Easy Samples First: Self-paced Reranking for Zero-Example Multimedia Search

Lu Jiang[1], Deyu Meng[2], Teruko Mitamura[1], Alexander G. Hauptmann[1]
[1] School of Computer Science, Carnegie Mellon University
[2] School of Mathematics and Statistics, Xi'an Jiaotong University
{lujiang, teruko, alex}@cs.cmu.edu, dymeng@mail.xjtu.edu.cn

ABSTRACT

Reranking has been a focal technique in multimedia retrieval due to its efficacy in improving initial retrieval results. Current reranking methods, however, mainly rely on the heuristic weighting. In this paper, we propose a novel reranking approach called Self-Paced Reranking (SPaR) for multimodal data. As its name suggests, SPaR utilizes samples from easy to more complex ones in a self-paced fashion. SPaR is special in that it has a concise mathematical objective to optimize and useful properties that can be theoretically verified. It on one hand offers a unified framework providing theoretical justifications for current reranking methods, and on the other hand generates a spectrum of new reranking schemes. This paper also advances the state-of-the-art self-paced learning research which potentially benefits applications in other fields. Experimental results validate the efficacy and the efficiency of the proposed method on both image and video search tasks. Notably, SPaR achieves by far the best result on the challenging TRECVID multimedia event search task.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Measurement, Experimentation

Keywords

Multimodal Reranking; Multimedia Event Detection; Self-paced Learning; Zero-Example Search; Content-based Search

1. INTRODUCTION

In the era where multimedia contents are being produced and shared in an unprecedented pace, multimedia search has become increasingly crucial in providing quality service for users to issue semantic queries, e.g. searching for visual objects or events. Reranking is a focal technique to improve the quality of search results [3]. The intuition is that the initial ranked result brought by the query is noisy [33], and can be refined by the multimodal information residing in the retrieved videos/images. For example, in image search the reranking is performed based on the results of text-based search, in which the initial results are retrieved by matching images' surrounding texts [17, 39]. Studies show that reranking methods can improve the Mean Average Precision (MAP) of the initial result by a relative 17% on a representative dataset [32, 26].

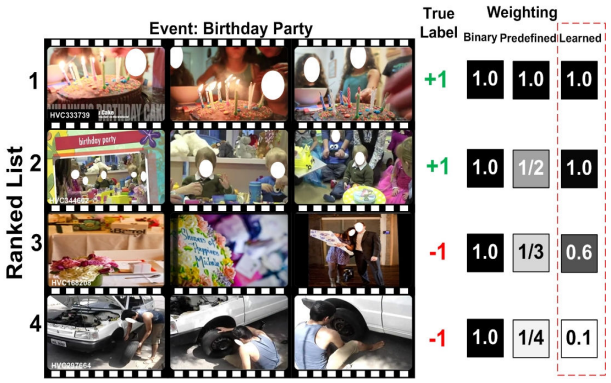

Figure 1: Comparison of binary, predefined and learned weights on the event "Birthday Party". All videos are used as positive in reranking. Learned weights are learned by the proposed method.

This observation is confirmed by a recent study on multimodal content-based search [13], in which the reranking is performed on multimodal content features extracted from the video content, and no textual metadata, such as the title, is available. Reranking by multimodal content-based search is still an understudied problem. To advance the new technologies in this direction, NIST initiated a task called Multimedia Event Detection (MED) 0Ex (Zero-Example) [7, 13, 4] in TRECVID 2013. The task is to detect the occurrence of a main event occurring in a video clip, e.g. "Birthday party", in the absence of example videos, which resembles a real-world search scenario where example videos are often unavailable. It is more challenging than reranking by text-based search in image search, since the content features not only come from multiple modalities but also much more noisy. Nevertheless, the method in [13] still manages to yield a 158% relative MAP improvement over the plain retrieval result. Due to the significant improvement, NIST includes reranking as an official condition in TRECVID 2014[1].

[1]See FullSysPRF `www-nlpir.nist.gov/projects/tv2014/tv2014.html`

MM'14, November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2654918.

An important step in reranking is to assign weights to videos, based on which the reranking is carried out. The main strategy in current reranking methods is to assign binary (or predefined) weights to videos at different rank positions. These weighting schemes are simple to implement, yet may lead to suboptimal solutions, as demonstrated in our experiments. For example, the reranking methods in [38, 13, 8] assume that top-ranked videos are of equal importance (binary weights). The fact is that, however, videos ranked higher are generally more accurate, and thus more "important", than those ranked lower. On the other hand, the predefined weights [9] are derived independently of reranking models, and thus may not faithfully reflect the latent importance. Fig. 1 illustrates a ranked list of "Birthday Party" videos, where all videos are used as positive in reranking; the top two are true positive; the third video is a negative but closely related video on wedding shower as they share key concepts such as "gift", "cake" and "cheering"; the fourth video is completely unrelated. As illustrated, neither binary nor predefined weights reflect the latent importance residing in the videos. Furthermore, since the binary and predefined weights are designed based on empirical experience lacking of theoretical justifications, it is unclear where, or even whether reranking with these weights converges.

An ideal reranking method would assign appropriate weights to videos in a theoretically sound manner. To this end, we propose a method called Self-Paced Reranking (SPaR) which assigns weights adaptively in a self-paced fashion. The method is established on the self-paced learning theory [1, 19]. The theory is inspired by the learning process of humans and animals, where samples are not presented randomly but organized in a meaningful order which illustrates from easy to gradually more complex ones [1]. In the context of the reranking problem, the easy samples are the top-ranked videos that have smaller loss. As opposed to utilizing all samples to learn a model simultaneously, the proposed model is learned gradually from easy to more complex samples. As the name "self-paced" suggests, in every iteration, SPaR examines the "easiness" of each sample based on what it has already learned, and adaptively determines their weights to be used in the subsequent iterations.

SPaR represents a general method of addressing multimodal pseudo relevance feedback for 0Ex video search. Compared with existing reranking methods, SPaR has the following three benefits. First, it is established on a solid theory, and of useful properties that can be theoretically verified. For example, SPaR has a concise mathematical objective to optimize, and its convergence property can be theoretically proved in Theorem 1. Besides, since the self-paced learning is expected to effectively converge to minima of quality [1, 19], SPaR inherently tends to find reasonably good solutions for reranking problems. The experimental results in Section 5 substantiate this hypothesis. Second, SPaR represents a general framework for reranking on multimodal data, which includes other methods, such as [38, 23, 13], as special cases (See Section 4.2). The connection is significant because once an existing method is modeled as a special case of SPaR, the optimization methods discussed in this paper become immediately applicable to analyze, and even solve the problem. Third, SPaR offers a compelling insight into reranking by multimodal content-based search [7, 13, 4], where the initial ranked lists are retrieved by content-based search. Although reranking may not be a novel idea, reranking by multimodal content-based search is clearly understudied and worthy of exploration, since existing studies mainly concentrate on reranking only by text-based search.

On the other hand, this paper also advances the state-of-the-art self-paced learning frontier. Existing self-paced learning algorithms adopt a simple binary weighting [1, 19] in order to obtain the global optimum for subproblems. We generalize it to real-valued weighting by introducing a spectrum of self-paced functions that preserve the global optimum (See Lemma 1 in Section 4.1). The proposed functions not only augment the choices of weighting schemes in reranking, but also may benefit applications in other fields that need to discriminate samples' importance.

The experimental results show promising results on two challenging datasets. On the TRECVID content-based search task MED 0Ex, SPaR improves the plain retrieval baseline by a relative 230% in terms of the MAP. To the best of our knowledge, the reported MAP is by far the best result on this TRECVID task. In addition, SPaR also outperforms the state-of-the-art reranking methods on an image reranking dataset called Web Query. In summary, the contribution of this paper is fourfold:

- We propose a novel reranking method that has a solid theoretical background and theoretically verifiable properties.
- The proposed method provides a general reranking framework by multimodal content-based search, which includes existing methods as special cases.
- We discuss a spectrum of self-paced functions that substantially augment the choices of self-paced learning. Our work is potentially beneficial to self-paced learning applications that need to discriminate samples.
- The proposed method achieves by far the best result on the TRECVID multimedia event search.

2. RELATED WORK

Existing reranking methods are mainly performed by text-based search, in which the initial ranked list is retrieved by text/keyword matching [39, 33]. In terms of the types of the reranking model, these methods can be categorized into Classification, Clustering, Graph and LETOR (LEarning-TO-Rank) based reranking. In Classification-based reranking [38], a classifier is trained upon the pseudo label set, and then tested on retrieved videos to obtain a reranked list. Similarly, in LETOR-based reranking [6] instead of a binary classifier, a ranking function is learned by the pairwise [23] or list-wise [33, 32] RankSVM. In Clustering-based reranking [9], the retrieved videos are aggregated into clusters, and the clusters' conditional probabilities of the pseudo samples are used to obtain a reranked list. The role of clustering is to reduce the noise in the initial reranking. In Graph-based reranking [10, 27], the graph of retrieved samples needs to be first constructed, on which the initial ranking scores are propagated by collective classification methods such as Random Walk [12], under the assumption that visually similar videos usually have similar ranks. Generally, reranking methods, including the above methods, are unsupervised methods, but there also exist some studies on supervised reranking [17, 39].

Reranking by multimodal content-based search is still an understudied problem. To advance the new technologies in this direction, NIST initiated a content-based search task

1: $t = 0$; //Iteration zero
2: Choose starting values for $\mathbf{y}, \mathbf{v}$;
3: **while** $t \leq$ max iteration **do**
4: $\quad \Theta_1^{(t+1)}, ..., \Theta_m^{(t+1)} = \arg\max \mathbb{E}_{\mathbf{y},\mathbf{v}}(\Theta_1^{(t)}, ..., \Theta_m^{(t)}; C)$;
5: $\quad \mathbf{y}^{(t+1)}, \mathbf{v}^{(t+1)} = \arg\max \mathbb{E}_{\Theta}(\mathbf{y}^{(t)}, \mathbf{v}^{(t)}; k)$;
6: $\quad$ **if** t is small **then** increase $1/k$;
7: **end while**
8: **return** $[v_1 y_1, \cdots, v_n y_n]^T$;

Algorithm 1: Reranking in Optimization Perspective.

1: $t = 0$; //Iteration zero
2: Choose the initial pseudo labels and weights;
3: **while** $t \leq$ max iteration **do**
4: $\quad$ Train a reranking model on the fixed labels and weights;
5: $\quad$ Update the pseudo labels and weights;
6: $\quad$ **if** t is small **then** add more pseudo positives;
7: **end while**
8: **return** The list of samples after reranking;

Algorithm 2: Reranking in Conventional Perspective.

called MED 0Ex (Zero-Example) in TRECVID 2013. Previously, research is mainly concentrated on the event search with 10 or 100 exemplar videos [24, 40, 12, 25]. Only a few methods have been proposed to tackle the 0Ex search problem [7, 13, 4]. A closely related work is [13], in which the authors discussed a reranking method named MMPRF, which yields significant improvements over the plain retrieval result. As we see in Section 4.2, MMPRF is a special case of the proposed method that only uses the binary weighting.

Self-paced (or curriculum) learning [19, 1] is a recently proposed theory, inspired by the teaching of students, where easy concepts are taught before complex ones. The idea is to learn the model gradually from easy to complex examples iteratively in a self-paced fashion. This theory has been successfully applied to various applications, including domain adaptation [30], dictionary learning [31], segmentation [20], tracking [29], etc. Existing methods all adopt a simple binary weighting to obtain the global optimum for subproblems. We generalize it to real-valued weighting that still preserves the global optimum, which is potentially conducive for self-paced learning applications in other fields.

3. SELF-PACED RERANKING

3.1 Objective Function

The proposed Self-paced Reranking is a general reranking framework for multimedia search, which is inherently performed using features from multiple modalities. Given a dataset of n videos with features extracted from m modalities, let $\mathbf{x}_{ij}$ denote the feature of the i^{th} sample from the j^{th} modalities, e.g., $\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{x}_{i3}$ can be the visual, acoustic and optical character feature vectors extracted from different channels of the i^{th} video. $y_i \in \{-1, 1\}$ is the *pseudo label* for the i^{th} video whose values are assumed since the true labels are unknown to reranking methods. The kernel SVM is used to illustrate the algorithm due to its robustness and decent performance in reranking [21, 8]. We will discuss how to generalize it to other models in Section 4.2. Let $\Theta_j = \{\mathbf{w}_j, b_j\}$ denote the classifier parameters for the j^{th} modality, which includes a coefficient vector $\mathbf{w}_j$ and a bias term b_j. Let $\mathbf{v} = [v_1, ..., v_n]^T$ denote the weighting parameters for all samples. Inspired by the self-paced learning [19], supposed n is the total number of samples and m is the total number of modalities, the objective function $\mathbb{E}$ can be formulated as:

$$\begin{aligned} &\min_{\Theta_1,...,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1, ..., \Theta_m, \mathbf{v}, \mathbf{y}; C, k) = \\ &\min_{\substack{\mathbf{y},\mathbf{v},\mathbf{w}_1,...,\mathbf{w}_m,\\ b_1,...,b_m,\{\ell_{ij}\}}} \sum_{j=1}^{m} \frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i \ell_{ij} + mf(\mathbf{v}; k) \\ &\text{s.t. } \forall i, \forall j, y_i(\mathbf{w}_j^T \phi(\mathbf{x}_{ij}) + b_j) \geq 1 - \ell_{ij}, \ell_{ij} \geq 0 \\ &\mathbf{y} \in \{-1, +1\}^n, \mathbf{v} \in [0, 1]^n, \end{aligned} \tag{1}$$

where ℓ_{ij} is the standard hinge loss, calculated from:

$$\ell_{ij} = \max\{0, 1 - y_i \cdot (\mathbf{w}_j^T \phi(\mathbf{x}_{ij}) + b_j)\}. \tag{2}$$

$\phi(\cdot)$ is a feature mapping function to obtain non-linear decision boundaries. C ($C > 0$) is the standard regularization parameter trading off the hinge loss and the margin. $\sum_{j=1}^{m} v_i \ell_{ij}$ represents the weighted loss of the i^{th} sample. The weight v_i reflects the sample's importance, and when $v_i = 0$, the loss incurred by the i^{th} sample is always zero, i.e. it will be unselected in the training. In the conventional SVM, all samples share the equal weight 1.

$f(\mathbf{v}; k)$ is a regularization term that specifies how the samples are selected and how their weights are calculated. It is called the self-paced function as it determines the learning pace, controlled by the parameter k ($k > 0$), at which the model learns new samples. There is an m in front of $f(\mathbf{v}; k)$ as $\sum_{j=1}^{m} f(\mathbf{v}; k) = mf(\mathbf{v}; k)$. $f(\mathbf{v}; k)$ can be defined in various forms in terms of the learning pace, which will be discussed in Section 3.3. The objective is subjected to two sets of constraints: the first set of constraints in Eq. (1) is the soft margin constraint inherited from the conventional SVM. The second constraints in Eq. (1) define the domains of pseudo labels and their weights, respectively.

Eq. (1) turns out to be difficult to optimize directly due to its non-convexity and complicated constraints. However, it can be effectively optimized by Cyclic Coordinate Method (CCM) [5]. CCM is an iterative method for non-convex optimization, in which the variables are divided into a set of disjoint blocks, in this case two blocks, i.e. classifier parameters $\Theta_1, ..., \Theta_m$, and pseudo labels $\mathbf{y}$ and weights $\mathbf{v}$. In each iteration, a block of variables can be optimized while keeping the other block fixed. Suppose $\mathbb{E}_\Theta$ represents the objective with the fixed block $\Theta_1, ..., \Theta_m$, and $\mathbb{E}_{\mathbf{y},\mathbf{v}}$ represents the objective with the fixed block $\mathbf{y}$ and $\mathbf{v}$. Eq. (1) can be solved by Alg. 1, which takes the input of the initial ranked list, and outputs the reranked list. In Step 2, it initializes the starting values for the pseudo labels and weights. Then it optimizes Eq. (1) iteratively via Step 4 and 5, until the convergence is reached.

Alg. 1 provides a theoretical justification for reranking from the perspective of optimization. Alg. 2 lists general steps for reranking that have one-to-one correspondence with the steps in Alg. 1. The two algorithms present the same methodology from two perspectives. For example, optimizing $\Theta_1, ..., \Theta_m$ can be interpreted as training a reranking model. In the first few iterations, Alg. 1 gradually increases the $1/k$ to control the learning pace, which, correspondingly, translates to adding more pseudo positives, e.g. top 10 [13] in training reranking models.

Alg. 1 and 2 offer complementary insights. Alg. 1 theoretically justifies Alg. 2 on the convergence and the decrease of objective. On the other hand, the empirical experience from studying Alg. 2 offers valuable advices on how to set starting

values from the initial ranked lists, which is less concerned in the optimization perspective. According to Alg. 2, to use SPaR one needs to alternate between two steps: training reranking models and determining the pseudo samples and their weights for the subsequent iteration. We will discuss how to optimize $\mathbb{E}_{\mathbf{y},\mathbf{v}}$ (training reranking models on pseudo samples) in Section 3.2, and how to optimize $\mathbb{E}_{\Theta}$ (selecting pseudo samples and their weights based on the current reranking model) in Section 3.2.

3.2 Learning with the Fixed Pseudo Labels and Weights

With the fixed $\mathbf{y}, \mathbf{v}$, Eq. (1) represents the sum of weighted hinge loss across all modalities, i.e,

$$\begin{aligned}&\min_{\Theta_1,...,\Theta_m} \mathbb{E}_{\mathbf{y},\mathbf{v}}(\Theta_1,...,\Theta_m;C)\\ =&\min_{\mathbf{w}_1,...,\mathbf{w}_m,b_1,...,b_m,\{\ell_{ij}\}} \sum_{j=1}^{m}\frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i\ell_{ij} \qquad (3)\\ &\text{s.t. } \forall i,\forall j, y_i(\mathbf{w}_j^T\phi(\mathbf{x}_{ij})+b_j)\geq 1-\ell_{ij}, \ell_{ij}\geq 0.\end{aligned}$$

As mentioned, $v_i\ell_{ij}$ is the discounted hinge loss of the i^{th} sample from the j^{th} modality. Eq. (3) represents a non-conventional SVM as each sample is associated with a weight reflecting its importance. Eq. (3) is non-trivial to optimize directly due to its complex constraints. As a result, we introduce a method that finds the global optimum for Eq. (3). The objective of Eq. (3) can be decoupled, and each modality can be optimized independently. Now consider the j^{th} modality ($j = 1, ..., m$). We introduce Lagrange multipliers λ and α, and define the Lagrangian of the problem as:

$$\begin{aligned}&\Lambda(\mathbf{w}_j, b_j, \alpha, \lambda) = \frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n} v_i\ell_{ij}\\ &+\sum_{i=1}^{n}\alpha_{ij}(1-\ell_{ij}-y_i\mathbf{w}_j^T\phi(\mathbf{x}_{ij})-y_ib_j)+\sum_{i=1}^{n}-\lambda_{ij}\ell_{ij} \qquad (4)\\ &\text{s.t. } \forall i, \alpha_{ij}\geq 0, \lambda_{ij}\geq 0.\end{aligned}$$

Since only the j^{th} modality is considered, j is a fixed constant. The Slater's condition trivially holds for the Lagrangian, and thus the duality gap vanishes at the optimal solution. According to the KKT conditions [2], the following conditions must hold for its optimal solution:

$$\begin{aligned}&\frac{\nabla\Lambda}{\mathbf{w}_j} = \mathbf{w}_j - \sum_{i=1}^{n}\alpha_{ij}y_i\phi(\mathbf{x}_{ij}) = \mathbf{0}, \frac{\nabla\Lambda}{b_j} = \sum_{i=1}^{n}\alpha_{ij}y_i = 0,\\ &\forall i, \frac{\partial\Lambda}{\partial\ell_{ij}} = Cv_i - \alpha_{ij} - \lambda_{ij} = 0.\end{aligned} \qquad (5)$$

According to Eq. (5), $\forall i, \lambda_{ij} = Cv_i - \alpha_{ij}$, and since Lagrange multipliers are nonnegative, we have $0 \leq \alpha_{ij} \leq Cv_i$. Substitute these inequations and Eq. (5) back into Eq. (4), the problem's dual form can be obtained by:

$$\begin{aligned}&\max_{\alpha}\sum_{i=1}^{n}\alpha_{ij} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_{ij}\alpha_{kj}y_iy_k\kappa(\mathbf{x}_{ij},\mathbf{x}_{kj}),\\ &\text{s.t. } \sum_{i=1}^{n} y_i\alpha_{ij} = 0, 0\leq\alpha_{ij}\leq Cv_i,\end{aligned} \qquad (6)$$

where $\kappa(\mathbf{x}_{ij},\mathbf{x}_{kj}) = \phi(\mathbf{x}_{ij})^T\phi(\mathbf{x}_{kj})$ is the kernel function. Compared with the dual form of conventional SVMs, Eq. (6) imposes a sample-specific upper-bound on the support vector coefficient, which is the key in computing decision boundary. A sample's upper-bound is proportional to its weight, and therefore a sample with a smaller weight v_i is less influential as its support vector coefficient is bounded by a small value of Cv_i. Eq. (6) degenerates to the dual form of conventional SVMs when $\mathbf{v} = \mathbf{1}$. According to the Slater's condition, strong duality holds, and therefore Eq. (4) and Eq. (6) are equivalent problems. Since Eq. (6) is a quadratic programming problem in its dual form, there exists a plethora of algorithms/toolkits to solve it [2].

3.3 Learning with the Fixed Classification Parameters

With the fixed classification parameters $\Theta_1, ..., \Theta_m$, the objective function of Eq. (1) becomes:

$$\begin{aligned}&\min_{\mathbf{y},\mathbf{v}}\mathbb{E}_{\Theta}(\mathbf{y},\mathbf{v};k) = \min_{\mathbf{y},\mathbf{v}} C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i\ell_{ij} + mf(\mathbf{v};k)\\ &\text{s.t. } \mathbf{y}\in\{-1,+1\}^n, \mathbf{v}\in[0,1]^n.\end{aligned} \qquad (7)$$

The goal of Eq. (7) is to learn not only the pseudo labels $\mathbf{y}$ but also their weights $\mathbf{v}$. Learning $\mathbf{y}$ is easier as its optimal values are independent of $\mathbf{v}$. Therefore, we first optimize each pseudo label by:

$$y_i^* = \underset{y_i=\{+1,-1\}}{\arg\min}\ \mathbb{E}_{\Theta}(\mathbf{y},\mathbf{v}) = \underset{y_i=\{+1,-1\}}{\arg\min}\ C\sum_{j=1}^{m}\ell_{ij}, \qquad (8)$$

where y_i^* denotes the optimum for the i^{th} pseudo label. Solving Eq. (8) is simple as all labels are independent with each others in the sum, and each label can only take binary values. Its global optimum can be efficiently obtained by enumerating each y_i. For n samples, we only need to enumerate $2n$ times.

Having found the optimal $\mathbf{y}$, the task switches to optimizing $\mathbf{v}$. $f(\mathbf{v};k)$ is the self-paced function, and in [19], it is defined based on the l_1 norm of $\mathbf{v}\in[0,1]^n$:

$$f(\mathbf{v};k) = -\frac{1}{k}\|\mathbf{v}\|_1 = -\frac{1}{k}\sum_{i=1}^{n} v_i. \qquad (9)$$

Substituting Eq. (9) back into Eq. (7), the optimal $\mathbf{v}^* = [v_1^*, ..., v_n^*]^T$ is then calculated from

$$v_i^* = \begin{cases}1 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} < \frac{1}{k}\\ 0 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} \geq \frac{1}{k}.\end{cases} \qquad (10)$$

The underlying intuition of the self-paced learning can be justified by the closed-form solution in Eq. (10). If a sample's average loss is less than a certain threshold, $1/k$ in this case, it will be selected, or otherwise unselected, as a training example. The parameter k controls the number of samples to be included in training. Physically, $1/k$ corresponds to the "age" of the model. When $1/k$ is small, only easy samples with small loss will be considered. As $1/k$ grows, more samples with larger loss will be gradually appended to train a "mature" reranking model.

According to Eq. (10), the variable $\mathbf{v}$ takes only binary values. This scheme is called Hard Weighting as a sample can be either selected ($v_i = 1$) or unselected ($v_i = 0$). Hard Weighting is less appropriate in our problem as it cannot discriminate the importance of samples (see Fig. 2). Correspondingly, Soft Weighting, which assigns real-valued weights, reflects the latent importance of samples in training more faithfully. The comparison is analogous to the hard/soft assignment in Bag-of-Words quantization, where an interest point can be assigned either to its closest cluster

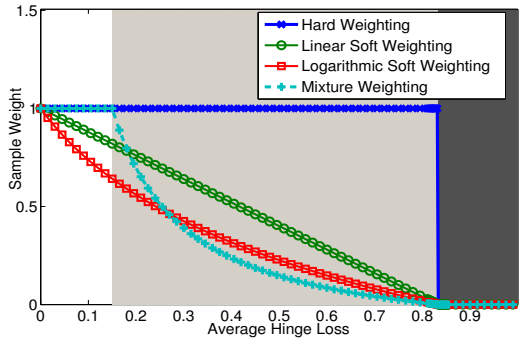

Figure 2: Comparison of different weighting schemes ($k = 1.2$, $k' = 6.7$). Hard Weighting assigns binary weights. The figure is divided into 3 colored regions, i.e. "white", "gray" and "black" in terms of the loss.

(hard), or to a number of clusters in its vicinity (soft). Generally, the soft assignment outperforms the hard assignment [15]. Before introducing specific soft weighting schemes, we first examine the general property of the self-paced function.

DEFINITION 1 (SELF-PACED FUNCTION). *Suppose that v denotes a weight variable, l is the loss, and k is the learning pace parameter. $f(v;k)$ is called a self-paced function, if*

1. *$f(v;k)$ is convex with respect to $v \in [0,1]$.*
2. *$v^*(k,l)$ is monotonically decreasing with respect to l, and it holds that $\lim_{l\to 0} v^*(k,l) = 1, \lim_{l\to\infty} v^*(k,l) = 0$.*
3. *$v^*(k,l)$ is monotonically increasing with respect to $1/k$, and it holds that $\lim_{k\to 0} v^*(k,l) = 1, \lim_{k\to\infty} v^*(k,l) = 0$.*

where $v^*(k,l) = \arg\min_{v\in[0,1]} vl + f(v;k)$.

According to [19], the self-paced function can be decomposed into $f(\mathbf{v};k) = \sum_{i=1}^n f(v_i;k)$. The three conditions in Definition 1 provide an axiom for self-paced learning. Condition 2 indicates that the model inclines to select easy samples (with smaller errors) in favor of complex samples (with larger errors). Condition 3 states that when the model "age" $1/k$ gets larger, it embarks on incorporating more, probably complex, samples to train a "mature" model. The limits in these constraints impose the upper bound and lower bound for v. The convexity in Condition 1 further ensures the model can find good solutions.

It is easy to verify that existing functions in self-paced learning [19, 30] follow Definition 1. Lemma 1 in Section 4.1 indicates the global optimum of Eq. (7) can be obtained for all functions following Definition 1. Attributed to this concise definition, a spectrum of rational self-paced learning functions can be formulated, and we discuss three of them, namely, linear, logarithmic and mixture weighting[2]. Note that the proposed functions may not be optimal as there is no single weighting scheme that can always work the best for all datasets. However, they evidently augment the choices of the soft weighting in the literature.

Linear soft weighting: Probably the most common approach is to linearly weight samples with respect to their loss. This weighting can be realized by the following self-paced function:

$$f(\mathbf{v};k) = \frac{1}{k}(\frac{1}{2}\|\mathbf{v}\|_2^2 - \sum_{i=1}^n v_i). \tag{11}$$

[2]Exponential weighting is missing in our discussion as it violates the convexity condition in Definition 1.

Eq. (11) is a convex function of $\mathbf{v}$, and thus the global minimum can be obtained at $\nabla_{\mathbf{v}}\mathbb{E}_{\Theta}(\mathbf{v}) = \mathbf{0}$. We have

$$\frac{\partial \mathbb{E}_\Theta}{\partial v_i} = \sum_{j=1}^m C\ell_{ij} + m(\frac{1}{k}v_i - \frac{1}{k}) = 0. \tag{12}$$

Considering $v_i \in [0,1]$, the close-formed optimal solution for v_i $(i = 1, 2, ..., n)$ can be written as:

$$v_i^* = \begin{cases} -k(\frac{1}{m}\sum_{j=1}^m C\ell_{ij}) + 1 & \frac{1}{m}\sum_{i=1}^m C\ell_{ij} < \frac{1}{k} \\ 0 & \frac{1}{m}\sum_{i=1}^m C\ell_{ij} \geq \frac{1}{k}. \end{cases} \tag{13}$$

Similar as the hard weighting in Eq. (10), the weight is 0 for the samples whose average loss is larger than $1/k$; Otherwise, the weight is linear to the loss (see Fig. 2).

Logarithmic soft weighting: The linear soft weighting penalizes the weight linearly in terms of the loss. A more conservative approach is to penalize the weight logarithmically, which can be achieved by the following function:

$$f(\mathbf{v};k) = \sum_{i=1}^n (\zeta v_i - \frac{\zeta^{v_i}}{\log\zeta}), \tag{14}$$

where $\zeta = (k-1)/k$ and $k > 1$. Its partial gradient equals:

$$\frac{\partial \mathbb{E}_\Theta}{\partial v_i} = \sum_{j=1}^m C\ell_{ij} + m(\zeta - \zeta^{v_i}) = 0. \tag{15}$$

We then can easily deduce:

$$\log(\frac{1}{m}\sum_{j=1}^m C\ell_{ij} + \zeta) = v_i \log\zeta. \tag{16}$$

The closed-form optimal solution for the Logarithmic soft weighting is then given by:

$$v_i^* = \begin{cases} \frac{1}{\log\zeta}\log(\frac{1}{m}\sum_{j=1}^m C\ell_{ij} + \zeta) & \frac{1}{m}\sum_{i=1}^m C\ell_{ij} < \frac{1}{k} \\ 0 & \frac{1}{m}\sum_{i=1}^m C\ell_{ij} \geq \frac{1}{k}. \end{cases} \tag{17}$$

Mixture weighting: Mixture weighting is a hybrid of the soft and the hard weighting. One can imagine that the loss range is divided into three colored areas, as illustrated in Fig. 2. If the loss is either too small ("white" area) or too large ("black" area), the hard weighting is applied. Otherwise, for the loss in the "gray" area, the soft weighting is applied. Compared with the soft weighting scheme, the mixture weighting tolerates small errors up to a certain point. To define the start of the "gray" area, an additional parameter k' is introduced. Formally,

$$f(\mathbf{v};k,k') = -\zeta\sum_{i=1}^n \log(v_i + \zeta k), \tag{18}$$

where $\zeta = \frac{1}{k'-k}$ and $k' > k > 0$. The partial gradient is:

$$\frac{\partial \mathbb{E}_\Theta}{\partial v_i} = \sum_{j=1}^m C\ell_{ij} - \frac{m\zeta}{v_i + k\zeta} = 0. \tag{19}$$

The closed-form optimal solution is given by:

$$v_i^* = \begin{cases} 1 & \frac{1}{m}\sum_{i=1}^m C\ell_{ij} \leq \frac{1}{k'} \\ 0 & \frac{1}{m}\sum_{i=1}^m C\ell_{ij} \geq \frac{1}{k} \\ \frac{m\zeta}{\sum_{i=1}^m C\ell_{ij}} - k\zeta & \text{otherwise.} \end{cases} \tag{20}$$

Eq. (20) tolerates any loss lower than $1/k'$ by assigning the full weight. It penalizes the weight by the inverse of the loss

for samples in the "gray" area which starts from $1/k'$ and ends at $1/k$ (see Fig. 2). The mixture weighting has the properties of both hard and soft weighting schemes. The comparison of these weighting schemes is listed in the toy example below.

EXAMPLE 1. *Suppose we are given six samples from two modalities. The hinge loss of each sample calculated by Eq.* (2) *is listed in the following table, where $Loss_1$ and $Loss_2$ column list the losses w.r.t. the first and the second modality, whereas "Avg Loss" column lists the average loss. The last four columns present the weights calculated by Eq.* (10)*, Eq.* (13)*, Eq.* (17) *and Eq.* (20) *where $k = 1.2$ and $k' = 6.7$.*

ID	$Loss_1$	$Loss_2$	*Avg Loss*	*Hard*	*Linear*	*Log*	*Mixture*
1	*0.08*	*0.02*	*0.05*	*1*	*0.940*	*0.853*	*1.000*
2	*0.15*	*0.09*	*0.12*	*1*	*0.856*	*0.697*	*1.000*
3	*0.50*	*0.50*	*0.50*	*1*	*0.400*	*0.226*	*0.146*
4	*0.96*	*0.70*	*0.83*	*1*	*0.004*	*0.002*	*0.001*
5	*0.66*	*1.02*	*0.84*	*0*	*0.000*	*0.000*	*0.000*
6	*1.30*	*1.10*	*1.20*	*0*	*0.000*	*0.000*	*0.000*

As we see, Hard Weighting produces less reasonable solutions, e.g. the difference between the first (ID=1) and the fourth sample (ID=4) is 0.78 and they share the same weight 1; on the contrary, the difference between the fourth and the fifth sample is only 0.01, but suddenly they have totally different weights. This abrupt change is absent in other weighting schemes. Log is a more prudent scheme than Linear as it diminishes the weight more rapidly. Among all weighting schemes, Mixture is the only one that tolerates small errors.

3.4 Modality Weighting

The accuracies of different modalities usually vary considerably, and modality weighting may be beneficial for reranking, as suggested in [13, 11]. To incorporate modality weighting, following [13], we rewrite the objective function as:

$$\begin{aligned}\min_{\mathbf{y},\mathbf{v}} \mathbb{E}_\Theta(\mathbf{y},\mathbf{v};k) = \min_{\mathbf{y},\mathbf{v}} C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i \ell_{ij} + mf(\mathbf{v};k) \\ \text{s.t. } \mathbf{y}\in\{-1,+1\}^n, \mathbf{v}\in[0,1]^n \\ \mathbf{A}^T\mathbf{v} \le \mathbf{g}.\end{aligned} \tag{21}$$

Eq. (21) degenerates to Eq. (7) when $\mathbf{g} = [n, n, .., n]^T$. The added constraint limits the total weight that each modality can have. For n samples from m modalities, according to [13], $\mathbf{A}_{n\times m}$ is a matrix calculated from:

$$\mathbf{A}_{ij} = \frac{I(\mathbf{w}_j^T\phi(\mathbf{x}_{ij}) + b_j > 0)}{\sum_{j=1}^{m} I(\mathbf{w}_j^T\phi(\mathbf{x}_{ij}) + b_j > 0)}, \tag{22}$$

where $I(\cdot)$ is the indicator function which equals 1 when $\mathbf{w}_j^T\phi(\mathbf{x}_{ij}) + b_j > 0$, and 0 otherwise. For convenience of notation, let $0/0 = 0$. For each sample, Eq. (22) first counts the number of modalities that classify it as positive. Then each row of $\mathbf{A}$ is normalized so that it adds up to 1. According to [13], $\mathbf{A}$ needs to be calculated only once. $\mathbf{A}^T\mathbf{v}$ sums up the total weight for each modality. $\mathbf{g}$ ($\mathbf{g} > \mathbf{0}$) is a prior vector imposing the upper-bound on the total weight, and higher weight should be assigned to accurate modalities [13][3]. Due to the introduced constraint, the analytical solutions in Section 3.3 cannot be directly applied to optimize Eq. (21). To solve it, we first calculate $\mathbf{y}$ using Eq. (8) and then apply gradient descent to find $\mathbf{v}$. Lemma 1 guarantees that this method finds the global optimal solution for Eq. (21).

[3]Estimating $\mathbf{g}$ is beyond the topic of our paper. More information on this topic can be found in [13].

4. THEORETICAL DISCUSSIONS

4.1 Convergence

The proposed SPaR has some useful properties. The following lemma proves that the global optimum can be obtained for any self-paced function following Definition 1.

LEMMA 1. *For any self-paced function that follows Definition 1, the gradient descent method in Section 3.4 finds the global optimal solution for Eq.* (7) *and Eq.* (21)*.*

PROOF. Consider the objective of Eq. (21). Suppose $\mathbf{y}^* = [y_1^*, ..., y_n^*]^T$ is a solution found by the gradient descent method in Section 3.4. According to Eq. (8), $\forall y_i \in \{-1, +1\}$ and $\forall v_i \in [0, 1]$, we have:

$$\mathbb{E}_\Theta(y_i^*, v_i; k) \le \mathbb{E}_\Theta(y_i, v_i; k). \tag{23}$$

Therefore $\forall\mathbf{y}$, $\forall\mathbf{v}$, the following inequations hold:

$$\mathbb{E}_\Theta(\mathbf{y}^*,\mathbf{v};k) = \sum_{i=1}^{n}\mathbb{E}_\Theta(y_i^*, v_i; k) \le \sum_{i=1}^{n}\mathbb{E}_\Theta(y_i, v_i; k) = \mathbb{E}_\Theta(\mathbf{y},\mathbf{v};k). \tag{24}$$

In other words, $\mathbf{y}^*$ found by Eq. (8) is the global optimum for Eq. (21). Now consider the objective with the fixed $\mathbf{y}^*$. According to Definition 1, $f(\mathbf{v})$ is a convex function of $\mathbf{v}$, and because $\mathbf{A}^T\mathbf{v} \le \mathbf{g}$ is a linear constraint, Eq. (21) is a convex function of $\mathbf{v}$. Suppose that $\mathbf{v}^*$ is a solution found by gradient descent, due to the convexity, $\mathbf{v}^*$ is the global optimum for Eq. (21). Therefore, $\mathbf{y}^*, \mathbf{v}^*$ is the global optimal solution for Eq. (21). The proof trivially holds for Eq. (7) when $\mathbf{g} = [n, n, .., n]^T$. □

The following theorem proves the algorithmic convergence.

THEOREM 1. *Alg. 1 converges to a stationary solution for any fixed C and k.*

PROOF. Let the superscript index the variable value in that iteration, e.g. $\mathbf{v}^{(t)}$ represents the value of $\mathbf{v}$ in the t^{th} iteration. Denote $\Theta^{(t)} = \Theta_1^{(t)}, ..., \Theta_m^{(t)}$. $\mathbf{y}^{(0)}$ and $\mathbf{v}^{(0)}$ are arbitrary initial values in their feasible regions. As Eq. (6) is a quadratic programming problem, the solution $\Theta^{(t)}$ is the global optimum for $\mathbb{E}_{\mathbf{y},\mathbf{v}}$, i.e.

$$\mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}) \le \mathbb{E}(\Theta^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}). \tag{25}$$

According to Lemma 1, $\mathbf{v}, \mathbf{y}$ are also global optimum for $\mathbb{E}_\Theta$, i.e.

$$\mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t)}, \mathbf{v}^{(t)}) \le \mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}). \tag{26}$$

Substitute Eq. (26) back into Eq. (25), we have that $\forall t \ge 1$,

$$\mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t)}, \mathbf{v}^{(t)}) \le \mathbb{E}(\Theta^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}). \tag{27}$$

Eq. (27) indicates that the objective decreases in every iteration. Since $\mathbb{E}$ is the sum of finite elements, it is bounded from below. Consequently, according to [34], it is guaranteed that Alg. 1 (an instance of CCM algorithm) converges to a stationary solution of the problem. □

Theorem 1 guarantees the convergence of Alg. 1. Note that the convergence means a stationary solution whose values are finally stable, which may not necessarily mean a global optimal solution. Actually, since the objective function of Eq. (1) is non-convex, the quality of the solution may rely on its the initial ranked lists. If initial lists brought by queries are off-topic, reranking can degrade the initial result. This is a common problem in all reranking methods that has been observed in several studies [32, 39, 13]. There exist some works to alleviate the influence of bad initial values. One is to estimate the quality of a ranked list, and only conduct reranking on "good" ranked lists [32]. Another is to provide some supervision by annotating a few positive videos, also known as supervised reranking [17, 39]. More details can be referred to these papers.

4.2 Relation to Existing Reranking Models

A general form of Eq. (1) is written as

$$\min_{\Theta_1,...,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,...,\Theta_m,\mathbf{v},\mathbf{y};k) = \min_{\Theta_1,...,\Theta_m,\mathbf{y},\mathbf{v}} \sum_{i=1}^{n}\sum_{j=1}^{m} v_i Loss(\mathbf{x}_{ij};\Theta_j) + mf(\mathbf{v};k) \quad (28)$$
$$\text{s.t. Constraints on } \Theta_1,...,\Theta_m$$
$$\mathbf{y} \in \{-1,+1\}^n, \mathbf{v} \in [0,1]^n,$$

where $Loss(\mathbf{x}_{ij};\Theta_j)$ is a general function of the loss incurred by the i^{th} sample against the j^{th} modality, e.g., in Eq. (1) it is defined as the sum of the hinge loss and the margin. The constraints on $\Theta_1,...,\Theta_m$ are the constants in the specific reranking model. Alg. 1 is still applicable to solve Eq. (28).

Eq. (28) represents a general reranking framework for 0Ex video search, which includes existing Classification-based and LETOR-based reranking methods as special cases. For example, generally, when *Loss* takes the negative likelihood of Logistic Regression, and $f(\mathbf{v};k)$ takes Eq. (9) (hard weighting), SPaR corresponds to MMPRF in [13]. When *Loss* is the hinge loss, $f(\mathbf{v};k)$ is Eq. (9), the pseudo labels are assumed to be $+1$, and there is only one modality, SPaR corresponds to Classification-based PRF [38, 8]. Given *Loss* and constraints on Θ are from pair-wise RankSVM, SPaR can degenerate to LETOR-based reranking methods [23].

4.3 Time Complexity

The proposed SPaR is efficient. Suppose n and m is the total number of samples and modalities, respectively. u is the average feature dimension. l is the number of pseudo samples selected by the self-paced function. In an iteration, the complexity mainly comes from Step 4 and Step 5 in Alg. 1. In Step 4, the complexity lies in solving the quadratic programming (order $mu \cdot l^2$). In Step 5, it lies in searching the optimal $\mathbf{y}$ (order $mu \cdot n$) and the optimal $\mathbf{v}$ (order $mu \cdot n + l^2$), based on an efficient algorithms which first solves an unconstrained problem on n variables, and then solve a constrained problem on l variables. Therefore, the time complexity for each iteration is order $mu \cdot (l^2 + n)$. As in other reranking methods, the number of pseudo samples is far less than that of total samples, i.e. $l \ll n$. The proposed method is expected to be efficient, and the empirical runtime comparison substantiates this claim.

5. EMPIRICAL EXPERIMENTS

In this section, we empirically verify the efficacy and efficiency of the proposed method on two datasets. The TRECVID MED represents the task of video reranking by multimodal content-based search, and the Web Query represents the task of image reranking by text-based search.

5.1 TRECVID Multimedia Event Detection

Dataset and evaluation: We conduct experiments on the TRECVID Multimedia Event Detection (MED) 2013 development and MEDTest set including around 34,000 videos on 20 Pre-Specified events. The task is to detect the occurrence of a main event in a video clip, where all features have to be extracted from the video content. The performance is evaluated on the MEDTest set consisting of about 25,000 videos, by the official metric Mean Average Precision (MAP). The official test split released by NIST is used, and the reported MAP is comparable with other teams' MAPs on the same split. The experiments are all conducted in the 0Ex setting, in which no ground-truth positive videos are available. In the baseline comparison, we evaluate each experiment 10 times on randomly generated splits to reduce the bias brought by the partition. The mean and 90% confidence interval are reported.

Features: Four types of high-level features are used, namely Automatic Speech Recognition (ASR), Optical Character Recognition (OCR), Semantic INdexing (SIN) and DCNN (Deep Convolutional Neural Network). SIN and DCNN [18] is 346 visual concepts and 1,000 visual objects trained on TRECVID and ImageNet set, respectively. Two types of low-level features are used: dense trajectories [36] and MFCC. The detailed information about these features is in [21].

Baselines: The proposed method is compared against the following baselines: 1) *Without Reranking* is a plain retrieval method without Reranking, and the language model with Jelinek-Mercer smoothing is used [41]. 2) *Rocchio* is a classical reranking model for vector space model under tf-idf representation [16]. 3) *Relevance Model* is a famous reranking method for text, and the variant with the i.i.d. assumption in [22] is used. 4) *CPRF*(Classification-based PRF) is a seminal PRF-based reranking method. Following [38, 8], SVM classifiers with χ^2 are trained using the top-ranked and bottom-ranked videos [38]. 5) *Learning to Rank* is a LETOR-based method. Following [23], it is trained using the pairwise constraints derived from the pseudo-positives and pseudo-negatives. A LambdaMART [37] in the RankLib toolkit is used to train the RankSVM model; 6) *MMPRF* is a multimodal reranking method [13], and the variant with modality weighting is used [13]. The parameters of all methods, including the proposed SPaR, are tuned on MED Research Set that shares no overlap with our development set. The tuned parameters are then applied to the MEDTest set.

The baseline methods are selected based on two considerations: first, the methods cover the reranking methods in the fields of both information retrieval and multimedia retrieval. Second, the comparison between them helps to isolate the contribution of different components.

Predefined weighting schemes: Section 3.3 discusses four weighting schemes including the conventional hard weighting and the proposed three soft weighting schemes. The following two predefined schemes are also included for comparison: 1) Interpolation is a commonly used weighting scheme which assigns weights linearly to a sample' rank order [9, 33]:

$$v_i = \frac{1}{m}\sum_{j=1}^{m}(1 - \frac{rank(\mathbf{x}_{ij})}{N}), \quad (29)$$

where N is the number of total pseudo samples. The weight for the first sample is 1.0, and 0.0 for the last. $rank(\cdot)$ returns the sample's rank order in its list. 2) Inverse Rank assigns a sample's weight based on its inverse rank order. The weight v_i equals the average inverse rank across m modalities:

$$v_i = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{rank(\mathbf{x}_{ij})}. \quad (30)$$

Our model: Alg. 1 is used. Eq. (6) is solved by the quadratic programming package "quadprog" [28], in which the parameter C is fixed to 1 and the ϕ is set as the χ^2 explicit feature map [35]. By default, Eq. (18) is used. The modality weighting is used. $\mathbf{g}$ vector is calculated using

Table 1: MAP (× 100) comparison with the baseline methods across 20 Pre-Specified events.

Method	NIST's split	10 splits
Without Reranking	3.9	4.9 ± 1.6
Rocchio	5.7	7.4 ± 2.2
Relevance Model	2.6	3.4 ± 1.0
CPRF	6.4	8.3 ± 1.8
Learning to Rank	3.4	4.2 ± 1.4
MMPRF	10.1	13.6 ± 2.4
SPaR	**12.9**	**15.3 ± 2.6**

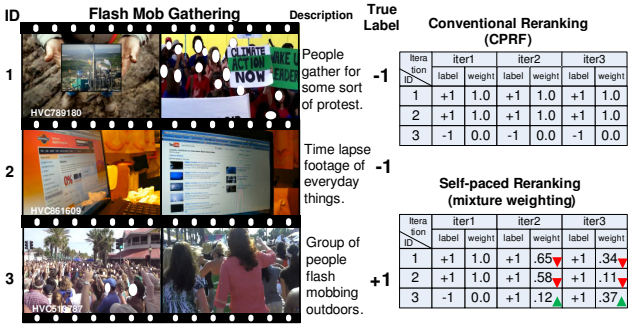

Figure 4: Weights changed by CPRF and SPaR on representative videos in different iterations.

the query likelihood method in [13]. The initial values of the pseudo labels and weights are derived using the method in [13]. Since, according to [13], pseudo negatives has little impact on the MAP, the learning is applied on pseudo positives. Eq. (21) is solved by LM-BFGS [42] in "stats" package. As in the baseline methods, the parameters are tuned on MED Research Set.

5.1.1 Comparison with Baseline methods

We first examine the overall MAP in Table 1, in which the best result is highlighted. The MAP of the proposed SPaR is by far the best MAP of the 0Ex task reported on this dataset, according to [13, 7, 4]. As we see, SPaR outperforms all baseline methods by statistically significant differences, on both the official NIST's split and the 10 splits. For example, on the NIST's split, it increases the MAP of the baseline without reranking by a relative 230% (absolute 9%), and the second best method MMPRF by a relative 28% (absolute 2.8%). Fig. 3 plots the AP comparison on each event, where the x-axis represents the event ID and the y-axis denotes the average precision. As we see, SPaR outperforms the baseline without reranking on 18 out of 20 events, and the second best MMPRF on 15 out of 20 events. The improvement is statistically significant at the p-level of 0.05, according to the paired t-test. Fig. 6 illustrates the top retrieved results on two events that have the highest improvement. As we see, the videos retrieved by SPaR are more accurate and visually coherent.

We observed two reasons for the improvements over the conventional reranking methods. First, SPaR can adjust the weights in a more reasonable way. For example, Fig. 4 illustrates the weights assigned by CPRF and SPaR on the event "E008 Flash Mob Gathering". Three representative videos are plotted where the third (ID=3) is true positive, and the others (ID=1,2) are negative. The tables on the right of Fig. 4 list their pseudo labels and weights in each iteration. Since the true labels are unknown to the methods, in the first iteration, both methods made mistakes. In Conventional Reranking, the initial pseudo labels and learned weights stay unchanged thereafter. However, SPaR adaptively assigns the weights as the iteration grows, e.g. it reduces the overestimated weights of videos (ID=1,2) in iteration 2 and 3 probably because of their dissimilarity from other pseudo positive videos. Second, modality weighting seems to be conducive in further improving the MAP. Table 2 lists the MAP with and without modality weighting [13], where the MAPs of MMPRF come from [13]. As we see, in both methods, the variant with modality weighting outperforms the variant without it.

Table 2: MAP (×100) comparison of methods with(w) and without(w/o) modality weighting on the NIST's split.

Method	w/o Weighting	w/ Weighting
MMPRF [13]	9.0	10.1
SPaR	10.8	12.9

Table 3: MAP and MAP@100 comparison with baseline methods on the Web Query dataset.

Method	MAP	MAP@100
Without Reranking [17]	0.569	0.431
CPRF [38]	0.658	-
Random Walk [10]	0.616	-
Bayesian Reranking [33, 32]	0.658	0.529
Preference Learning Model [32]	-	0.534
BVLS [26]	0.670	-
Query-Relative(visual) [17]	0.649	-
Supervised Reranking [39]	0.665	-
SPaR	**0.672**	**0.557**

We found two scenarios where SPaR fails. First, when the initial top-ranked videos retrieved by queries are completely off-topic. SPaR may not recover from the inferior starting values, e.g. the query brought by "E022 Cleaning an appliance" are off-topic (on cooking in kitchen). Second, SPaR may not help when the features used in reranking are not discriminative to the queries, e.g. for "E025 Marriage Proposal", our system lacks of meaningful detectors such as "stand on knees". Therefore even if 10 true positives are used, the AP is still bad (0.003).

5.1.2 Comparison of Weighting Schemes

We conduct experiments with different weighting schemes, and plot their MAPs in Fig. 5, where the x-axis denotes the iteration, and the y-axis is the MAP. The same step size is used in all methods. As we see, SPaR with the proposed soft weighting schemes, including linear, log and mixture weighting, consistently outperforms the binary and the predefined weighting across iterations. Among them, the mixture weighting is slightly better than others, suggesting the rationale for tolerating small errors on this dataset. The MAPs of the proposed soft weighting schemes seem to be robust and less sensitive to the iteration change, which is valuable in ranking since the number of true positives is usually unknown to the algorithms. The MAP drop seems to be related to the nature of the MED dataset. Evidence is that the similar pattern can be observed in other methods [13]. Nevertheless, SPaR still outperforms the binary, predefined weights and the baseline methods in Table 1.

5.2 Web Query Dataset

To verify SPaR's performance on image search, we conduct experiments on a web image query dataset consisting of 71,478 images from 353 queries, retrieved by a search engine named Exalead (`http://www.exalead.com/search/`). For each query, the top ranked images generated by Exalead

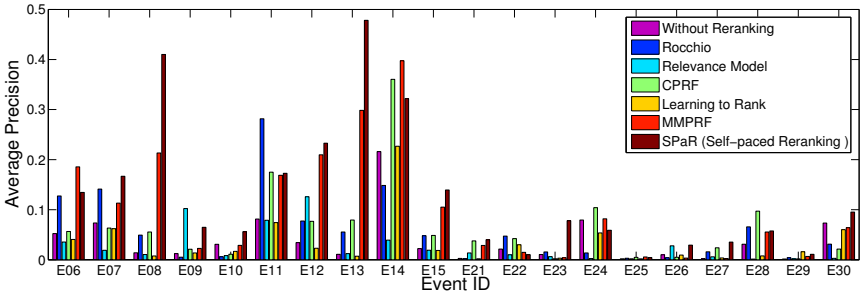

Figure 3: The AP comparison with the baseline methods. The MAP across all events is available in Table 1.

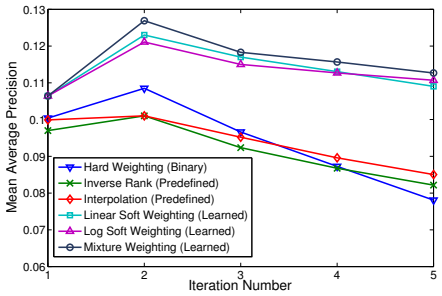

Figure 5: Comparison of binary, predefined, and learned weighting schemes in different iterations.

are provided, along with the true label for every image. The dataset is representative as the 353 queries cover a broad range of topics. The performance is evaluated by the non-interpolated MAP, as used in [17]. MAP@100 is also included for comparison. Note that as the initial result contains a single modality, no modality weighting is applied.

Following [32, 26], densely sampled SIFT are extracted. A codebook of 1,024 centroids is constructed. Spatial Tiling [14] is used to further improve the performance. We compare SPaR with the state-of-the-art reranking methods. SPaR is configured in a similar way as discussed in Section 5.1, and provided initial text-based search results are used. Following [26, 32], the parameters are tuned on a validation set consisting of a subset of queries.

We examine the overall MAP in Table 3. "-" denotes that the number is unavailable in the cited paper. As we see, SPaR achieves the promising MAP among state-of-the-art reranking methods, including Graph-based [10], LETOR-based [33, 32], Classification-based [38] and even supervised reranking methods [17, 39], in terms of both MAP and MAP@100. A similar pattern as in TRECVID MED can be observed that SPaR significantly boosts the MAP of plain retrieval without reranking, and obtain comparable or even better performance than the baseline methods. Generally, SPaR improves about 84% queries over the method without reranking. Since the initial ranked lists are retrieved by text matching, this result substantiates the claim that SPaR is general and applicable to reranking by text-based search.

5.3 Runtime Comparison

To empirically verify the efficiency of SPaR, we compare the runtime (second/query) in a single iteration. The experiments are conducted on Intel Xeon E5649 @ 2.53GHz with 16GB memory and the results are listed in Table 4. To test the speed of Rocchio and Relevance Model, we built our own inverted index on the Web Query dataset, and issue the query against the index. The reranking in MED, which is

Table 4: Runtime Comparison in a single iteration.

Method	MED	Web Query
Rocchio	5.3 (s)	2.0 (s)
Relevance Model	7.2 (s)	2.5 (s)
Learning to Rank	178 (s)	22.3 (s)
CPRF	145 (s)	10.1 (s)
MMPRF	149 (s)	10.1 (s)
SPaR	158 (s)	12.2 (s)

conducted only on semantic features, is slower because it involves multiple features and modalities. As we see, SPaR's overhead over CPRF is marginal on the both sets. This result suggests SPaR is inexpensive, and agrees with the time complexity analysis in Section 4.3. The current implementations for all methods are far from optimal, which involve a number of programming languages. We will accelerate the pipeline in future.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel approach called Self-Paced Reranking (SPaR) for multimodal reranking. SPaR reveals the missing link between reranking and an optimization problem that can be effectively solved by the self-paced learning. The proposed framework is general, and can be used to theoretically explain other reranking methods. This paper also advances the state-of-the-art self-paced learning research by generalizing binary to real-valued self-paced functions that preserve global optimum. Experimental results validate the efficacy and the efficiency of the proposed method on image and video search. SPaR consistently outperforms the plain retrieval without reranking, and obtains decent improvements over the existing reranking methods. Notably, SPaR achieves by far the best result on the challenging TRECVID MED 0Ex task.

Possible directions for future work may include automatically selecting appropriate self-paced functions for different types of reranking problems. Currently, parameters are tuned on a validation set. The tuning may heavily rely on the quality of the validation set. Another direction is to study the parameter tuning based on prior knowledge or heuristics.

7. ACKNOWLEDGMENTS

This work was partially supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. Deyu Meng was partially supported by 973 Program of China (3202013CB329404) and the NSFC project (61373114). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or

Figure 6: Top ranked videos/images ordered left-to-right using (a) plain retrieval without reranking and (b) self-paced reranking. True/false labels are marked in the lower-right of every frame.

endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

8. REFERENCES

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

[2] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[3] S.-F. Chang. How far we've come: Impact of 20 years of multimedia information retrieval. *TOMCCAP*, 9(1):42, 2013.

[4] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *ICMR*, page 1, 2014.

[5] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

[6] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *PAMI*, 30(8):1371–1384, 2008.

[7] A. Habibian, T. Mensink, and C. G. Snoek. Composite concept discovery for zero-shot video event detection. In *ICMR*, page 17, 2014.

[8] A. G. Hauptmann, M. G. Christel, and R. Yan. Video retrieval based on semantic concepts. *Proceedings of the IEEE*, 96(4):602–622, 2008.

[9] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *Multimedia*, pages 35–44, 2006.

[10] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *Multimedia*, pages 971–980, 2007.

[11] B. Huurnink, C. G. Snoek, M. de Rijke, and A. W. Smeulders. Content-based analysis improves audiovisual archive retrieval. *Multimedia, IEEE Transactions on*, 14(4):1166–1178, 2012.

[12] L. Jiang, A. G. Hauptmann, and G. Xiang. Leveraging high-level and low-level features for multimedia event detection. In *Multimedia*, pages 449–458, 2012.

[13] L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR*, page 297, 2014.

[14] L. Jiang, W. Tong, D. Meng, and A. G. Hauptmann. Towards efficient learning of optimal spatial bag-of-words representations. In *ICMR*, page 121, 2014.

[15] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR*, pages 494–501, 2007.

[16] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, DTIC Document, 1996.

[17] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web image search results using query-relative classifiers. In *CVPR*, pages 1094–1101, 2010.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[19] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, pages 1189–1197, 2010.

[20] M. P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *ICCV*, pages 1800–1807, 2011.

[21] Z.-Z. Lan, L. Jiang, S.-I. Yu, et al. Cmu-informedia at trecvid 2013 multimedia event detection. In *TRECVID*, 2013.

[22] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR*, pages 120–127, 2001.

[23] Y. Liu, T. Mei, X.-S. Hua, J. Tang, X. Wu, and S. Li. Learning to video search rerank via pseudo preference feedback. In *ICME*, pages 297–300, 2008.

[24] Z. Ma, Y. Yang, Z. Xu, N. Sebe, and A. G. Hauptmann. We are not equally negative: fine-grained labeling for multimedia event detection. In *Multimedia*, pages 293–302, 2013.

[25] M. Mazloom, A. Habibian, and C. G. Snoek. Querying for video events by semantic signatures from few examples. In *Multimedia*, pages 609–612, 2013.

[26] N. Morioka and J. Wang. Robust visual reranking via sparsity and ranking constraints. In *Multimedia*, pages 533–542, 2011.

[27] L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua. Harvesting visual concepts for image search with complex queries. In *Multimedia*, pages 59–68, 2012.

[28] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[29] J. S. Supančič III and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, pages 2379–2386, 2013.

[30] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting weights: Adapting object detectors from image to video. In *NIPS*, volume 1, page 2, 2012.

[31] Y. Tang, Y.-B. Yang, and Y. Gao. Self-paced dictionary learning for image classification. In *Multimedia*, pages 833–836, 2012.

[32] X. Tian, Y. Lu, L. Yang, and Q. Tian. Learning to judge image search results. In *Multimedia*, pages 363–372, 2011.

[33] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X.-S. Hua. Bayesian video search reranking. In *Multimedia*, pages 131–140, 2008.

[34] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[35] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *PAMI*, 34(3):480–492, 2012.

[36] H. Wang, A. Klaser, C. Schmid, and C. L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[37] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[38] R. Yan, A. G. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *CVIR*, pages 238–247, 2003.

[39] L. Yang and A. Hanjalic. Supervised reranking for web image search. In *Multimedia*, pages 183–192, 2010.

[40] Q. Yu, J. Liu, H. Cheng, A. Divakaran, and H. Sawhney. Semantic pooling for complex event detection. In *Multimedia*, pages 733–736, 2013.

[41] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342, 2001.

[42] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.