

# Majority Vote of Diverse Classifiers for Late Fusion

Emilie Morvant, Amaury Habrard, Stéphane Ayache

▶ **To cite this version:**

Emilie Morvant, Amaury Habrard, Stéphane Ayache. Majority Vote of Diverse Classifiers for Late Fusion. IAPR Joint International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recignition, Aug 2014, Joensuu, Finland. pp.20. hal-00985839v2

**HAL Id: hal-00985839**
**https://hal.science/hal-00985839v2**

Submitted on 18 Jun 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Majority Vote of Diverse Classifiers for Late Fusion

Emilie Morvant[1], Amaury Habrard[2], and Stéphane Ayache[3]

[1] Institute of Science and Technology (IST) Austria, A-3400 Klosterneuburg, Austria
[2] Université de Saint-Etienne, CNRS, LaHC, UMR 5516, F-42000 St-Etienne, France
[3] Aix-Marseille Université, CNRS, LIF UMR 7279, F-13000, Marseille, France

**Abstract.** In the past few years, a lot of attention has been devoted to multimedia indexing by fusing multimodal informations. Two kinds of fusion schemes are generally considered: The *early fusion* and the *late fusion*. 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-Bayesian theory. MinCq looks for the weighted combination, over a set of real-valued functions seen as voters, leading to the lowest misclassification rate, while maximizing the voters' diversity. We propose an extension of MinCq tailored to multimedia indexing. Our method is based on an order-preserving pairwise loss adapted to ranking that allows us to improve Mean Averaged Precision measure while taking into account the diversity of the voters that we want to fuse. We provide evidence that this method is naturally adapted to late fusion procedures and confirm the good behavior of our approach on the challenging PASCAL VOC'07 benchmark.

**Keywords:** Multimedia analysis, Classifier fusion, Majority vote, Ranking

## 1 Introduction

Combining multimodal information is an important issue in pattern recognition. Indeed, the fusion of multimodal inputs can bring complementary information from various sources, useful for improving the quality of the final decision. In this paper, we focus on multimodal fusion for image analysis in multimedia systems (see [1] for a survey). The different modalities correspond generally to a relevant set of features that can be grouped into views. Once these features have been extracted, another step consists in using machine learning methods in order to build voters—or classifiers—able to discriminate a given concept. In this context, two main schemes are generally considered [17]. On the one hand, in the *early fusion* approach, all the available 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 many heterogeneous features which are sometimes hard to combine. On the other hand, the *late fusion*[4] works at the decision level by combining the prediction scores available for each modality (see Fig. 1). Even if late fusion may not always outperform early fusion[5], it tends to give better results in multimedia analysis [17]. Several methods based on a fixed

[4] The late fusion is sometimes called multimodal classification or classifier fusion.
[5] For example, when one modality provides significantly better results than others.

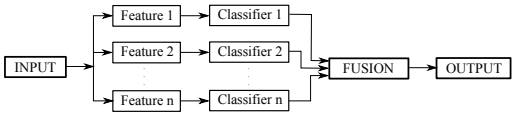



**Fig. 1.** Classical late classifier fusion scheme.

decision rule have been proposed for combining classifiers such as `max`, `min`, `sum`, etc [9]. Other approaches, often referred to as *stacking* [20], need of an extra learning step.

In this paper, we address the problem of *late fusion* with stacking. Let $h_i$ be the function that gives the score associated with the $i^{th}$ modality for any instance $\mathbf{x}$. A classical method consists in looking for a weighted linear combination of the scores seen as a majority vote and defined by: $H(\mathbf{x})=\sum_{i=1}^{n} q_i h_i(\mathbf{x})$, where $q_i$ is the weight associated with $h_i$. This approach is widely used because of its robustness, simplicity and scalability due to small computational costs [1]. It is also more appropriate when there exist dependencies between the views through the classifiers [21, 14]. The objective is then to find an optimal combination of the classifiers' scores by taking into account these dependencies. One solution is to use machine learning methods to assess automatically the weights [10, 4, 16, 18]. Indeed, from a theoretical machine learning standpoint: considering a set of classifiers with a high diversity is a desirable property [4]. One illustration is given by the algorithm AdaBoost [7] that weights *weak classifiers* according to different distributions of the training data, introducing some diversity. However, AdaBoost degrades the fusion performance when combining strong classifiers [19].

To tackle the late fusion by taking into account the diversity between score functions of strong classifiers, we propose a new framework based on a recent machine learning algorithm called MinCq [12]. MinCq is expressed as 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 usefulness of MinCq-based methods for classifier fusion. We provide evidence that they are able to find good linear weightings, and also performing non-linear combination with an extra kernel layer over the scores. Moreover, since in multimedia retrieval, the performance measure is related to the rank of positive examples, we extend MinCq to optimize 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. The framework of MinCq is introduced in Section 2. Our extension for late classifier fusion is presented in Section 3 and it is evaluated on an image annotation task in Section 4. We conclude in Section 5.

## 2 MinCq: A Quadratic Program for Majority Votes

We start from the presentation of MinCq [12], a quadratic program for learning a weighted majority vote of real-valued functions for binary classification. Note that this method is based on the machine learning PAC-Bayesian theory, first introduced in [15].

We consider binary classification tasks over a *feature space* $X \subseteq \mathbb{R}^d$ of dimension $d$. The *label space* (or output space) is $Y = \{-1, 1\}$. The training sample of size $m$ 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 set of $n$ real-valued voters $\mathcal{H}$, such that: $\forall h_i \in \mathcal{H},\ h_i : X \mapsto \mathbb{R}$. Given a voter $h_i \in \mathcal{H}$, 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 the weights $q_i$, leading to the *$\mathcal{Q}$-weighted majority vote* $B_\mathcal{Q}$ with the lowest risk. In the specific setting of MinCq[6], $B_\mathcal{Q}$ is defined by,

$$B_\mathcal{Q}(\mathbf{x}) = \mathrm{sign}\left[H_\mathcal{Q}(\mathbf{x})\right], \text{ with } H_\mathcal{Q}(\mathbf{x}) = \sum_{i=1}^{n} q_i h_i(\mathbf{x}),$$

where $\forall i \in \{1, \ldots, n\},\ \sum_{i=1}^n |q_i| = 1$ and $-1 \leq q_i \leq 1$. Its true risk $R_\mathcal{D}(B_\mathcal{Q})$ is defined as the probability that $B_\mathcal{Q}$ misclassifies an example drawn according to $\mathcal{D}$,

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

The core of MinCq is the minimization of the empirical version of a bound—the $C$-Bound [11, 12]—over $R_\mathcal{D}(B_\mathcal{Q})$. The $C$-Bound is based on the notion of $\mathcal{Q}$-margin, which is defined for every example $(\mathbf{x}, y) \sim \mathcal{D}$ by: $yH_\mathcal{Q}(\mathbf{x})$, and models the confidence on its label. Before expounding the $C$-Bound, we introduce the following notations respectively for the first moment $\mathcal{M}_\mathcal{Q}^\mathcal{D}$ and for the second moment $\mathcal{M}_{\mathcal{Q}^2}^\mathcal{D}$ of the $\mathcal{Q}$-margin,

$$\begin{aligned}
\mathcal{M}_\mathcal{Q}^\mathcal{D} &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} yH_\mathcal{Q}(\mathbf{x}) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \sum_{i=1}^{n} yq_i h_i(\mathbf{x}), \\
\mathcal{M}_{\mathcal{Q}^2}^\mathcal{D} &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \left(yH_\mathcal{Q}(\mathbf{x})\right)^2 = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \sum_{i=1}^{n}\sum_{i'=1}^{n} q_i q_{i'} h_i(\mathbf{x}) h_{i'}(\mathbf{x}). \qquad (1)
\end{aligned}$$

By definition, $B_\mathcal{Q}$ correctly classifies an example $\mathbf{x}$ if the $\mathcal{Q}$-margin is strictly positive. Thus, under the convention that if $y\mathbb{E}_{h\sim\mathcal{Q}}\, h(\mathbf{x}) = 0$, then $B_\mathcal{Q}$ errs on $(\mathbf{x}, y)$, we have:

$$\forall D \text{ over } X \times Y,\ R_\mathcal{D}(B_\mathcal{Q}) = \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim\mathcal{D}}\Big(yH_\mathcal{Q}(\mathbf{x}) \leq 0\Big).$$

Knowing this, the authors of [11, 12] have proven the following $C$-bound over $R_D(B_\mathcal{Q})$ by making use of the Cantelli-Chebitchev inequality.

**Theorem 1 (The C-bound).** *Given $\mathcal{H}$ a class of $n$ functions, for any weights $\{q_i\}_{i=1}^n$, 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,*

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

In the supervised binary classification setting, [12] have then proposed to minimize the empirical counterpart of the $C$-bound for learning a good majority vote over $\mathcal{H}$, justified by an elegant PAC-Bayesian generalization bound. Following this principle the authors

[6] In PAC-Bayes these weights are modeled by a distribution $\mathcal{Q}$ over $\mathcal{H}$ s.t. $\forall h_i \in \mathcal{H},\ q_i = \mathcal{Q}(h_i)$.

have derived the following quadratic program called MinCq.

$$\text{argmin}_{\mathbf{Q}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q}, \tag{2}$$

$$\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{nm} \sum_{j=1}^{m} \sum_{i=1}^{n} y_j h_i(\mathbf{x}_j), \tag{3}$$

$$\text{and } \forall i \in \{1, \ldots, n\},\ 0 \leq q'_i \leq \tfrac{1}{n}, \tag{4}$$

$$(MinCq)$$

where $^t$ is 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 $(i, i')$ in $\{1, \ldots, n\}^2$, $\mathbf{A}_S = \left( \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} h_1(\mathbf{x}_j) h_i(\mathbf{x}_j), \ldots, \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} h_n(\mathbf{x}_j) h_i(\mathbf{x}_j) \right)^t$, and, $\mathbf{m}_S = \left( \frac{1}{m} \sum_{j=1}^{m} y_j h_1(\mathbf{x}_j), \ldots, \frac{1}{m} \sum_{j=1}^{m} y_j h_n(\mathbf{x}_j) \right)^t$.

Finally, the majority vote learned by MinCq is $B_{\mathcal{Q}}(\mathbf{x}) = \text{sign}[H_{\mathcal{Q}}(\mathbf{x})]$, with,

$$H_{\mathcal{Q}}(\mathbf{x}) = \sum_{i=1}^{n} \underbrace{\left(2q'_i - \tfrac{1}{n}\right)}_{q_i} h_i(\mathbf{x}).$$

Concretely, MinCq minimizes the denominator of the $C$-bound (Eq. (2)), given a fixed numerator, *i.e.* a fixed $\mathcal{Q}$-margin (Eq. (3)), under a particular regularization (Eq. (4))[7]. Note that, MinCq has showed good performances for binary classification.

## 3 A New Framework for Classifier Late Fusion

MinCq stands in the particular context of machine learning binary classification. In this section, we propose to extend it for designing a new framework for multimedia late fusion. We actually consider two extensions for dealing with ranking, one with pairwise preferences and a second based on a relaxation of these pairwise preferences to lighten the process. First of all, we discuss in the next section the usefulness of MinCq in the context of multimedia late fusion.

### 3.1 Justification of MinCq as a Classifier Late Fusion Algorithm

It is well known that diversity is a key element in the success of classifier combination [1, 10, 4, 6]. From a multimedia indexing standpoint, fuzing diverse voters is thus necessary for leading to good performances. We quickly justify that this is exactly what MinCq does by favoring majority votes with maximally uncorrelated voters.

In the literature, a general definition of diversity does not exist. However, there are popular diversity metrics based on pairwise difference on every pair of individual classifiers, such as $Q$-statistics, correlation coefficient, disagreement measure, *etc.* [10, 13] We consider the following diversity measure assessing the disagreement between the predictions of a pair of voters according to the distribution $\mathcal{D}$,

$$\text{diff}_{\mathcal{D}}(h_i, h_{i'}) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} h_i(\mathbf{x}) h_{i'}(\mathbf{x}).$$

[7] For more technical details on MinCq please see [12].

We then can rewrite the second moment of the $\mathcal{Q}$-margin (see Eq.(1)),

$$\mathcal{M}^{\mathcal{D}}_{\mathcal{Q}^2} = \sum_{i=1}^{n}\sum_{i'=1}^{n} q_i q_{i'} \operatorname{diff}_{\mathcal{D}}(h_i, h_{i'}). \tag{5}$$

The first objective of MinCq is to reduce this second moment, implying a direct optimization of Eq. (5). This implies a maximization of the diversity between voters: MinCq favors maximally uncorrelated voters and appears to be a natural way for late fusion to combine the predictions of classifiers separately trained from various modalities.

### 3.2 MinCq for Ranking

In many applications, such as information retrieval, it is well known that ranking documents is a key point to help users browsing results. The traditional measures to evaluate the ranking ability of algorithms are related to precision and recall. Since a low-error vote is not necessarily a good ranker, we propose in this section an adaptation of MinCq to allow optimization of the Mean Averaged Precision (MAP) measure.

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 $\mathcal{H}=\{h_1,\ldots,h_n\}$ be the set of the $n$ associated prediction functions and their opposite. Now at this step, the fusion is achieved by MinCq: We learn from $S$ the weighted majority vote over $\mathcal{H}$ with the lowest risk.
We now 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 $h$ over $S$ 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 is that we prefer positive examples with a score higher than negative ones.

**MinCq with Pairwise Preference.** To achieve this goal, we propose to make use of *pairwise preferences* [8] 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 [23] in the context of multiclass classification. Following this idea, [22] have proposed a SVM-based method with a hinge-loss relaxation of a MAP-loss. In our specific case of MinCq for late fusion, we design an order-preserving pairwise loss for correctly ranking the positive examples. 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) the following hinge-loss relaxation of the previous equation (where $[a]_+ = \max(a, 0)$ is the hinge-loss),

$$\frac{1}{m^+m^-}\sum_{j^+=1}^{m^+}\sum_{j^-=1}^{m^-}\Big[\sum_{i=1}^{n}\underbrace{\left(2q_i - \tfrac{1}{n}\right)\left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+})\right)}_{H_{\mathcal{Q}}(\mathbf{x}_{j^-}) - H_{\mathcal{Q}}(\mathbf{x}_{j^+})}\Big]_+. \tag{6}$$

To deal with the hinge-loss of (6), we consider $m^+ \times m^-$ additional *slack variables* $\boldsymbol{\xi}_{S^+\times S^-} = (\xi_{j^+j^-})_{1\leq j^+\leq m^+, 1\leq j^-\leq m^-}$ 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 obtain the following quadratic program ($MinCq_{PW}$),

$$
\begin{aligned}
&\operatorname{argmin}_{\mathbf{Q},\boldsymbol{\xi}_{S^+\times S^-}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q} + \beta\, \mathbf{Id}^t \boldsymbol{\xi}_{S^+\times S^-},\\
&\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n} y_j h_i(\mathbf{x}_j),\\
&\forall (j^+, j^-) \in \{1,..,m^+\}\times\{1,..,m^-\},\ \xi_{j^+j^-} \geq 0,\ \xi_{j^+j^-} \geq \frac{1}{m^+m^-}\sum_{i=1}^{n}\left(2q_i' - \tfrac{1}{n}\right)\left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+})\right),\\
&\text{and } \forall i \in \{1,\ldots,n\},\ 0 \leq q_i' \leq \tfrac{1}{n}, \qquad\qquad (MinCq_{PW})
\end{aligned}
$$

where $\mathbf{Id} = (1,\ldots,1)$ 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 relax this approach as follows.

**MinCq with Average Pairwise Preference.** We relax the constraints by considering the average score over the negative examples: we force the positive ones to be higher than the average negative scores. This leads us to the following alternative problem ($MinCq_{PWav}$) with only $m^+$ additional variables.

$$
\begin{aligned}
&\operatorname{argmin}_{\mathbf{Q},\boldsymbol{\xi}_{S^+}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q} + \beta\, \mathbf{Id}^t \boldsymbol{\xi}_{S^+},\\
&\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n} y_j h_i(\mathbf{x}_j),\\
&\quad \forall j^+ \in \{1,\ldots,m^+\},\ \xi_{j^+} \geq 0,\ \xi_{j^+} \geq \frac{1}{m^+m^-}\sum_{j^-=1}^{m^-}\sum_{i=1}^{n}\left(2q_i' - \tfrac{1}{n}\right)\left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+})\right),\\
&\text{and } \forall i \in \{1,\ldots,n\},\ 0 \leq q_i' \leq \tfrac{1}{n}, \qquad\qquad (MinCq_{PWav})
\end{aligned}
$$

where $\mathbf{Id} = (1,\ldots,1)$ of size $m^+$.

Note that the two approaches stand in the original framework of MinCq. In fact, we regularize the search of the weights for majority vote leading to an higher MAP. To conclude, our extension of MinCq aims at favoring $\mathcal{Q}$-majority vote implying a good trade-off between classifiers maximally uncorrelated and leading to a relevant ranking.

## 4 Experiments on PascalVOC'07 benchmark

**Protocol.** In this section, we show empirically the usefulness of late fusion MinCq-based methods with stacking. We experiment these approaches on the PascalVOC'07 benchmark [5], where the objective is to perform the classification for 20 concepts. The corpus is constituted of $10,000$ images split into train, validation and test sets. For most of concepts, the ratio between positive and negative examples is less than $10\%$, which leads to unbalanced dataset and requires to carefully train each classifier. For simplicity reasons, we generate a training set 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. Indeed, 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. We consider $9$ different visual features, that are

| concept | $MinCq_{PWav}$ | $MinCq_{PW}$ | $MinCq$ | $\Sigma$ | $\Sigma_{MAP}$ | $best$ | $h_{best}$ |
|---|---|---|---|---|---|---|---|
| aeroplane | 0.487 | 0.486 | **0.526** | 0.460 | 0.241 | 0.287 | 0.382 |
| bicycle | 0.195 | 0.204 | **0.221** | 0.077 | 0.086 | 0.051 | 0.121 |
| bird | 0.169 | 0.137 | **0.204** | 0.110 | 0.093 | 0.113 | 0.123 |
| boat | 0.159 | 0.154 | 0.159 | 0.206 | 0.132 | 0.079 | **0.258** |
| bottle | 0.112 | **0.126** | 0.118 | 0.023 | 0.025 | 0.017 | 0.066 |
| bus | 0.167 | 0.166 | **0.168** | 0.161 | 0.098 | 0.089 | 0.116 |
| car | **0.521** | 0.465 | 0.495 | 0.227 | 0.161 | 0.208 | 0.214 |
| cat | **0.230** | 0.219 | 0.220 | 0.074 | 0.075 | 0.065 | 0.116 |
| chair | **0.257** | 0.193 | 0.230 | 0.242 | 0.129 | 0.178 | 0.227 |
| cow | 0.102 | 0.101 | **0.118** | 0.078 | 0.068 | 0.06 | 0.101 |
| diningtable | 0.118 | 0.131 | 0.149 | **0.153** | 0.091 | 0.093 | 0.124 |
| dog | **0.260** | 0.259 | 0.253 | 0.004 | 0.064 | 0.028 | 0.126 |
| horse | 0.301 | 0.259 | 0.303 | **0.364** | 0.195 | 0.141 | 0.221 |
| motorbike | 0.141 | 0.113 | 0.162 | **0.193** | 0.115 | 0.076 | 0.130 |
| person | **0.624** | 0.617 | 0.604 | 0.001 | 0.053 | 0.037 | 0.246 |
| pottedplant | **0.067** | 0.061 | 0.061 | 0.057 | 0.04 | 0.046 | 0.073 |
| sheep | 0.067 | 0.096 | 0.0695 | **0.128** | 0.062 | 0.064 | 0.083 |
| sofa | 0.204 | **0.208** | 0.201 | 0.137 | 0.087 | 0.108 | 0.147 |
| train | 0.331 | 0.332 | **0.335** | 0.314 | 0.164 | 0.197 | 0.248 |
| tvmonitor | **0.281** | **0.281** | 0.256 | 0.015 | 0.102 | 0.069 | 0.171 |
| Average | 0.240 | 0.231 | **0.243** | 0.151 | 0.104 | 0.100 | 0.165 |

**Table 1.** MAP obtained on the PascalVOC'07 test sample.

SIFT, Local Binary Pattern (LBP), Percepts, 2 Histograms Of Gradient (HOG), 2 Local Color Histograms (LCH) and 2 Color Moments (CM):

- LCH are $3\times3\times3$ histogram on a grid of $8\times6$ or $4\times3$ blocs. Color Moments are represented by the two first moments on a grid of $8\times6$ or $4\times3$ blocs.
- HOG is computed on a grid of $4\times3$ blocs. Each bin is defined as the sum of the magnitude gradients from 50 orientations. Thus, overall EDH feature has 600 dimensions. HOG feature is known to be invariant to scale and translation.
- LBP is computed on grid of $2\times2$ blocs, leading to a $1,024$ dimensional vector. The LBP operator labels the pixels of an image by thresholding the $3\times3$-neighborhood of each pixel with the center value and considering the result as a decimal number. LBP is known to be invariant to any monotonic change in gray level.
- Percept features are similar to SIFT codebook where visual words are related to semantic classes at local level. There are 15 semantic classes such as 'sky', 'skin', 'greenery', 'rock', etc. We also considered SIFT features from a dense grid, then map it on a codebook of 1000 visual words generated with Kmeans.

We train a SVM-classifier for each feature with the LibSVM library [2] and a RBF kernel with parameters tuned by cross-validation. The set $\mathcal{H}$ is then constituted by the $9$ score functions associated with the SVM-classifiers.

In a first series of experiments, the set of voters $\mathcal{H}$ is constituted by the $9$ SVM-classifiers. We compare our $3$ MinCq-based methods to the following $4$ baselines:

- The best classifier of $\mathcal{H}$: $h_{best} = \text{argmax}_{h_i\in\mathcal{H}}\, MAP_S(h_i)$.
- The one with the highest confidence: $best(\mathbf{x}) = \text{argmax}_{h_i\in\mathcal{H}}\, |h_i(\mathbf{x})|$.
- The sum of the classifiers (unweighted vote): $\Sigma(\mathbf{x}) = \sum_{h_i\in\mathcal{H}} h_i(\mathbf{x})$.

| concept | $MinCq^{rbf}_{PWav}$ | $MinCq^{rbf}$ | SVM$^{rbf}$ |
|---|---|---|---|
| aeroplane | **0.513** | **0.513** | 0.497 |
| bicycle | **0.273** | 0.219 | 0.232 |
| bird | **0.266** | 0.264 | 0.196 |
| boat | **0.267** | 0.242 | 0.240 |
| bottle | **0.103** | 0.099 | 0.042 |
| bus | **0.261** | **0.261** | 0.212 |
| car | **0.530** | **0.530** | 0.399 |
| cat | **0.253** | 0.245 | 0.160 |
| chair | **0.397** | **0.397** | 0.312 |
| cow | 0.158 | **0.177** | 0.117 |
| diningtable | **0.263** | 0.227 | 0.245 |
| dog | **0.261** | 0.179 | 0.152 |
| horse | **0.495** | 0.450 | 0.437 |
| motorbike | **0.295** | 0.284 | 0.207 |
| person | **0.630** | 0.614 | 0.237 |
| pottedplant | 0.102 | **0.116** | 0.065 |
| sheep | **0.184** | 0.175 | 0.144 |
| sofa | **0.246** | 0.211 | 0.162 |
| train | **0.399** | 0.385 | 0.397 |
| tvmonitor | **0.272** | 0.257 | 0.230 |
| Average | **0.301** | 0.292 | 0.234 |

**Table 2.** MAP obtained on the PascalVOC'07 test sample with a RBF kernel layer.

- 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 for increasing the diversity over the set $\mathcal{H}$. We consider a larger $\mathcal{H}$ as follows. Each example is represented by the vector of its scores with the 9 SVM-classifiers and $\mathcal{H}$ is now the set of kernels over the sample $S$: Each $\mathbf{x} \in S$ is seen as a voter $k(\cdot, \mathbf{x})$. We compare this approach to classical stacking with SVM.

Finally, for tuning the hyperparameters we use a 5-folds cross-validation process, where instead of selecting the parameters leading to the lowest risk, we select the ones leading to the best MAP. The MAP-performances are reported on Tab. 1 for the first series and on Tab. 2 for the second series.

**Results.** Firstly, the performance of $\Sigma_{MAP}$ fusion is lower than $\Sigma$, which means that the performance of single classifiers is not correlated linearly with its importance on the fusion step. On Tab. 1, for the first experiments, we clearly see that the linear MinCq-based algorithms outperform on average the linear baselines. MinCq-based method produces the highest MAP for 16 out of 20 concepts. Using a Student paired t-test, this result is statistically confirmed with a p-value $< 0.001$ in comparison with $\Sigma_{MAP}$, $best$ and $h_{best}$. In comparison of $\Sigma$, the p-values respectively associated with ($MinCq_{PWav}$), ($MinCq_{PW}$) and ($MinCq_{PW}$) are $0.0139$, $0.0232$ and $0.0088$. We can remark that ($MinCq_{PW}$) implies lower performances than its relaxation ($MinCq_{PWav}$). A Student test leads to a p-value of $0.223$, which statistically means that the two approaches produce similar results. Thus, when our objective is to rank the positive examples before the negative examples, the average constraints appear to be a good solution. However, we note that the order-preserving hinge-loss is not really helpful: The classical ($MinCq$) shows the best MAP results (with a p-value of $0.2574$). Indeed, the trade-off between diversity and ranking is difficult to apply here since the 9 voters are probably

not enough expressive. On the one hand, the preference constraints appear hard to satisfy, on the other hand, the voters' diversity do not really vary.
The addition of a kernel layer allows us to increase this expressivity. Indeed, Tab. 2 shows that the MinCq-based methods achieve the highest MAP for every concept in comparison with SVM classifier. This confirms that the diversity between voters is well modeled by MinCq algorithm. Especially, $MinCq^{rbf}_{PWav}$ with the averaged pairwise preference is significantly the best: a Student paired test implies a p-value of $0.0003$ when we compare $MinCq^{rbf}_{PWav}$ to SVM, and the p-value is $0.0038$ when it is compared to $MinCq^{rbf}$. Thus, the the order-preserving loss is a good compromise between improving the MAP and keeping a reasonable computational cost. Note that we do not report the results for ($MinCq_{PW}$) in this context, because the computational cost is much higher and the performance is lower. The full pairwise version implies too many variables which penalize the resolution of ($MinCq_{PW}$). Finally, it appears that at least one MinCq-based approach is the best for each concept, showing that MinCq methods outperform the other compared methods. Moreover, a Student test implies a p-value $< 0.001$ when we compare $MinCq^{rbf}_{PWav}$ to the approaches without kernel layer. $MinCq^{rbf}_{PWav}$ is significantly then the best approach in our experiments.

We conclude from these experiments that MinCq-based approaches are a good alternative for late classifiers fusion as it takes into account the diversity of the voters. In the context of multimedia documents retrieval, the diversity of the voters comes from either the variability of input features or by the variability of first layer classifiers.

## 5 Conclusion and Perspectives

In this paper, we proposed to make use of a well-founded learning quadratic program called MinCq for multimedia late fusion tasks. MinCq was originally developed for binary classification, aiming at minimizing the error rate of the weighted majority vote by considering the diversity of the voters [12]. We designed an adaptation of MinCq able to deal with ranking problems by considering pairwise preferences while taking into account the diversity of the models. In the context of multimedia indexing, this extension of MinCq appears naturally appropriate for combining the predictions of classifiers trained from various modalities in a late classifier fusion setting. Our experiments have confirmed that MinCq is a very competitive alternative for classifier fusion in the context of an image indexing task. Beyond these results, this work gives rise to many interesting remarks, among which the following ones. Taking advantage of a margin constraint for late classifier fusion may allow us to prove a new $C$-bound specific to ranking problems, and thus to derive other algorithms for classifier fusion by maximizing the diversity between the classifiers. This could be done by investigating some theoretical results using the Cantelli-Chebychev's inequality [3] as in [12]. Additionally, it might be interesting to study the impact of using other diversity metrics [10] on performances for image and video retrieval. Such an analysis would be useful for assessing a trade-off between the quality of the ranking results and the diversity of the inputs for information retrieval. Finally, another perspective, directly founded on the general PAC-Bayes theory [15], could be to take into account a prior belief on the classifiers of $\mathcal{H}$. Indeed, general PAC-Bayesian theory allows one to obtain theoretical guarantees on majority

votes with respect to the distance between the considered vote and the prior belief measured by the Kullback-Leibler divergence. The idea is then to take into account prior information for learning good majority votes for ranking problems.

**Acknowledgments.** This work was in parts funded by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036. the authors would like to thanks Thomas Peel for useful comments.

## References

1. P. K. Atrey, M. A. Hossain, A. El-Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Syst.*, 16(6):345–379, 2010.
2. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.
3. L. Devroye, L. Gyorfi, , and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer Verlag, 1996.
4. T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, 2000.
5. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge 2007 (VOC2007) results, 2007.
6. A. Fakeri-Tabrizi, M.-R. Amini, and P. Gallinari. Multiview semi-supervised ranking for automatic image annotation. In *ACM Multimedia*, pages 513–516, 2013.
7. Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proc. of ICML*, pages 148–156, 1996.
8. J. Fürnkranz and E. Hüllermeier (eds). *Preference Learning*. Springer-Verlag, 2010.
9. J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *TPAMI*, 20:226–239, 1998.
10. L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. 2004.
11. A. Lacasse, F. Laviolette, M. Marchand, P. Germain, and N. Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the gibbs classifier. In *NIPS*, 2006.
12. F. Laviolette, M. Marchand, and J.-F. Roy. From PAC-Bayes bounds to quadratic programs for majority votes. In *ICML*, 2011.
13. D. Leonard, D. Lillis, L. Zhang, F. Toolan, R. W. Collier, and J. Dunnion. Applying machine learning diversity metrics to data fusion in information retrieval. In *ECIR*, 2011.
14. A. J. Ma, P. C. Yuen, and J.-H. Lai. Linear dependency modeling for classifier fusion and feature combination. *TPAMI*, 35(5):1135–1148, 2013.
15. D. A. McAllester. PAC-bayesian model averaging. In *COLT*, pages 164–170, 1999.
16. M. Re and G. Valentini. Ensemble methods: a review. *Advances in machine learning and data mining for astronomy*, pages 563–582, 2012.
17. C. Snoek, M. Worring, and A. W. M. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*, pages 399–402, 2005.
18. S. Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.
19. J. Wickramaratna, S. Holden, and B. Buxton. Performance degradation in boosting. In *Multiple Classifier Systems*, volume 2096 of *LNCS*, pages 11–21. Springer, 2001.
20. D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.
21. Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith. Optimal multimodal fusion for multimedia data analysis. In *ACM Multimedia*, pages 572–579, 2004.
22. Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR*, pages 271–278, 2007.
23. T. Zhang. Statistical analysis of some multi-category large margin classification methods. *JMLR*, 5:1225–1251, 2004.