.

In addition to such methods, we think it could be interesting to test a multiview learning algorithm that promotes cooperation between views. The basic principle of such an algorithm is that whenever the classifier learned on a view fails on a region of examples in the instance space, it could entrust the other views with the classification of these examples. One of the major difficulties is then to delimit the concerned subareas of the instance space, without loss of generalization capabilities. Instead of precisely locating these subareas, the boosting-like algorithm MuMBo [8] slightly removes the harder examples from the learning space of one view, while their weights increase in the other views. This way, examples are expected to be processed by the most appropriate views.

The underlying principles of MuMBo on one hand, and the usual early-fusion boosted classifiers on the other hand, are quite different. This work compares these two approaches on speech data that comes from the recordings of phone calls.

3. Multi-Modal Boosting Algorithm

3.1. The multiview setting

In the multiview setting, each example of a training sample S is represented by several sets of features. These sets of features, which are called *views*, can be used to train several models using different learning algorithms (stumps, SVM, neural nets, etc.). Even though these models are learned on different representations of the same examples, they are by no means equal performance-wise whatever the learning algorithm. Indeed, the performances of these models can be quite different due to the presence of more or less noise in the sets of features, and/or the lack of informations. In order to characterize this behavior, we use the notion of *strength of a view*.

Let S be a sample of n tagged examples chosen according to some distribution $\mathcal{D}$ over $X \times Y$, where $X = X_1 \times ... \times X_m$ is an instance space made up of m different views, and Y is the class space. Let V_j be the view corresponding to X_j, $\mathcal{H}_j$ be the space of all the hypothesis that we can learn on V_j and h_j be the optimal classifier that we can learn on $\mathcal{H}_j$. Finally, let ρ be the error of random guessing over S and $\sigma_j \leq \rho$ be the lower bound of the error of h_j on S.

We can define the notion of weak and strong view as follows: V_j is a strong view if σ_j is near 0, while V_j is a weak view if the edge $\gamma_j = \rho - \sigma_j$ is near 0. This definition follows the definition of weak classifiers as presented in [9].

In other words, the strength of a view is defined as the possibility to learn a good classifier on that view whatever the hypothesis space is. Then, the weakness of a view reflects the impossibility of learning a good classifier from the instance space X_j of this view.

MuMBo is a boosting algorithm that was designed in order to promote the cooperation between strong and weak views, by implementing the sharing of classification information among views.

3.2. Framework and notations

MuMBo [8] is a multiview, multiclass learning algorithm which was designed within the framework defined by [9], where basically γ denotes the edge of a classifier with regards to random guessing. We use the following typings:

- matrices are denoted by bold capital letters like $\mathbf{C}$,
- element of row i and column j in a matrix $\mathbf{C}$ is denoted $\mathbf{C}(i,j)$; then $\mathbf{C}(i)$ denotes the row i in the matrix $\mathbf{C}$,
- the indicator function is denoted by $\mathbb{1}[\cdot]$,
- the cartesian product is denoted by $X_1 \times X_2$.

Let $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ be the learning sample, where $x_i \in X$ is the description of the i^{th} example of S, and $y_i \in Y$ is the class of x_i. The set of classes is $Y = \{1, ..., k\}$. The set of features X is made up of different subsets: $X = X_1 \times ... \times X_m$, where each subset represents a view, as in [6]. Then, the representation of example x_i within view m is written $x_{i,m}$[1].

3.3. A multiview algorithm : MuMBo

MuMBo (algorithm 1) is an attempt to promote the collaboration between major and minor views[2], in order to enhance the performances of classifiers usually learned only on the major view, using additional informations from the classifiers learned on weaker views. It is a boosting algorithm theoretically founded on the multi-class boosting framework presented in [9].

One of the main ideas of this framework is to *replace the weights of the examples with a cost matrix*. That cost matrix $\mathbf{C}$ is defined so that $\mathbf{C}(i,l)$ corresponds to the cost of assigning the class (label) l to the example i. The higher the cost (i,l) is, the most irrelevant is to assign l to i; thus, the higher $\sum_{l \neq y_i} \mathbf{C}(i,l)$ the hardest is the classification of x_i, following the basic principle of boosting-like algorithms.

[1] When possible, we simplify $x_{i,m}$ to x_i – or i – in the scope of view m.

[2] Major and minor views can be strong or weak views, but major views are always stronger than minor views.

Since MuMBo deals with several views, it uses one cost matrix $\mathbf{C}_j$ per view j, in addition to a global cost matrix $\mathbf{C}_G$ that embeds the cost of each example for each class, according to all the views considered as a global set of features. Thus $m+1$ cost matrices are maintained. They are uniformely initialized, although some prior knowledge could be represented within these matrices.

As a typical boosting algorithm, Mumbo runs for T rounds: at each round t, m weak classifiers $h_{t,j}$ are trained satisfying the weak learning condition defined in [9]. For each $h_{t,j}$, the parameter $\alpha_{t,j}$ is computed using the edge of $h_{t,j}$ on the cost matrix $\mathbf{C}_{t,j}$. It can be seen as a measure of the importance of $h_{t,j}$.

Algorithm 1 MuMBo: MUltiModal BOosting

Initialize each $\mathbf{C}_{1,j}$
for $t = 1$ **to** T **do**
  for $j = 1$ **to** m (for each view) **do**
    Train $h_{t,j}$ with learning algorithm L_j on $\mathbf{C}_{t,j}$
    Compute edge $\delta_{t,j}$ and $\alpha_{t,j} = \frac{1}{2} \ln \frac{1+\delta_{t,j}}{1-\delta_{t,j}}$
  end for
  Update cost matrix $\mathbf{C}_{t,j}$ (for each view)
  Choose

$$\begin{cases} h_t = \underset{h_{t,j}}{\text{argmax}}(\text{edge } h_{t,j} \text{ on } \mathbf{C}_{t,G}) \\ \delta_t = \{edge\ of\ h_t\ on\ \mathbf{C}_{t,G}\} \end{cases}$$

  Compute $\alpha_t = \frac{1}{2} \ln \frac{1+\delta_t}{1-\delta_t}$
  Update $\mathbf{C}_{t,G}$, the global cost matrix
end for
Output final hypothesis :

$$H(x) = \underset{l \in 1,...,k}{\text{argmax}} f_T(x,l)$$

As stated before, one of the main ideas of MuMBo is to have some sort of collaboration between the different views. This idea of cooperation is implemented in two different parts of MuMBo:

1. during the update of the cost matrices,
2. when choosing the unique classifier h_t for the round t.

The first part is implemented through the functions $f_{t,j}$ computed on each classifier (view) j at round t, as follows: let

$f_{t,j}(i,l) = \sum_{r=1}^{t} \mathbb{1}[h_{r,j}(i) = l] \alpha_{r,j} d_{r,j}(i)$,

$g_{[\cdot]}(i,l,p) = f_{[\cdot]}(i,l) - f_{[\cdot]}(i,p)$, and

$$d_{r,j}(i) = \begin{cases} 1 & \text{if } h_{r,j}(i) = y_i \text{ or} \\ & \nexists q \in \{1, ..., m\}, h_{r,q}(i) = y_i \\ 0 & \text{else} \end{cases}$$

Then the update rule for each cost matrix is defined as :

$$\mathbf{C}_{t,j}(i,l) = \begin{cases} \exp(g_{t,j}(i,l,y_i)) & \text{if } l \neq y_i \\ -\sum_{p=1;p\neq y_i}^{k} \exp(g_{t,j}(i,l,y_i)) & \text{if } l = y_i \end{cases} \quad (1)$$

The i^{th} line, corresponding to the example x_i in the matrix of the view j, is updated only if the classifier learned on this view classifies correctly x_i OR if all the $m-1$ other weak classifiers misclassify it. Intuitively this means that a view gives up

on its hardest examples and lets the other views handle them. In the scenario of one major and several minor views, this allows the minor views to focus on the hardest examples of the major view.

As for the second part of the cooperation at each round t, MuMBo chooses the classifier h_t among the m classifiers (one per view) that minimizes the error on the global cost matrix $\mathbf{C}_G$[3]. The confidence α_t is computed for h_t, based on its edge δ_t on $\mathbf{C}_G$. Then the global cost matrix $\mathbf{C}_G$ is updated using a rule similar to equation 1. In this case, we define $f_t(i,l) = \sum_{r=1}^{t} \mathbb{1}[h_r(i) = l]\alpha_r$.

Similarly to other boosting algorithms, *the final hypothesis* H is a weighted vote of the T selected weak classifiers h_t, where α_t is the weight assigned to h_t.

Some important theoretical results on MuMBo are proven in [8], namely the convergence of the empirical error with t, and a bound of the generalization error along with t.

4. Experimental results

We present in this section the application of MuMBo to a human-human conversation classification task, in the context of the DECODA project.

4.1. Human-human spoken dialog classification

In the DECODA project[4] we are dealing with the call-center of the Paris transport authority (RATP). In this context we have collected human-human conversations between operators and public transport users in Paris. The themes of the conversations can be traffic info, route planning, lost and found, prices, timetable, etc. The automatic processing of these conversations is challenging because of the spontaneous nature of the language used and the surrounding noise in most of the speech files (people phoning from a cell phone while in a street, or a subway station). The task targetted in this study is conversation classification. This is a very common task in speech analytics of call-center data for obtaining automatic distributions of categories of calls for a given period of time. We work in this paper in a subset of the DECODA corpus made of 660 dialogs, split into 508 dialogs for training and 152 for the test corpus. We use a set of 9 semantic categories for characterizing the main theme of each conversation. This corpus is described in table 1. As we can see, the Word Error Rate (WER) on the test corpus is very high. However a lot of these errors come from the misrecognition of speech disfluencies, the insertions of small non-content words due to noisy signals, and therefore these errors will not affect too much the performance of the conversation classification task.

It is interesting to notice that there is more than a 10% absolute WER difference bewteen the automatic transcriptions of the operators' turns and the users' one. This can be explained by the fact that most of the operators are both in the training and test corpus and therefore the acoustic models are adapted to them. Another explanation is that the users' signal is significantly noisier than the operators' one since they often phone from a street or a subway station. However, because the recording device in the call-center mixes the two channels, the noise affects the whole signal.

To test the multiview approach of MuMBo on this corpus we had first to define what set of features to use and how to group them into different *views*. We defined 5 sets of features considered as 5 different *views* on the dialog:

1. word transcription of the user's turns
2. word transcription of the operator's turns
3. turn taking description between the operator and the users
4. dialog and average turn duration, % of speech for each speaker, speech rate (in letters per second)
5. bag of named entites extracted from the word transcriptions

The first two views are bags of ngrams extracted from the conversation transcriptions. The operators and users transcriptions are put in two different views firstly because they carry different kinds of information; secondly because the transcription quality is different for each of them as presented in table 1. The third view describes the interaction between the operator and the user. Each turn is represented by a letter, O for operator and U for users, and a number representing the duration of the turn normalized into 6 duration lengths. With this process a whole conversation is represented by a sequence such as: $O1U4O2U5O4U1O1U1\ldots$ The weak classifiers trained on this view are n-grams extracted from such sequences. The fourth view is about speech duration and speech rate for each speaker, expressed in numerical values. Finally the fifth view corresponds to the named entities (location, services, facilities) detected in each dialog.

	train		test		test
speaker	#turn	#word	#turn	#word	WER
Operators	12423	81422	4341	27408	56.3
Users	10867	75545	3632	23551	67.3

Table 1: Description of the conversation corpus and WER obtained on the test corpus

4.2. Results

Two experimental setting have been tested:

1. *test(gold)*: the classification performance of each method is evaluated on features obtained on the reference transcriptions (*gold*) of the test corpus
2. *test(ASR)*: in this case the features are obtained on the ASR transcriptions.

To compare the results of MuMBo with a standard classification approach, we use the *Icsiboost* [5] implementation of AdaBoost. Firstly, to check the strength of each view, we present in table 2 the classification error rate obtained on the two experimental settings with AdaBoost. As expected the two lexical views ($V1$ and $V2$) clearly outperform the other views.

setting	error	$V1$	$V2$	$V3$	$V4$	$V5$
test(gold)	#	40	43	107	104	82
	%	26.3	28.3	70.4	68.4	53.9
test(ASR)	#	65	55	107	104	82
	%	42.7	36.2	70.4	68.4	53.9

Table 2: Classification errors with AdaBoost on the five views ($V1\ldots5$) without fusion (152 examples in the test sample).

Three fusion algorithms between the views are compared in table 3: AdaBoost with late fusion (fusion of the decisions

[3] Many other selecting criterions can be applied!

[4] http://decoda.univ-avignon.fr

taken independently by each view); AdaBoost with early fusion (all the views are merged together); and our multiview algorithm MuMBo. The weak classifiers used in the three cases are decision stumps. We noticed that adding the *weak* views $V3, V4, V5$ to the strong ones improves the classification performance of all the algorithms, except in the case of early AdaBoost on *test(gold)*: it means that weaker views still carry some useful information. Late or early fusion don't have a strong impact on the AdaBoost performance in our experiments.

Algo	Errors	*test(gold)*	*test(ASR)*
AdaBoost(early)	#	37/152	49/152
	%	24.3	32.2
AdaBoost(late)	#	37/152	46/152
	%	24.3	30.3
MuMBo	#	**32/152**	**38/152**
	%	**21.1**	**25.0**

Table 3: Classification errors for MuMBo and Adaboost (late fusion and early fusion) on the two experimental settings

As we can see *MuMBo* gives better results than AdaBoost on both settings, especially when the classification process is applied on the noisy ASR data. This is particularly encouraging as our main motivation for using MuMBo was to be able to spread the classification weights more equally on the different views in order to be more robust when one view fails, as this is the case in the setting *test(ASR)* where the main lexical views are affected by the high WER of the ASR transcriptions.

MuMBo	view 1	view 2	view 3	view 4	view 5
test(gold)	393	429	16	92	70
test(ASR)	393	429	16	92	70
AdaBoost	view 1	view 2	view 3	view 4	view 5
test(gold)	418	452	18	80	48
test(ASR)	418	452	18	80	48

Table 4: # of selection for each view with *MuMBo* and *Adaboost* on the DECODA corpus

Table 4 shows that MuMBo encourages the cooperation between the different views as the number of selections of the stronger views ($V1$ and $V2$) is smaller for MuMBo than for AdaBoost. This cooperation allows the classification weights to be distributed more equally among the views.

5. Conclusion

We presented in this paper an application of a multiview boosting-like algorithm to human-human conversation classification task. This multiview approach, named Mumbo, was designed in order to make use of the informations obtained from the different views defined on the learning examples and encourages the cooperation between the different views by a better distribution of the weights among the views. This redistribution allows Mumbo to perform better than AdaBoost in the considered settings. More importantly, Mumbo outperforms AdaBoost in the *test(ASR)* setting, meaning that it is more robust to noisy examples than AdaBoost. These results are encouraging, showing that multiview approaches can be applied to monomodal classification tasks, by considering different views on the data to process. These views are not necessarily independent as long as they have different strenghts and weaknesses.

6. Acknowledgements

This work is supported by the French agency ANR, Project DECODA, contract no 2009-CORD-005-01, and the French business clusters Cap Digital and SCS. For more information about the DECODA project, please visit the project home-page, `http://decoda.univ-avignon.fr`

7. References

[1] Frederic Bechet, Benjamin Maza, Nicolas Bigouroux, Thierry Bazillon, Marc El-Beze, Renato De Mori, and Eric Arbillot. DECODA: a call-centre human-human spoken conversation corpus. In *LREC 2012*, Istambul, 2012.

[2] Avrim B. Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[3] Mario C. Christoudias, Raquel Urtasun, Ashish Kapoor, and Trevor Darrell. Co-training with noisy perceptual observations. In *Conference on Computer Vision and Pattern Recognition*, page 8pp, 2009.

[4] Mark Culp, George Michailidis, and Kjell Johnson. On multi-view learning with additive models. *Annals of Applied Statistics*, 3:292–318, 2009.

[5] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet. Icsiboost. http://code.google.come/p/icsiboost, 2007.

[6] Jean-Christophe Janodet, Marc Sebban, and Henri-Maxime Suchier. Boosting classifiers built from different subsets of features. *Fundam. Inf.*, 96:89–109, January 2009.

[7] Jana Kludas, Eric Bruno, and Stéphane Marchand-Maillet. Information fusion in multimedia information retrieval. In Nozha Boujemaa, Marcin Detyniecki, and Andreas Nrnberger, editors, *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics*, volume 4918 of *Lecture Notes in Computer Science*, pages 147–159. Springer Berlin / Heidelberg, 2008.

[8] Sokol Koço and Cécile Capponi. A boosting approach to multiview classification with cooperation. In *European Conference on Machine Learning (ECML)*, pages 209–228, 2011.

[9] Indraneel Mukherjee and Robert Schapire. A theory of multiclass boosting. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1714–1722, 2010.

[10] Cees Snoek, Marcel Worring, and Arnold Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 399–402, New York, NY, USA, 2005. ACM.

[11] Karthik Sridharan and Sham M. Kakade. An information theoretic framework for multi-view learning. In *Annual Conference on Computational Learning Theory*, 2008.

[12] Michal Wozniak and Konrad Jackowski. Some remarks on chosen methods of classifier fusion based on weighted voting. In Emilio Corchado, Xindong Wu, Erkki Oja, lvaro Herrero, and Bruno Baruque, editors, *Hybrid Artificial Intelligence Systems*, volume 5572 of *Lecture Notes in Computer Science*, pages 541–548. Springer Berlin / Heidelberg, 2009.