PAC-Bayesian Majority Vote for Late Classifier Fusion[*]

Emilie Morvant   Stéphane Ayache

Aix-Marseille Univ., LIF-QARMA, CNRS, UMR 7279, F-13013, Marseille, France

{*firstname.name*}*@lif.univ-mrs.fr*

Amaury Habrard

Univ. of St-Etienne, Lab. Hubert Curien, CNRS, UMR 5516, F-42000, St-Etienne, France

amaury.habrard@univ-st-etienne.fr

July 5, 2012

Abstract

A lot of attention has been devoted to multimedia indexing over the past few years. In the literature, we often consider two kinds of fusion schemes: The *early fusion* and the *late fusion*. In this paper we focus on late classifier fusion, where one combines the scores of each modality at the decision level. To tackle this problem, we investigate a recent and elegant well-founded quadratic program named MinCq coming from the Machine Learning PAC-Bayes theory. MinCq looks for the weighted combination, over a set of real-valued functions seen as voters, leading to the lowest misclassification rate, while making use of the voters' diversity. We provide evidence that this method is naturally adapted to late fusion procedure. We propose an extension of MinCq by adding an order-preserving pairwise loss for ranking, helping to improve Mean Averaged Precision measure. We confirm the good behavior of the MinCq-based fusion approaches with experiments on a real image benchmark.

Keywords: Machine Learning, Multimedia fusion, Multi-modality search, Ranking and re-ranking

1 Introduction

Combining multimodal information is an important issue in Multimedia and a lot of research effort has been dedicated to this problem (see Atrey et al. (2010) for a survey). Indeed, the fusion of multimodal inputs can bring complementary information, from various sources, useful for improving the quality of any multimedia analysis method such as for semantic concept detection, audio-visual event detection, object tracking, etc.

The different modalities correspond generally to a relevant set of features that can be grouped into different views. For example, classical visual or textual features commonly used in multimedia are based on TF-IDF, bag of words, texture, color, SIFT, spatio-temporal descriptors, etc. Once these features have been extracted, another step consists in using machine learning methods in order to build classifiers able to discriminate a given concept.

Two main schemes are generally considered Snoek et al. (2005). In the *early fusion* approach, all the available data/features are merged into one feature vector before the learning and classification steps. This can be seen as a unimodal classification. However, this kind of approach has to deal with heterogeneous data or features which are sometimes difficult to combine. The *late fusion* model works at the decision level by combining the prediction scores available for each modality. This is usually called multimodal classification or classifier fusion. *Late fusion* may not always outperform unimodal classification. Especially when one modality provides significantly better results than others or when one has to deal with imbalanced input features. However, *late fusion* scheme tends to give better results for learning semantic concepts in case of multimodal video Snoek et al. (2005). Several methods based on a fixed decision rule have been proposed for combining classifiers such as max, min, product, sum, etc Kittler et al. (1998). Other approaches, often referred to as *stacking* Wolpert (1992), need of an extra learning step.

In this paper, we address the problem of *late* multimodal fusion at the decision level with stacking. Let h_i be the classifier that gives the score associated with the ith modality for any instance $\mathbf{x}$. A classical method consists in looking for a weighted linear combination of the different scores,

$$H(\mathbf{x}) = \sum_{i=1}^{n} q_i h_i(\mathbf{x}),$$

where q_i represents the weight associated with h_i. It is usually required that $0 \leq q_i \leq 1$ and $\sum_{i=1}^{n} q_i = 1$. This linear weighting scheme can be seen as a majority vote. This approach is widely used because of its robustness, simplicity and scalability due to small computational costs Atrey et al. (2010). It is also more appropriate when there exist dependencies between the views Wu et al. (2004). An important issue is then to find an optimal way to combine the scores. One solution is to use machine learning methods to assess the weights Atrey et al. (2010). From a machine learning standpoint considering a set of classifiers with a high diversity is generally a desirable property Dietterich (2000). One illustration is given by the algorithm AdaBoost Freund and Schapire (1996), frequently used as a multimodal fusion method. AdaBoost weights the classifiers according to different distributions of the training data, introducing some diversity, but requires at least *weak classifiers* to perform well. Another recent approach based on the portfolio theory Wang and Kankanhalli (2010) proposes a fusion procedure trying to minimize some risks over the different modalities and a correlation measure. While it is well-founded, it needs to define some appropriate functions and is not completely fully adapted to the classifier fusion problem since it does not directly take into account the diversity between the outputs of the classifiers.

We propose to study a new machine learning method, namely MinCq, introduced in Laviolette et al. (2011). It proposes a quadratic program for learning a weighted majority vote over real-valued functions called voters (such as score functions of classifiers). The algorithm is based on the minimization of a generalization bound that takes into account both the risk of committing an error and the diversity of the voters, offering strong theoretical guarantees on the learned majority vote. In this article, our aim is to show the interest of this algorithm for classifier fusion. We provide evidence that MinCq is able to find good linear weightings but also very performing non-linear combination with an extra kernel layer over the scores. Since in multimedia retrieval, the performance measure is related to the rank of positive examples, we propose to extend MinCq to improve the Mean Average Precision. We base this extension on an additional order-preserving loss for verifying ranking pairwise constraints.

The paper is organized as follows. Section 2 deals with the theoretical framework of MinCq. We extend MinCq as a late fusion method in Section 3. Before concluding in Section 5, we evaluate empirically the MinCq late fusion in Section 4.

2   PAC-Bayesian MinCQ

In this section we present the algorithm MinCq of Laviolette *et al.* Laviolette et al. (2011) for learning a $\mathcal{Q}$-weighted majority vote of real-valued functions (*e.g.* classifier scores). This method is based on the PAC-Bayes theory McAllester (1999). We first recall the setting of MinCq.

We consider binary classification tasks over a *feature space* $X \subseteq \mathbb{R}^d$ of dimension d. The *label space* is $Y = \{-1, 1\}$. The training sample is $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ where each example $(\mathbf{x}_i, y_i)$ is drawn *i.i.d.* from a fixed — but unknown — probability distribution $\mathcal{D}$ defined over $X \times Y$. We consider a space of real-valued voters $\mathcal{H}$, such that $\forall h_i \in \mathcal{H}$, $h_i : X \mapsto \mathbb{R}$. Given a voter h_i, the predicted label of $\mathbf{x} \in X$ is given by $\mathrm{sign}[h_i(\mathbf{x})]$, where $\mathrm{sign}[a] = 1$ if $a \geq 0$ and -1 otherwise. Then, the learner aims at choosing a distribution $\mathcal{Q}$ over $\mathcal{H}$ — the weights q_i — leading to the $\mathcal{Q}$-weighted majority vote $B_{\mathcal{Q}}$ with the lowest risk. $B_{\mathcal{Q}}$ is defined by,

$$B_{\mathcal{Q}}(\mathbf{x}) = \mathrm{sign}\left[H_{\mathcal{Q}}(\mathbf{x})\right],$$

$$\text{with } H_{\mathcal{Q}}(\mathbf{x}) = \sum_{i=1}^{|\mathcal{H}|} q_i h_i(\mathbf{x}).$$

The associated true risk $R_{\mathcal{D}}(B_{\mathcal{Q}})$ is defined as the probability that the majority vote misclassifies an example drawn according to $\mathcal{D}$,

$$R_{\mathcal{D}}(B_{\mathcal{Q}}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left(B_{\mathcal{Q}}(\mathbf{x}) \neq y\right).$$

In the case of MinCq, $\mathcal{H}$ has to be a finite *auto-complemented* family of $2n$ real-valued voters $\mathcal{H} = \{h_1, \dots, h_{2n}\}$ such that,

$$\forall \mathbf{x} \in X, \forall i \in \{1, \dots, n\}, \; h_{i+n}(\mathbf{x}) \;=\; -h_i(\mathbf{x}). \tag{1}$$

Moreover, the algorithm considers *quasi-uniform* distributions $\mathcal{Q}$ over $\mathcal{H}$, *i.e.* the sum of the weight of a voter and its opposite is $\frac{1}{n}$,

$$\forall i \in \{1, \dots, n\}, \; \mathcal{Q}(h_i) + \mathcal{Q}(h_{i+n}) \;=\; q_i + q_{i+1} \;=\; \frac{1}{n}. \tag{2}$$

This constraint is not too restrictive since every distribution over $\mathcal{H}$ can be represented by a quasi-uniform distribution Laviolette et al. (2011). The assumptions (1) and (2) are actually an elegant trick to avoid the use of a prior distribution over $\mathcal{H}$ which is often required by usual PAC-Bayesian method McAllester (1999), making the algorithm more easily applicable.

We now present the principle of the algorithm MinCq. The core of MinCq is the minimization of the empirical version of a bound — the C-Bound — over the risk of the $\mathcal{Q}$-weighted majority vote.

Theorem 1 (C-Bound Laviolette et al. (2011)) *Given $\mathcal{H} = \{h_1, \ldots, h_{2n}\}$ a class of $2n$ functions, for any weights $\{q_i\}_{i=1}^{2n}$, i.e. distribution $\mathcal{Q}$, on $\mathcal{H}$ and any distribution $\mathcal{D}$ over $X \times Y$, if $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \, H_{\mathcal{Q}}(\mathbf{x}) > 0$ then $R_{\mathcal{D}}(B_{\mathcal{Q}}) \leq C_{\mathcal{Q}}^{\mathcal{D}}$ where,*

$$C_{\mathcal{Q}}^{\mathcal{D}} = \frac{\mathbf{Var}_{(\mathbf{x},y)\sim\mathcal{D}}(yH_{\mathcal{Q}}(\mathbf{x}))}{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}(yH_{\mathcal{Q}}(\mathbf{x}))^2} = 1 - \frac{(\mathcal{M}_{\mathcal{Q}}^{\mathcal{D}})^2}{\mathcal{M}_{\mathcal{Q}^2}^{\mathcal{D}}},$$

with $\mathcal{M}_{\mathcal{Q}}^{\mathcal{D}} = \mathbb{E}_{(\mathbf{x},y)\sim D} \sum_{i=1}^{2n} y q_i h_i(\mathbf{x})$, and $\mathcal{M}_{\mathcal{Q}^2}^{\mathcal{D}} = \mathbb{E}_{(\mathbf{x},y)\sim D} \sum_{i=1}^{2n} \sum_{i'=1}^{2n} q_i q_{i'} h_i(\mathbf{x})h_{i'}(\mathbf{x})$ are respectively the first and the second moments of the $\mathcal{Q}$-margin: $yH_{\mathcal{Q}}(\mathbf{x})$.

Following some generalization bounds, MinCq proposes to minimize the empirical version of the C-bound, $C_{\mathcal{Q}}^S = 1 - \frac{(\mathcal{M}_{\mathcal{Q}}^S)}{\mathcal{M}_{\mathcal{Q}^2}^S}$, over a sample S. The idea is to fix the empirical first moment $\mathcal{M}_{\mathcal{Q}}^S$ to a margin $\mu > 0$ and to minimize the empirical second moment $\mathcal{M}_{\mathcal{Q}^2}^S$ measuring the correlation of the voters. This leads to minimize the bound and thus the risk of the majority vote by taking into account the diversity between the voters.

Definition 1 (MinCq algorithm Laviolette et al. (2011)) *Given a set $\mathcal{H} = \{h_1, \ldots, h_{2n}\}$ of voters, a training set $S = \{(\mathbf{x}_j, y_j)\}_{j=1}^m$, and a margin $\mu > 0$, among all quasi-uniform distributions $\mathcal{Q}$ of empirical margin $\mathcal{M}_{\mathcal{Q}}^S$ exactly equal to μ, the MinCq algorithm consists in finding one that minimizes the empirical $\mathcal{M}_{\mathcal{Q}^2}^S$.*

Due to the auto-complemented (1) and quasi-uniformity (2) assumptions, the algorithm can be expressed as a quadratic program ($MinCq$) by only considering the first n voters $h_i \in \mathcal{H}$.

$$\operatorname{argmin}_{\mathbf{Q}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q},$$

$$\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{2nm} \sum_{j=1}^m \sum_{i=1}^n y_j h_i(\mathbf{x}_j),$$

$$\text{and } \forall i \in \{1, \ldots, n\}, \, 0 \leq q_i \leq \frac{1}{n}, \qquad\qquad (MinCq)$$

where ^{t} denotes the transposed function, $\mathbf{Q} = (q_1, \ldots, q_n)^t$ is the vector of the first n weights q_i, $\mathbf{M}_S$ is the $n \times n$ matrix formed by $\frac{1}{m} \sum_{j=1}^m h_i(\mathbf{x}_j)h_{i'}(\mathbf{x}_j)$ for i and i' in $\{1, \ldots, n\}$, and,

$$\mathbf{m}_S = \left(\frac{1}{m} \sum_{j=1}^m y_j h_1(\mathbf{x}_j), \ldots, \frac{1}{m} \sum_{j=1}^m y_j h_n(\mathbf{x}_j) \right)^t,$$

$$\mathbf{A}_S = \left(\frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m h_1(\mathbf{x}_j)h_i(\mathbf{x}_j), \ldots, \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m h_n(\mathbf{x}_j)h_i(\mathbf{x}_j) \right)^t.$$

Finally, the $\mathcal{Q}$-weighted majority vote learned by MinCq is then

$$B_{\mathcal{Q}}(\mathbf{x}) = \operatorname{sign}[H_{\mathcal{Q}}(\mathbf{x})],$$

$$\text{with } H_{\mathcal{Q}}(\mathbf{x}) = \sum_{i=1}^n \left(2q_i - \frac{1}{n} \right) h_i(\mathbf{x}).$$

3　MinCq as a Late Fusion Method

PAC-Bayesian MinCq has been proposed in the particular context of binary classification where the objective is to minimize the misclassification rate of the $\mathcal{Q}$-weighted majority vote by taking into account the diversity of the voters. From a multimedia indexing standpoint, MinCq thus appears to be a natural way for late classifiers fusion to combine the predictions of classifiers separately trained from different modalities.

Concretely, given a training sample of size $2m$ we split it randomly into two subsets S' and $S = \{(\mathbf{x}_j, y_j)\}_{j=1}^m$ of the same size. Let n be the number of modalities. For each modality i, we train a classifier h_i from S'. Let

$\mathcal{H} = \{h_1, \dots, h_n, -h_i, \dots, -h_n\}$ be the set of the n associated prediction functions and their opposites. At this step, the fusion is achieved by MinCq: We learn from S the $\mathcal{Q}$-weighted majority vote over $\mathcal{H}$ with the lowest risk. However, in many applications, such as multimedia document retrieval, people are interested in performance measures related to precision or recall. Since a low-error vote is not necessarily a good ranker, we propose an adaptation of MinCq to improve the popular Mean Averaged Precision (MAP).

We first recall the definition of the MAP measured on S for a given real-valued function h. Let $S^+ = \{(\mathbf{x}_j, y_j) : (\mathbf{x}_j, y_j) \in S \wedge y_j = 1\} = \{(\mathbf{x}_{j^+}, 1)\}_{j^+=1}^{m^+}$ be the set of the m^+ positive examples from S and $S^- = \{(\mathbf{x}_j, y_j) : (\mathbf{x}_j, y_j) \in S \wedge y_j = -1\} = \{(\mathbf{x}_{j^-}, -1)\}_{j^-=1}^{m^-}$ the set of the m^- negative examples from S ($m^+ + m^- = m$). For evaluating the MAP, one ranks the examples in descending order of the scores. The MAP of h is,

$$MAP_S(h) = \frac{1}{|m^+|} \sum_{j:y_j=1} Prec@j,$$

where $Prec@j$ is the percentage of positive examples in the top j. The intuition behind this definition is that we prefer positive examples with a score higher than negative ones. To achieve this goal, we propose to learn with *pairwise preference* Fürnkranz and Hüllermeier (eds) (2010) on pairs of positive-negative instances. Indeed, pairwise methods are known to be a good compromise between accuracy and more complex performance measure like MAP. Especially, the notion of order-preserving pairwise loss was introduced in Zhang (2004) in the context of multiclass classification. Following this idea, Yue *et al.* Yue et al. (2007) have proposed a SVM-based method with a hinge-loss relaxation of a MAP-loss. In our specific case of MinCq for multimedia fusion, we design an order-preserving pairwise loss for correctly ranking the positive examples. Actually, for each pair $(\mathbf{x}_{j^+}, \mathbf{x}_{j^-}) \in S^+ \times S^-$, we want: $H_{\mathcal{Q}}(\mathbf{x}_{j^+}) > H_{\mathcal{Q}}(\mathbf{x}_{j^-}) \Leftrightarrow H_{\mathcal{Q}}(\mathbf{x}_{j^-}) - H_{\mathcal{Q}}(\mathbf{x}_{j^+}) < 0$. This can be forced by minimizing (according to the weights q_i) the following hinge-loss relaxation of the previous equation,

$$\frac{1}{m^+ m^-} \sum_{j^+=1}^{m^+} \sum_{j^-=1}^{m^-} \left[H_{\mathcal{Q}}(\mathbf{x}_{j^-}) - H_{\mathcal{Q}}(\mathbf{x}_{j^+}) \right]_+, \tag{3}$$

where $[a]_+ = \max(a, 0)$ is the hinge-loss. In the setting of MinCq, with $\mathcal{H}$ auto-complemented (Eq.(1)) and $\mathcal{Q}$ quasi-uniform (Eq.(2)), we reduce the term (3) to,

$$\frac{1}{m^+ m^-} \sum_{j^+=1}^{m^+} \sum_{j^-=1}^{m^-} \left[\sum_{i=1}^{n} \left(2q_i - \frac{1}{n} \right) \left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+}) \right) \right]_+. \tag{4}$$

To deal with the hinge-loss of (4), we consider $m^+ \times m^-$ additional *slack variables* $\boldsymbol{\xi}_{S^+ \times S^-} = (\xi_{j^+j^-})_{1 \le j^+ \le m^+, 1 \le j^- \le m^-}$ weighted by a parameter $\beta > 0$. We make a little abuse of notation to highlight the difference with $(MinCq)$: Since $\boldsymbol{\xi}_{S^+ \times S^-}$ appear only in the linear term, we simply add (4) after the $(MinCq)$ formulation. We obtain the quadratic program $(MinCq_{PW})$,

$$\operatorname{argmin}_{\mathbf{Q}, \boldsymbol{\xi}_{s^+ \times s^-}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q} + \beta \, \mathbf{Id}^t \boldsymbol{\xi}_{S^+ \times S^-},$$

$$\text{s.t.} \ \ \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{2nm} \sum_{j=1}^{m} \sum_{i=1}^{n} y_j h_i(\mathbf{x}_j),$$

$$\forall j^+ \in \{1, \dots, m^+\}, \forall j^- \in \{1, \dots, m^-\}, \ \xi_{j^+j^-} \ge 0,$$

$$\xi_{j^+j^-} \ge \frac{1}{m^+ m^-} \sum_{i=1}^{n} \left(2q_i - \frac{1}{n} \right) \left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+}) \right),$$

$$\text{and} \ \forall i \in \{1, \dots, n\}, \ 0 \le q_i \le \frac{1}{n}, \tag{$MinCq_{PW}$}$$

where $\mathbf{Id}$ is the unit vector of size $m^+ \times m^-$. However, one drawback of this method is the incorporation of a quadratic number of additive variables ($m^+ \times m^-$) which makes the problem harder to solve. To overcome this problem, we propose to relax the constraints by considering the average score of the negative examples: We force the positive examples to be higher than the average negative scores. This leads us to the following alternative

Table 1: MAP obtained on the PascalVOC'07 test sample. On the left, experiments with rbf kernel layer. On the right, without.

concept	$MinCq_{PWav}^{rbf}$	$MinCq^{rbf}$	SVM^{rbf}	$MinCq_{PWav}$	$MinCq_{PW}$	$MinCq$	Σ	Σ_{MAP}	$best$	h_{best}
aeroplane	*0.513*	*0.513*	0.497	0.487	0.486	**0.526**	0.460	0.241	0.287	0.382
bicycle	**0.273**	0.219	0.232	0.195	0.204	*0.221*	0.077	0.086	0.051	0.121
bird	0.2659	**0.264**	0.196	0.169	0.137	*0.204*	0.110	0.093	0.113	0.123
boat	**0.267**	0.242	0.240	0.1593	0.154	0.159	0.206	0.132	0.079	*0.258*
bottle	*0.103*	0.099	0.042	0.112	**0.126**	0.118	0.023	0.025	0.017	0.066
bus	**0.261**	**0.261**	0.212	*0.167*	0.166	0.168	0.161	0.098	0.089	0.116
car	**0.530**	**0.530**	0.399	*0.521*	0.465	0.495	0.227	0.161	0.208	0.214
cat	**0.253**	0.245	0.160	*0.230*	0.219	0.220	0.074	0.075	0.065	0.116
chair	**0.397**	**0.397**	0.312	*0.257*	0.193	0.230	0.242	0.129	0.178	0.227
cow	0.158	**0.177**	0.117	0.102	0.101	*0.118*	0.078	0.068	0.06	0.101
diningtable	**0.263**	0.227	0.245	0.118	0.131	0.149	*0.153*	0.091	0.093	0.124
dog	**0.261**	0.179	0.152	*0.260*	0.259	0.253	0.004	0.064	0.028	0.126
horse	**0.495**	0.4504	0.437	0.3011	0.259	0.303	*0.364*	0.195	0.141	0.221
motorbike	**0.295**	0.284	0.207	0.1412	0.113	0.162	*0.193*	0.115	0.076	0.130
person	**0.630**	0.614	0.237	*0.624*	0.617	0.604	0.001	0.053	0.037	0.246
pottedplant	0.102	**0.116**	0.065	*0.067*	0.061	0.061	0.057	0.04	0.046	0.073
sheep	**0.184**	0.175	0.144	0.0666	*0.096*	0.0695	0.128	0.062	0.064	0.083
sofa	**0.246**	0.211	0.162	0.204	*0.208*	0.201	0.137	0.087	0.108	0.147
train	**0.399**	0.385	0.397	0.331	0.332	*0.335*	0.314	0.164	0.197	0.248
tvmonitor	*0.272*	0.257	0.230	**0.281**	**0.281**	0.256	0.015	0.102	0.069	0.171
Average	**0.301**	0.292	0.234	0.240	0.231	*0.243*	0.151	0.104	0.100	0.165

problem ($MinCq_{PWav}$) with only m^+ additional variables.

$$\text{argmin}_{\mathbf{Q}, \boldsymbol{\xi}_{S+}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q} + \beta \, \mathbf{Id}^t \boldsymbol{\xi}_{S+},$$

$$\text{s.t.} \ \ \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{2nm} \sum_{j=1}^{m} \sum_{i=1}^{n} y_j h_i(\mathbf{x}_j),$$

$$\forall j^+ \in \{1, \dots, m^+\}, \ \xi_{j^+} \geq 0,$$

$$\xi_{j^+} \geq \frac{1}{m^+ m^-} \sum_{j^-=1}^{m^-} \sum_{i=1}^{n} \left(2q_i - \frac{1}{n} \right) \left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+}) \right),$$

$$\text{and } \forall i \in \{1, \dots, n\}, \ 0 \leq q_i \leq \frac{1}{n}, \qquad (MinCq_{PWav})$$

where $\mathbf{Id}$ is the unit vector of size m^+.

Note that the two approaches still respect the framework of the original MinCq. We simply regularize the search of the weights for a $\mathcal{Q}$-weighted majority vote leading to an higher MAP.

Finally, for tuning the hyperparameters (μ, β) we use a cross-validation process (CV). Instead of selecting the parameters leading to the lowest risk, we select the ones leading to the best MAP.

4   Experiments

In this section, we show empirically the interest of MinCq, and our extension, as a late fusion method with stacking (implemented with MOSEK solver). We experiment the MinCq-based approaches on the PascalVOC'07 benchmark Everingham et al. (2007), where the goal is a list of 20 visual concepts to identify in images. The corpus is constituted of 5000 training and 5000 test images. In general, the ratio between positive and negative examples is less than 10%. For each concept, we generate a training sample constituted of all the training positive examples and negative examples independently drawn such that the positive ratio is $1/3$. We keep the original test set.

Our objective is not to provide the best results on this benchmark but rather to evaluate if the MinCq-based methods could be helpful for the late fusion step in multimedia indexing. To do so, we split the training sample into two subsets, S' and S, of the same size. We consider 9 different visual features: 1 SIFT, 1 LBP, 1 Percepts, 2 HOG, 2 Local Color Histograms and 2 Color Moments. Then, we train from S' a SVM-classifier for each visual feature (with the LibSVM library Chang and Lin (2001) and a rbf kernel with parameters tuned by CV). The final classifier fusion is learned from S.

In a first series of experiments, the set of voters $\mathcal{H}$ is constituted by the 9 SVM-classifiers (MinCq also considers the opposites). We compare the 3 linear MinCq methods ($MinCq$), ($MinCq_{PW}$), ($MinCq_{PWav}$) to the following 4 baseline fusion approaches.

- The best classifier of $\mathcal{H}$:
$$h_{best} = \mathrm{argmax}_{h_i \in \mathcal{H}} \, MAP_S(h_i).$$

- The one with the highest margin:
$$best(\mathbf{x}) = \mathrm{argmax}_{h_i \in \mathcal{H}} \, |h_i(\mathbf{x})|.$$

- The sum of the classifiers (unweighted vote):
$$\Sigma(\mathbf{x}) = \sum_{h_i \in \mathcal{H}} h_i(\mathbf{x}).$$

- The MAP-weighted vote:
$$\Sigma_{MAP}(\mathbf{x}) = \sum_{h_i \in \mathcal{H}} \frac{MAP_S(h_i)}{\sum_{h_{i'} \in \mathcal{H}} MAP_S(h_{i'})} h_i(\mathbf{x}).$$

In a second series, we propose to introduce non-linear information with a rbf kernel layer. We represent each example by the vector of its scores of the 9 SVM-classifiers, $\mathcal{H}$ being the set of kernels over the sample S: Each $\mathbf{x} \in S$ is seen as a voter $k(\cdot, \mathbf{x})$. We then compare our method to stacking with SVM tuned by CV (SVM^{rbf}). Note that we do not report the results of ($MinCq_{PW}$) in this context, because the computational cost is much higher and the performance is lower. The full pairwise version implies too many variables which may penalize the resolution of ($MinCq_{PW}$).

In either case, the hyperparameters of MinCq-based methods are tuned with a grid search by a 5-folds CV. The MAP-performances are reported on Table 1, we can make the following remarks.

- On the right, for the first experiments, we clearly see that the linear MinCq-based algorithms outperform on average the linear baselines. At least one MinCq-based method produces the highest MAP, except for "boat" for which h_{best} is the best. We note that the order-preserving hinge-loss is not really helpful: The classical ($MinCq$) shows the best MAP. In fact, this can be explained by the limited number of voters.

- On the left, with a kernel layer, at least one MinCq-based method achieves the highest MAP and for $17/20$ both are better than SVM. Moreover, $MinCq_{PWav}^{rbf}$ with the averaged pairwise preference is the best for 17 concepts, showing the order-preserving loss is a good compromise between improving the MAP and keeping a reasonable computational cost.

- Globally, kernel-based MinCq methods outperform the other methods. Moreover, at least one MinCq-based approach is the best for each concept showing PAC-Bayesian MinCq is a good alternative for late classifiers fusion.

5   Conclusion

We propose in this paper to make use of a well-founded learning quadratic program called MinCq as a novel multimedia late fusion method. PAC-Bayesian MinCq was originally developed for binary classification and aims at minimizing the error rate of the weighted majority vote by considering the diversity of the voters Laviolette et al. (2011). In the context of multimedia indexing, we claim that MinCq thus appears naturally appropriate for late classifier fusion in order to combine the predictions of classifiers trained from different modalities. Our experiments show that MinCq is a very competitive alternative for classifier fusion. Moreover, the incorporation of average order-preserving constraints is sometimes able to improve the MAP-performance measure. Beyond these results, such PAC-Bayesian methods open the door to define other theoretically well-founded frameworks to design new algorithms in many multimedia tasks such as multi-modality indexing, multi-label classification, ranking, etc.

References

Atrey, P. K., Hossain, M. A., El-Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia Syst.*, 16(6):345–379.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines.*

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2007). The PASCAL visual object classes challenge 2007 (VOC2007) results.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. of ICML*, pages 148–156.

Fürnkranz, J. and Hüllermeier (eds), E. (2010). *Preference Learning*. Springer-Verlag.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. *PAMI*, 20:226–239.

Laviolette, F., Marchand, M., and Roy, J.-F. (2011). From PAC-Bayes bounds to quadratic programs for majority votes. In *Proceedings of ICML*.

McAllester, D. A. (1999). PAC-bayesian model averaging. In *Proceedings of COLT*, pages 164–170.

Snoek, C., Worring, M., and Smeulders, A. W. M. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of ACM Multimedia*, pages 399–402.

Wang, X. and Kankanhalli, M. S. (2010). Portfolio theory of multimedia fusion. In *Proceedings of ACM Multimedia*, pages 723–726.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2):241–259.

Wu, Y., Chang, E. Y., Chang, K. C.-C., and Smith, J. R. (2004). Optimal multimodal fusion for multimedia data analysis. In *Proceedings of ACM Multimedia*, pages 572–579.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of SIGIR*, pages 271–278.

Zhang, T. (2004). Statistical analysis of some multi-category large margin classification methods. *JMLR*, 5:1225–1251.