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

People

CMU Informedia Team

Deyu Meng

Teruko Mitamura

Alexander G. Hauptmann

Acknowledgement

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

Outline

- Background
- Related Work
- Self-Paced Reranking (SPaR)
- Experiment Results
- Conclusions

Outline

- **Background**
- Related Work
- Self-Paced Reranking (SPaR)
- Experiment Results
- Conclusions

Zero-Example Search

- Zero-Example Search (also known as 0Ex) represents a multimedia search condition where zero relevant examples are provided.
- An example: TRECVID Multimedia Event Detection (MED). The task is very challenging.
  - Detect every-day event in Internet videos
    - Birthday Party
    - Changing a vehicle tire
    - Wedding ceremony
  - Content-based search. **No textual metadata (title/description) is available**.

Zero-Example Search

- Event of Interest: Birthday Party

ELAMP
Prototype System

👁

👂

OCR DOC

Zero-Example Search

- Event of Interest: Birthday Party

ELAMP
Prototype System

👁 Birthday Cake, Kids, Gift

👂

OCR DOC

Zero-Example Search

- Event of Interest: Birthday Party

ELAMP
Prototype System

👁 **Birthday Cake, Kids, Gift**

👂 **Happy Birthday, Cheering**

OCR DOC

Carnegie Mellon University

Zero-Example Search

- Event of Interest: Birthday Party

ELAMP
Prototype System

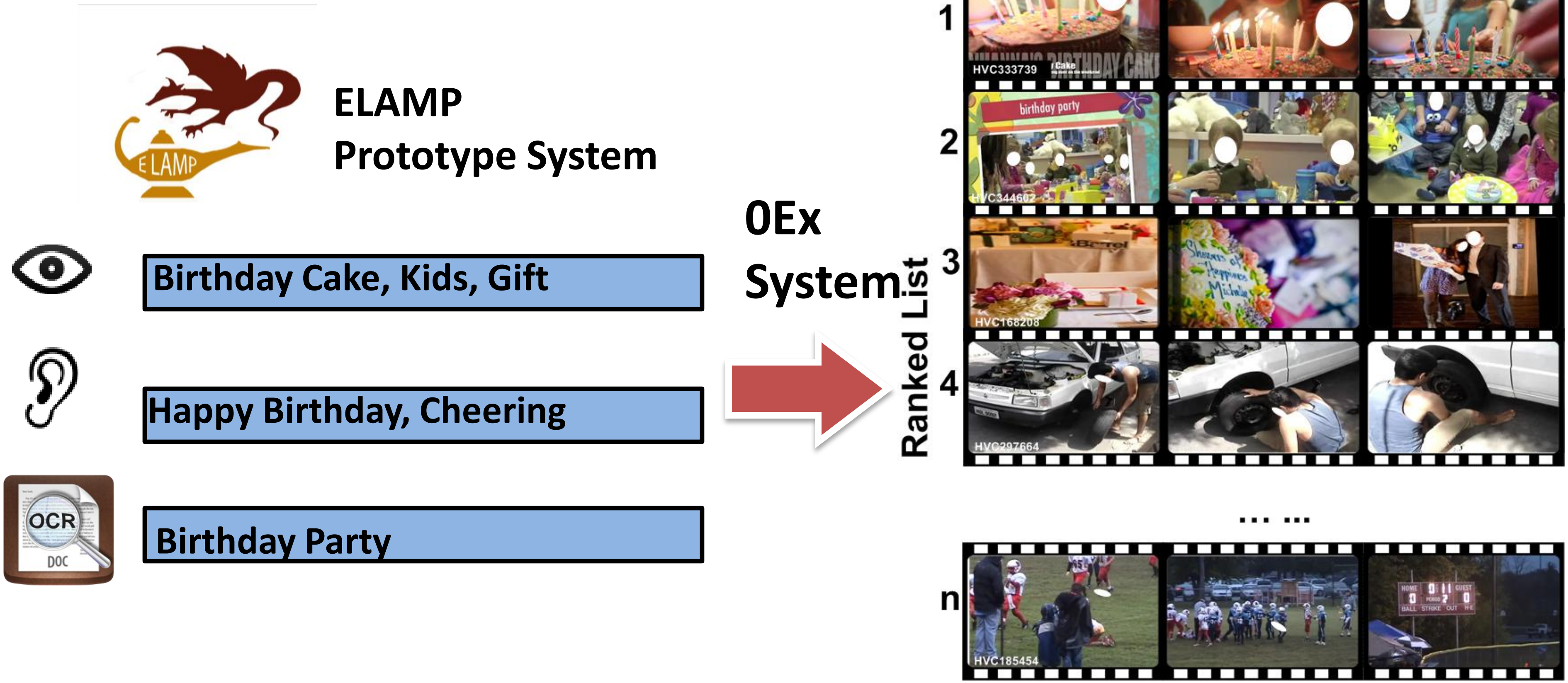
👁 **Birthday Cake, Kids, Gift**

👂 **Happy Birthday, Cheering**

OCR DOC **Birthday Party**

Zero-Example Search

- Event of Interest: Birthday Party



ELAMP

Prototype System

Birthday Cake, Kids, Gift

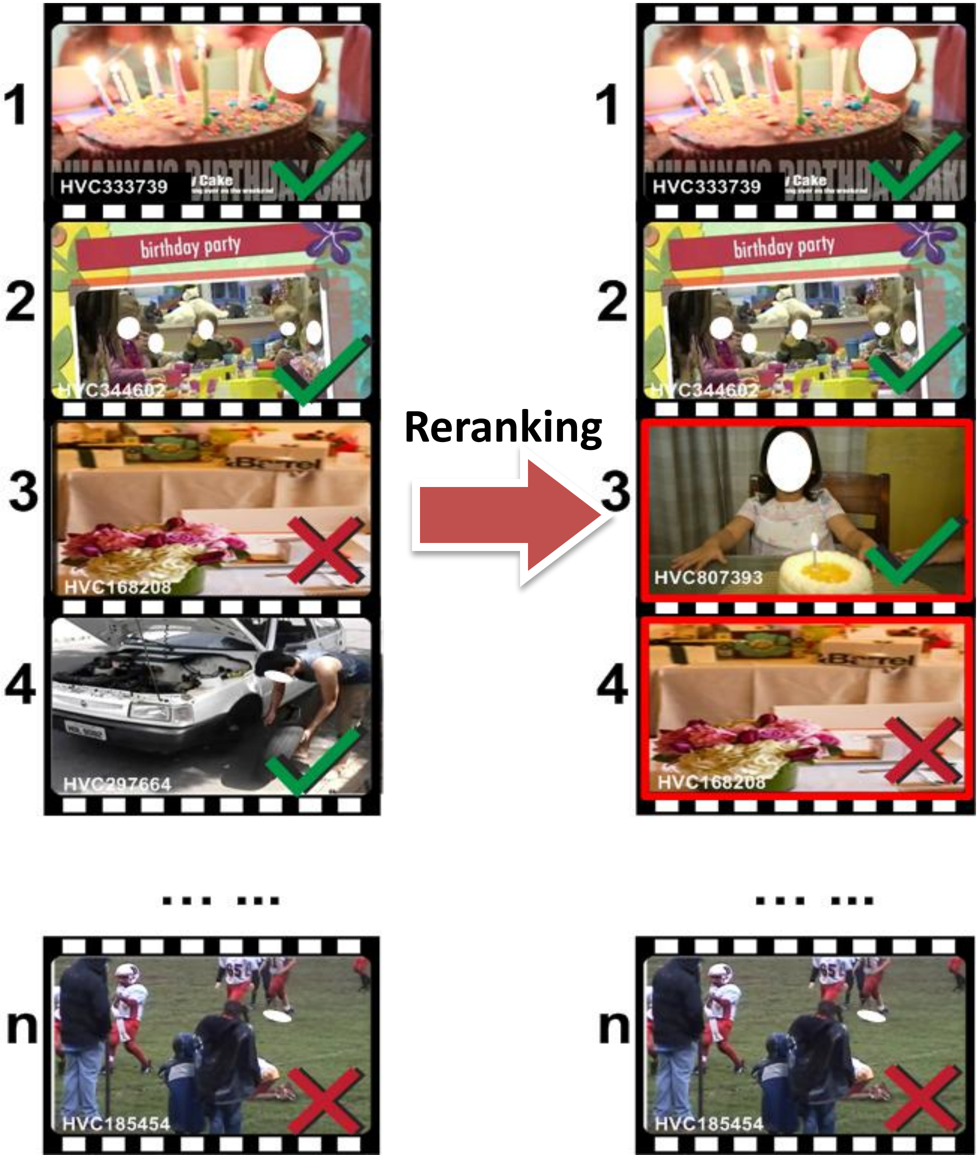
Happy Birthday, Cheering

Birthday Party

0Ex System

Ranked List

Reranking



- Intuition: initial ranked result is noisy.
- Refined by the multimodal info residing in the top ranked videos/images.

Outline

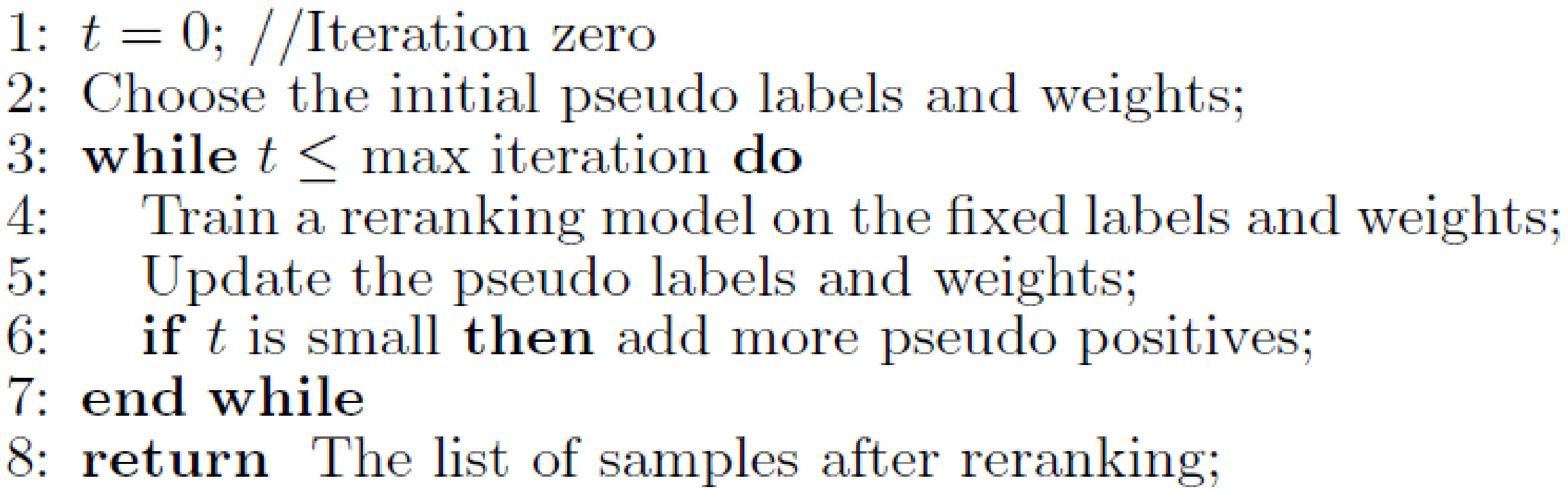
- Background
- **Related Work**
- Self-Paced Reranking (SPaR)
- Experiment Results
- Conclusions

Related Work

- Categorization of reranking methods:
  - Classification-based
    - (Yan et al. 2003) (Hauptmann et al. 2008)(Jiang et al. 2014)
  - Clustering-based
    - (Hsu et al. 2007)
  - LETOR(LEarning TO Rank)-based
    - (Liu et al. 2008) (Tian et al. 2008) (Tian et al. 2011)
  - Graph-based
    - (Hsu et al. 2007) (Nie et al. 2012)

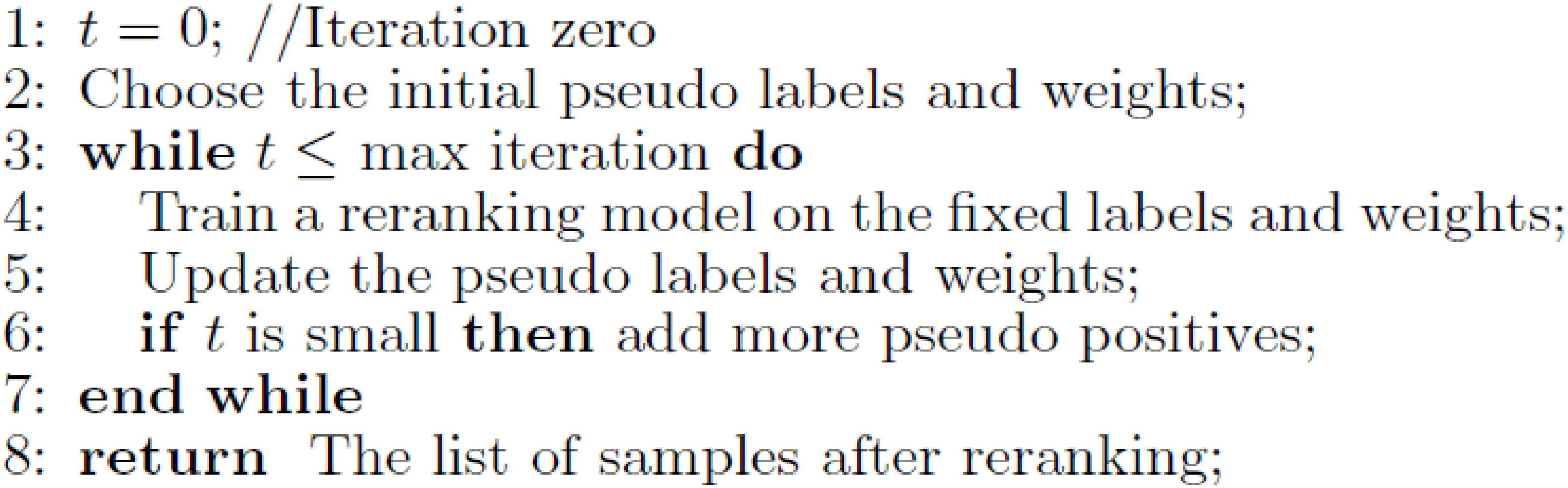
R. Yan, A. G. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *CVIR, 2003.*
A. G. Hauptmann, M. G. Christel, and R. Yan. Video retrieval based on semantic concepts. *Proceedings of the IEEE,* 96(4):602–622, 2008.
L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR, 2014*
W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *Multimedia, 2007.*
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, 2008.*
X. Tian, Y. Lu, L. Yang, and Q. Tian. Learning to judge image search results. In *Multimedia, 2011.*
X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X.-S. Hua. Bayesian video search reranking. In *Multimedia,* 2008.
W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *Multimedia, 2007.*
L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua. Harvesting visual concepts for image search with complex queries. In *Multimedia, 2012.*

Generic Reranking Algorithm



```
t = 0; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return  The list of samples after reranking;
```

Generic Reranking Algorithm



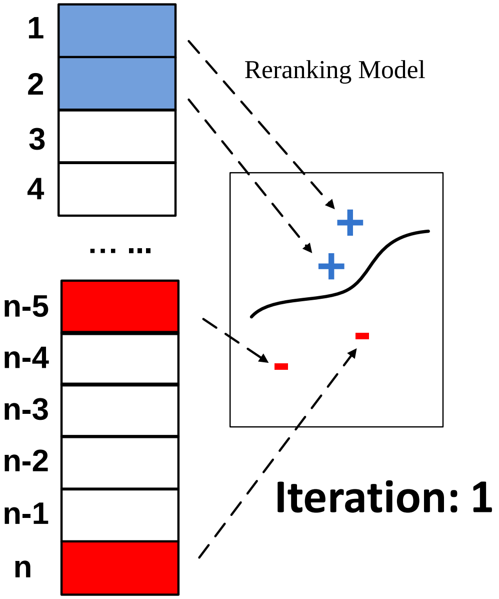
```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \le$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

Pseudo labels: assumed (hidden) labels for samples.

Zero-example: ground truth label unknown.

Generic Reranking Algorithm

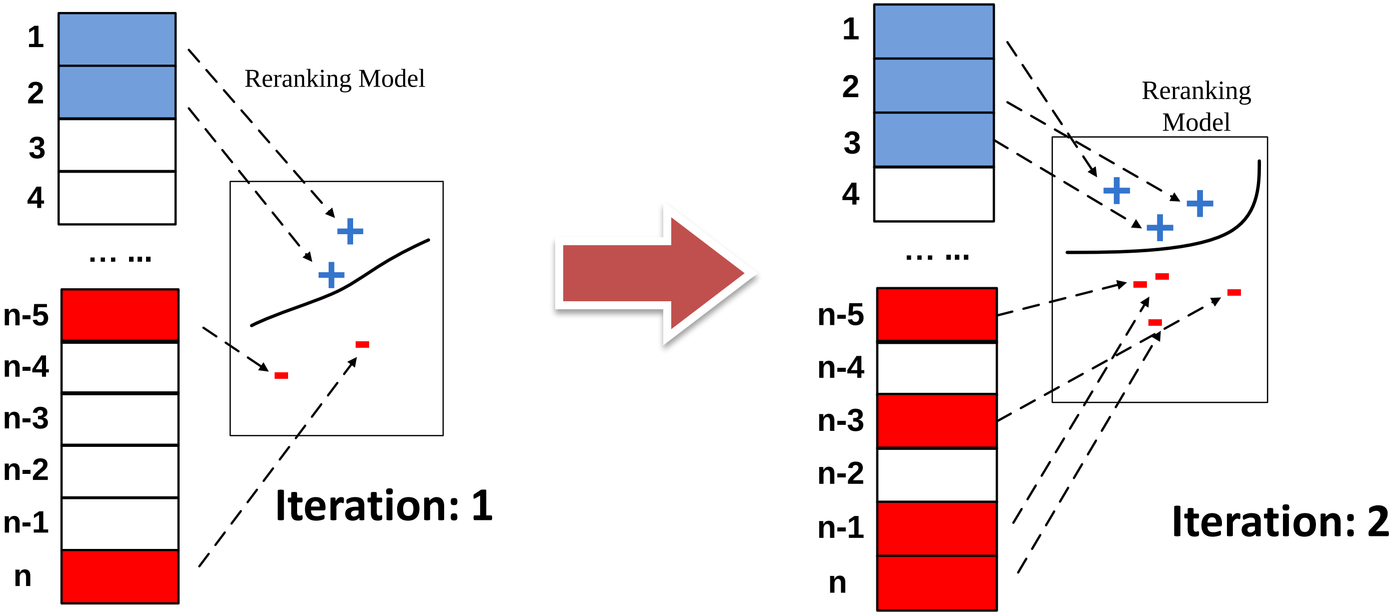
```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```



Iteration: 1

Generic Reranking Algorithm

```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return  The list of samples after reranking;
```

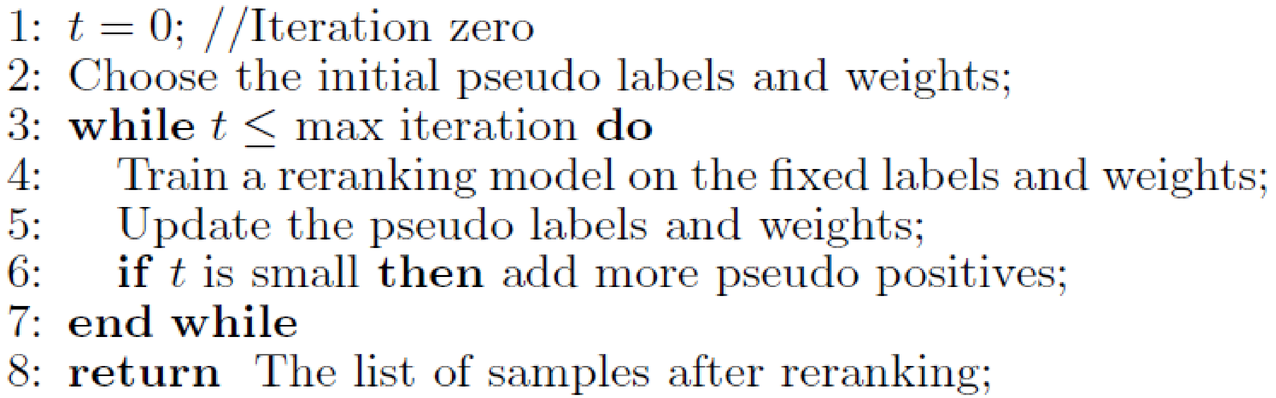


Intuition

True Label	Weighting		
	Binary	Predefined	Learned
+1	1.0	1.0	1.0
+1	1.0	1/2	1.0
-1	1.0	1/3	0.6
-1	1.0	1/4	0.1

- Existing methods assign equal weights to pseudo samples.
- Intuition: samples ranked at the top are generally more relevant than those ranked lower.
- Our approach: **learn the weight together with the reranking model**.

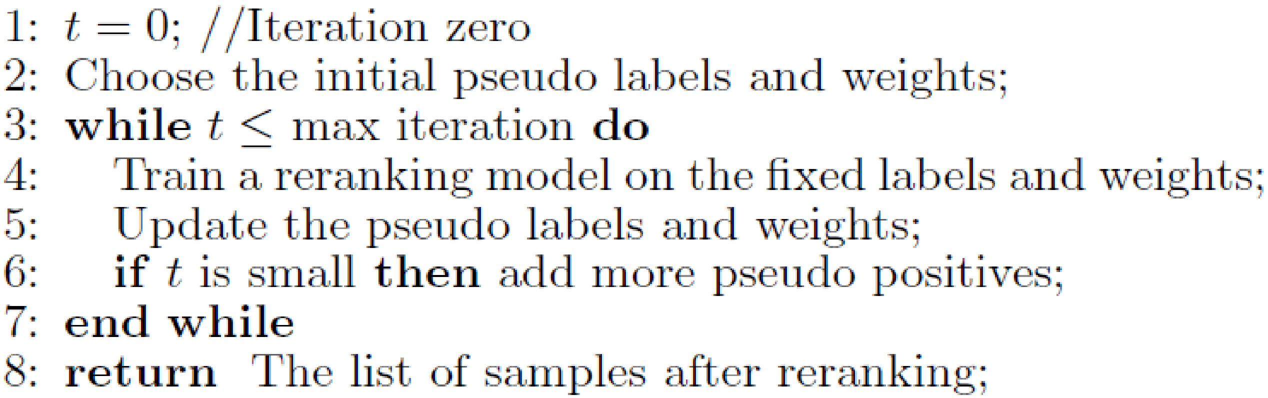
Generic Reranking Algorithm



```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

- Questions:
  1. Why the reranking algorithm performs iteratively?
  2. Does the process converge? If so, to where?
  3. Does the arbitrarily predefined weighting scheme converge?

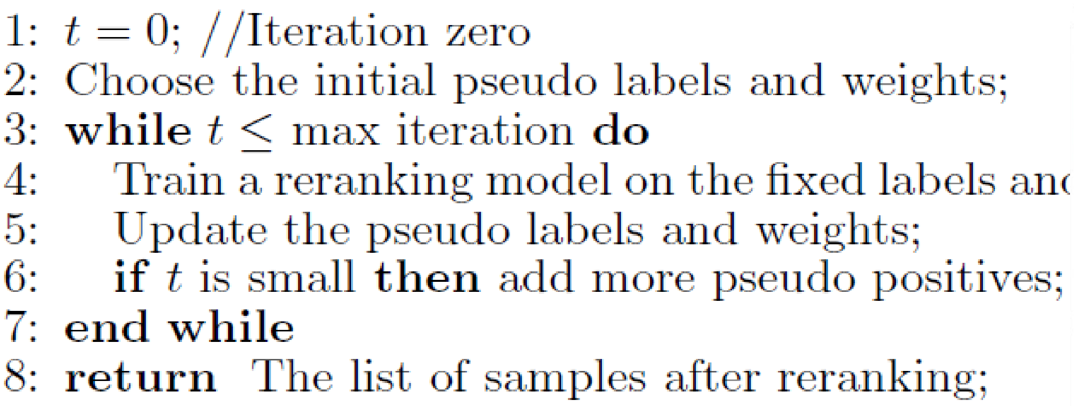
Generic Reranking Algorithm



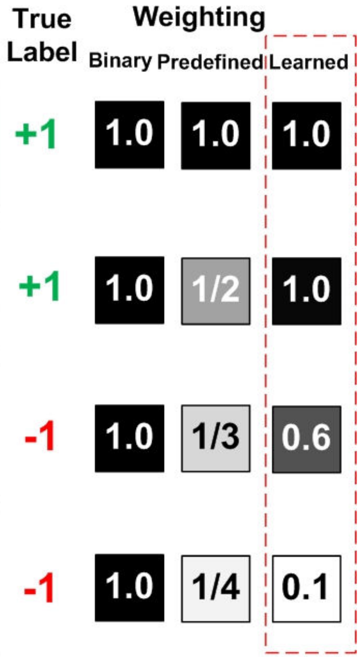
```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

- Questions:
  1. Why the reranking algorithm performs iteratively?
  2. Does the process converge? If so, to where?
  3. Does the arbitrarily predefined weighting scheme converge?

Generic Reranking Algorithm



```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels anc
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```



True Label	Weighting		
	Binary	Predefined	Learned
+1	1.0	1.0	1.0
+1	1.0	1/2	1.0
-1	1.0	1/3	0.6
-1	1.0	1/4	0.1

- Questions:
  1. Why the reranking algorithm performs iteratively?
  2. Does the process converge? If so, to where?
  3. Does the arbitrarily predefined weighting scheme converge?

Outline

- Background
- Related Work
- **Self-Paced Reranking (SPaR)**
- Experiment Results
- Conclusions

Self-paced Learning

- Curriculum Learning (Bengio et al. 2009) or self-paced learning (Kumar et al 2010) is a recently proposed learning paradigm that is inspired by the learning process of humans and animals.
- The samples are not learned randomly but organized in a meaningful order which illustrates from **easy** to gradually more **complex** ones.

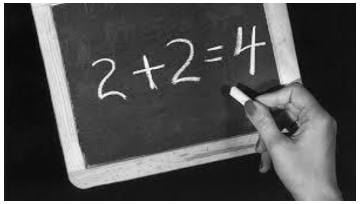
Prof. Bengio

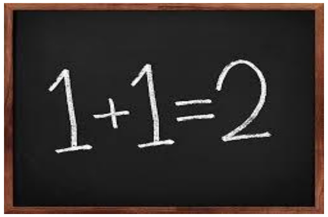
Prof. Koller

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML, 2009.*
M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In NIPS, pages 1189–1197, 2010.

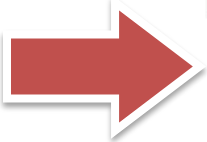
Self-paced Learning

- Easy samples to complex samples.
  - Easy sample → smaller loss to the already learned model.
  - Complex sample → bigger loss to the already learned model.





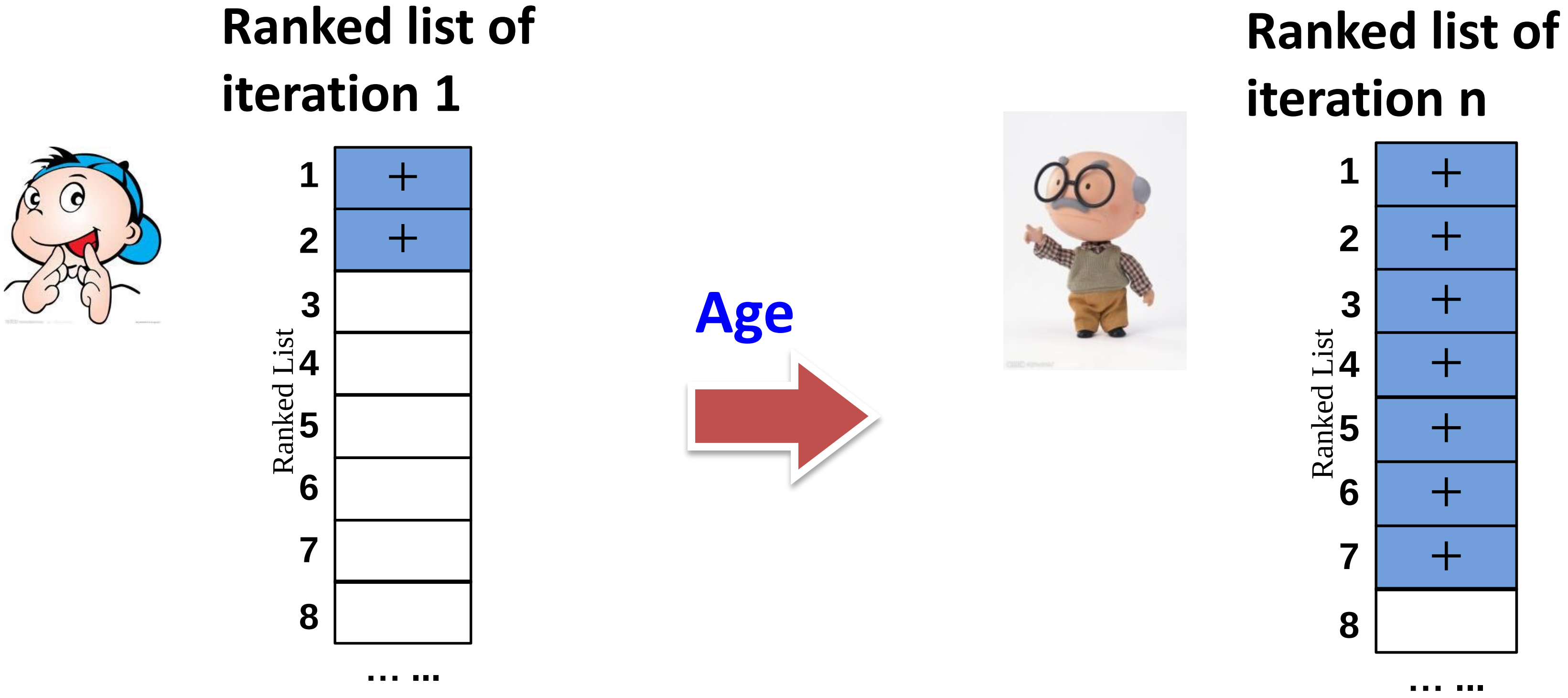
Age



$$
\begin{aligned}
\frac{1}{g-kv}\frac{dv}{dt} &= 1 \\
\int_0^T \frac{1}{g-kv}\frac{dv}{dt}\,dt &= \int_0^T dt \\
\int_{v_0}^{v(T)} \frac{1}{g-kv}\,dv &= T \\
-\frac{1}{k}\ln|g-kv|\,\Big|_{v_0}^{v(T)} &= T \\
\ln\left|\frac{g-kv(T)}{g-kv_0}\right| &= -kT \\
\frac{g-kv(T)}{g-kv_0} &= e^{-kT}
\end{aligned}
$$

Self-paced Learning

- In the context of reranking : **easy samples are the top-ranked videos that have smaller loss**.



Ranked list of iteration 1

Ranked list of iteration n

Self-paced Reranking (SPaR)

- We propose a novel framework named **Self-Paced Reranking (SPaR)** pronounced as /’spä/.
- Inspired by the self-paced learning theory.
- Formulate the problem as a concise optimization problem.

*Images from http://en.wikipedia.org/wiki/Hot_spring

Self-paced Reranking (SPaR)

- The propose model:

$$\min_{\Theta_1,\ldots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\ldots,\Theta_m,\mathbf{v},\mathbf{y};C,k)$$

$\Theta_1,\ldots,\Theta_m$ Reranking models for each modality.

$\mathbf{y} \in \{-1,1\}^n$ The pseudo label.

$\mathbf{v} \in [0,1]^n$ The weight for each sample.

Self-paced Reranking (SPaR)

- The propose model:

$$\min_{\Theta_1,\ldots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\ldots,\Theta_m,\mathbf{v},\mathbf{y};C,k)$$

$$= \min_{\mathbf{y},\mathbf{v},\Theta_1,\ldots,\Theta_m,} C\sum_{i=1}^{n} v_i \boxed{\text{loss-function}} + \boxed{\text{regularizer}}$$

$$\text{s.t.} \boxed{\text{constraints}}$$

$$\mathbf{y}\in\{-1,+1\}^n,$$

$$\mathbf{v}\in[0,1]^n,$$

$\Theta_1,\ldots,\Theta_m$ Reranking models for each modality.

$\mathbf{y}\in\{-1,1\}^n$ The pseudo label.

$\mathbf{v}\in[0,1]^n$ The weight for each sample.

The loss in the reranking model is discounted by a weight.

Self-paced Reranking (SPaR)

- The propose model:

$$\min_{\Theta_1,\ldots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\ldots,\Theta_m,\mathbf{v},\mathbf{y};C,k)$$

$$= \min_{\substack{\mathbf{y},\mathbf{v},\mathbf{w}_1,\ldots,\mathbf{w}_m,\\ b_1,\ldots,b_m,\{\ell_{ij}\}}} C\sum_{i=1}^{n} v_i \sum_{j=1}^{m} \ell_{ij} + \sum_{j=1}^{m}\frac{1}{2}\|\mathbf{w}_j\|_2^2 + \boxed{\text{regularizer}}$$

$$\text{s.t. } \forall i,\forall j, y_i(\mathbf{w}_j^T\phi(\mathbf{x}_{ij})+b_j) \geq 1-\ell_{ij}, \ell_{ij}\geq 0$$

$$\mathbf{y}\in\{-1,+1\}^n,$$

$$\mathbf{v}\in[0,1]^n,$$

$\Theta_1,\ldots,\Theta_m$ Reranking models for each modality.

$\mathbf{y}\in\{-1,1\}^n$ The pseudo label.

$\mathbf{v}\in[0,1]^n$ The weight for each sample.

For example the Loss in the SVM model.

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

Self-paced Reranking (SPaR)

- The propose model:

$$\min_{\Theta_1,\dots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\dots,\Theta_m,\mathbf{v},\mathbf{y};C,k)$$

$$= \min_{\mathbf{y},\mathbf{v},\Theta_1,\dots,\Theta_m,} C\sum_{i=1}^{n} v_i \boxed{\text{loss-function}} + mf(\mathbf{v};k)$$

$$\text{s.t.} \boxed{\text{constraints}}$$

$$\mathbf{y} \in \{-1,+1\}^n,$$

$$\mathbf{v} \in [0,1]^n,$$

$\Theta_1,\dots,\Theta_m$ Reranking models for each modality.

$\mathbf{y} \in \{-1,1\}^n$ The pseudo label.

$\mathbf{v} \in [0,1]^n$ The weight for each sample.

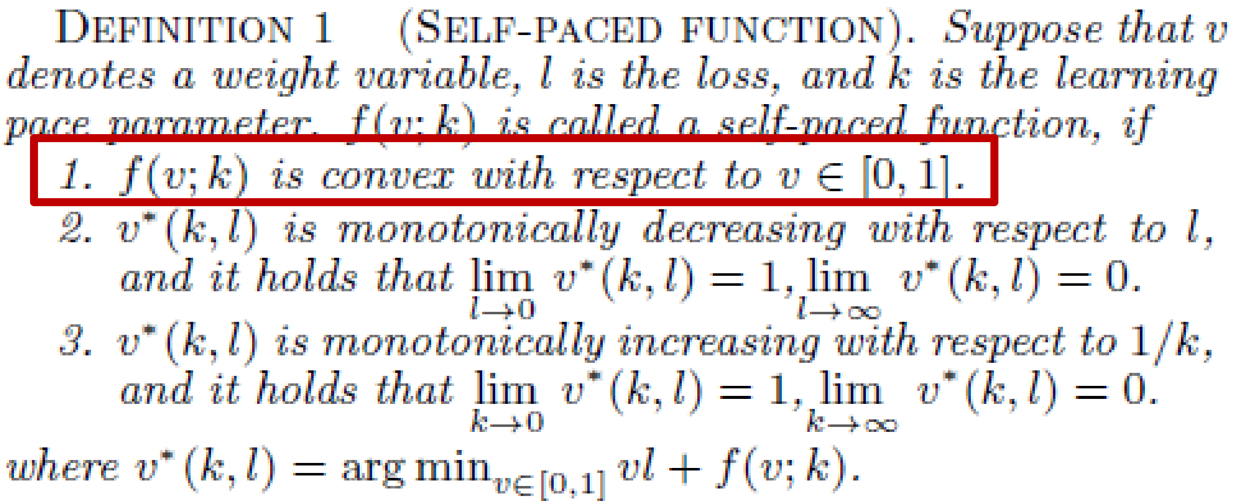
The self-paced is implemented by a regularizer. Physically corresponds to learning schemes that human use to learn different tasks.

m is the total number of modality.

f is the self-paced function in self-paced learning.

Self-paced Function

The definition which provides an axiom for self-paced learning.



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 $\lim_{k \to 0} v^*(k,l) = 1, \lim_{k \to \infty} v^*(k,l) = 0$.*

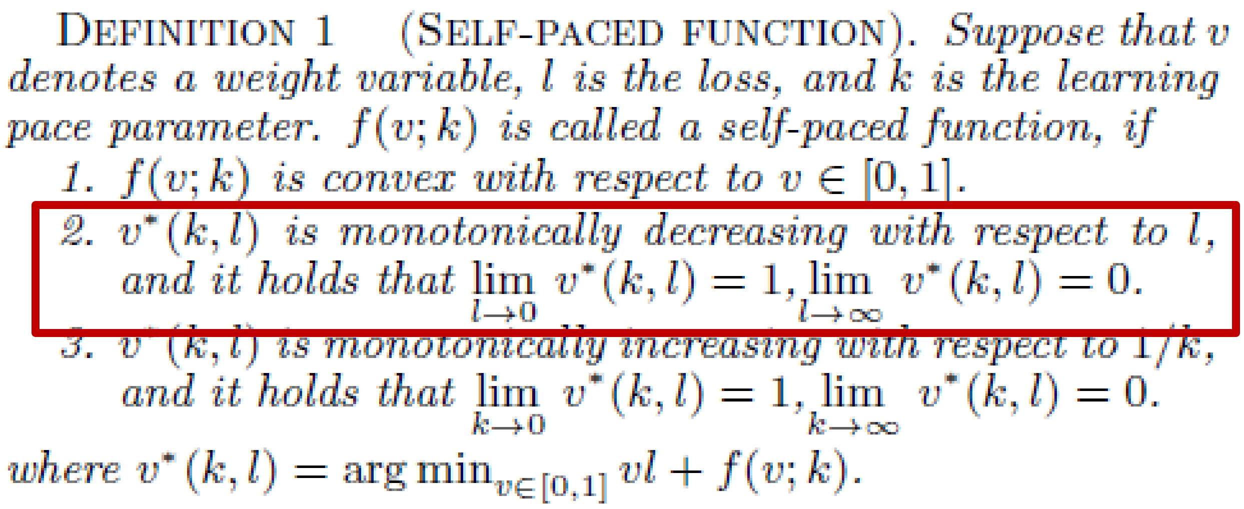
where $v^*(k,l) = \arg\min_{v \in [0,1]} vl + f(v;k)$.

Convex function.

$1/k$ is the age parameter in self-paced learning.
Physically it corresponds to the age of the learner.

Self-paced Function

We propose the definition which provides an axiom for self-paced learning.



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 $\lim_{k \to 0} v^*(k,l) = 1, \lim_{k \to \infty} v^*(k,l) = 0$.*

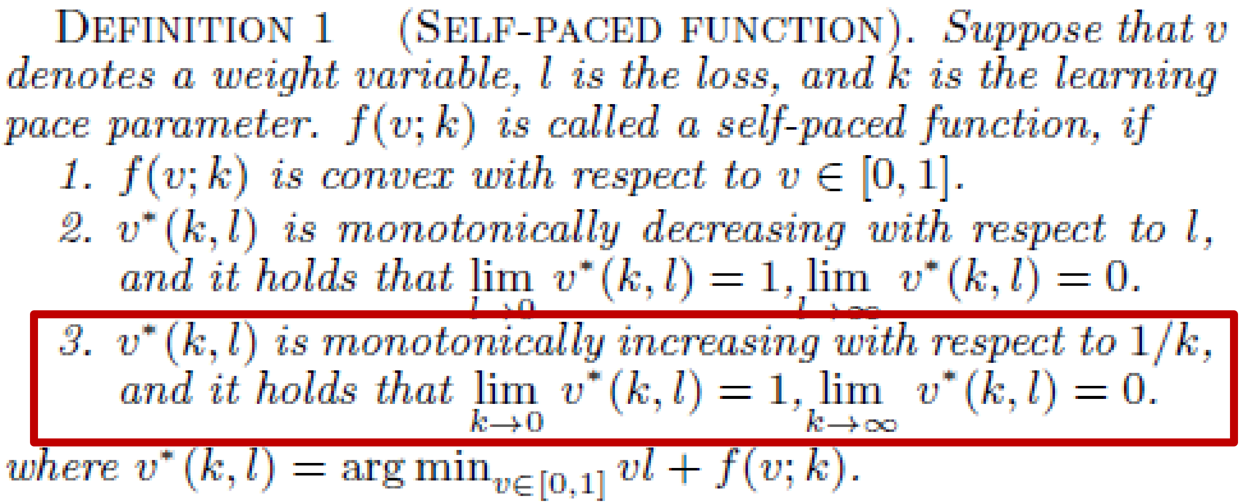
where $v^*(k,l) = \arg\min_{v \in [0,1]} vl + f(v;k)$.

Favors easy samples.

$1/k$ is the age parameter in self-paced learning.
Physically it corresponds to the age of the learner.

Self-paced Function

We propose the definition which provides an axiom for self-paced learning.



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

When the model is young use less samples;
When the model is mature use more;

$1/k$ is the age parameter in self-paced learning.
Physically it corresponds to the age of the learner.

Self-paced Function

Existing self-paced functions only support binary weighting (Kumar et al 2010).

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

Binary weighting

We argument the weight schemes and proposes the following soft weighting.

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

Linear weighting

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

Logarithmic weighting

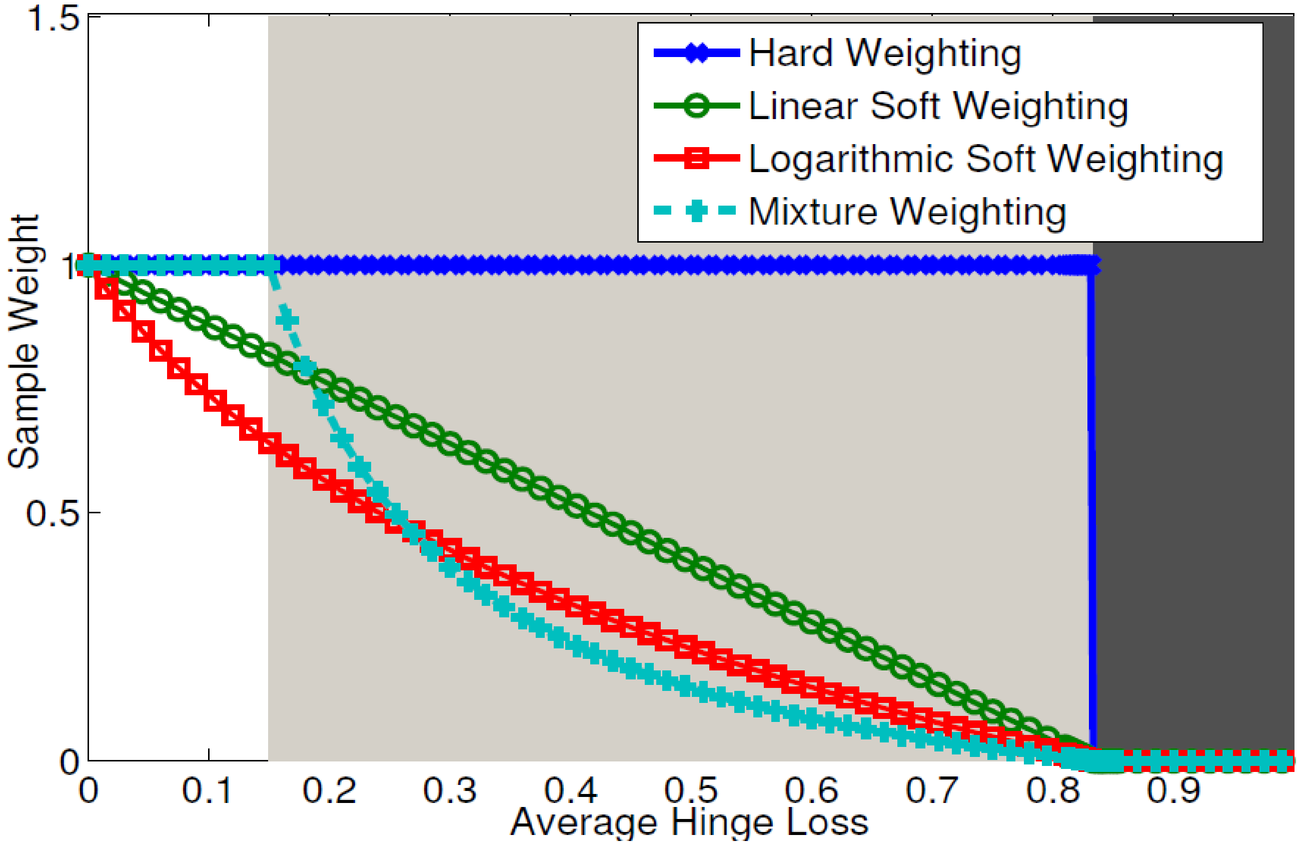
$$f(\mathbf{v};k,k') = -\zeta\sum_{i=1}^{n}\log(v_i + \zeta k),$$

Mixture weighting

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

Self-paced Function

Existing self-paced functions only support binary weighting (Kumar et al 2010).



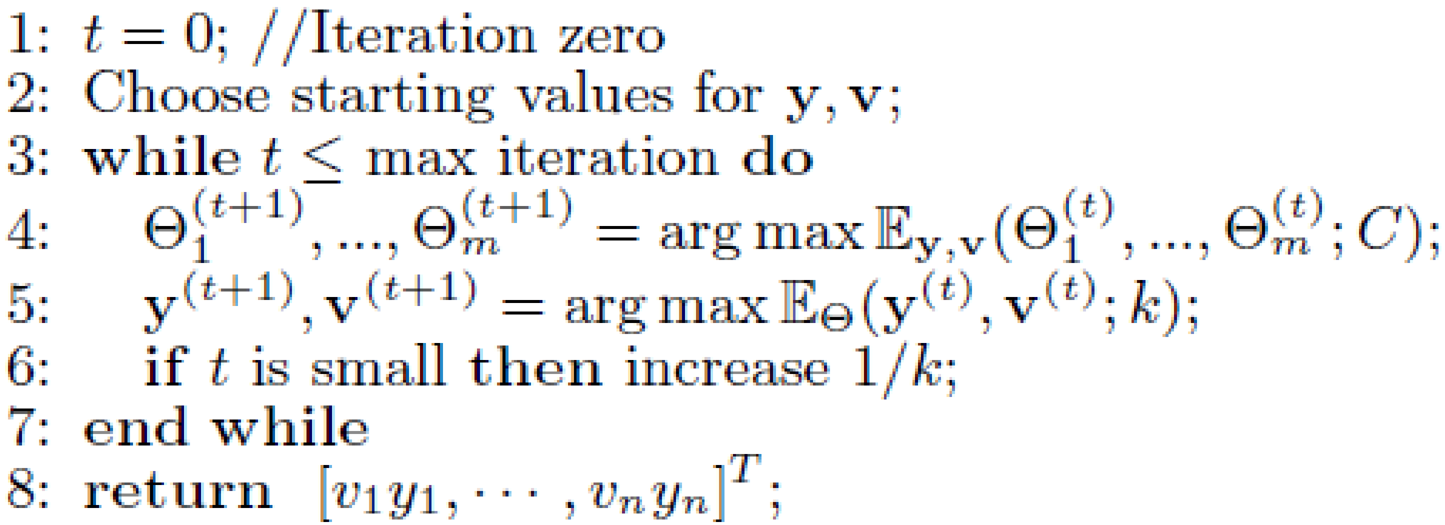
Binary weighting

Linear weighting

Logarithmic weighting

Mixture weighting

Reranking in Optimization and Conventional Perspective

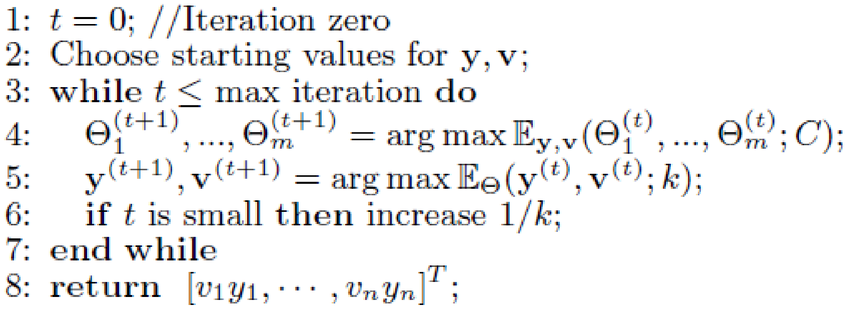


```
$t = 0$; //Iteration zero
Choose starting values for $\mathbf{y}, \mathbf{v}$;
while $t \leq$ max iteration do
    $\Theta_1^{(t+1)}, ..., \Theta_m^{(t+1)} = \arg\max \mathbb{E}_{\mathbf{y},\mathbf{v}}(\Theta_1^{(t)}, ..., \Theta_m^{(t)}; C)$;
    $\mathbf{y}^{(t+1)}, \mathbf{v}^{(t+1)} = \arg\max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

Algorithm 1: Reranking in Optimization Perspective.

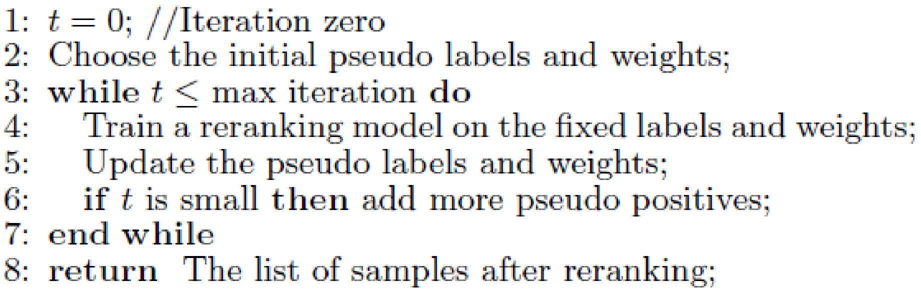
CCM (Cyclic Coordinate Method) is used to solve the problem. Fixing one variable and optimizing the other variables.

Reranking in Optimization and Conventional Perspective



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

Optimization perspective

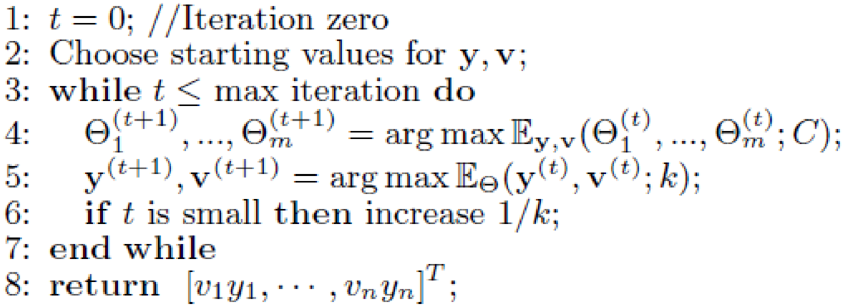


```
t = 0; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

Conventional perspective

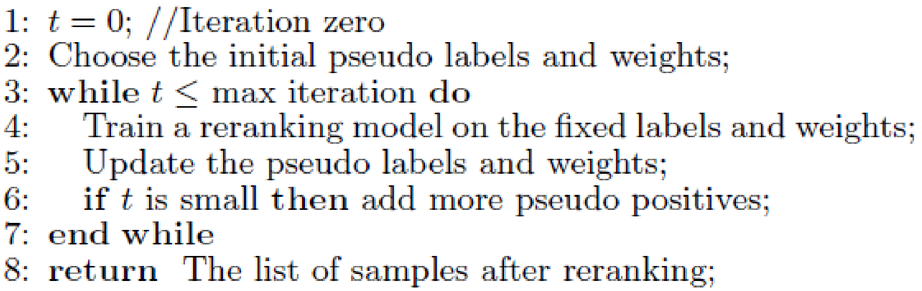
- Optimization perspective → theoretical justifications
- Conventional perspective offers practical lessons
- **Reranking is a self-paced learning process.**

Reranking in Optimization and Conventional Perspective



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

Algorithm 1: Reranking in Optimization Perspective.



```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

Algorithm 2: Reranking in Conventional Perspective.

Q1: Why the reranking algorithm performs iteratively?

A: Self-paced learning mimicking human and animal learning process (from easy to complex examples).

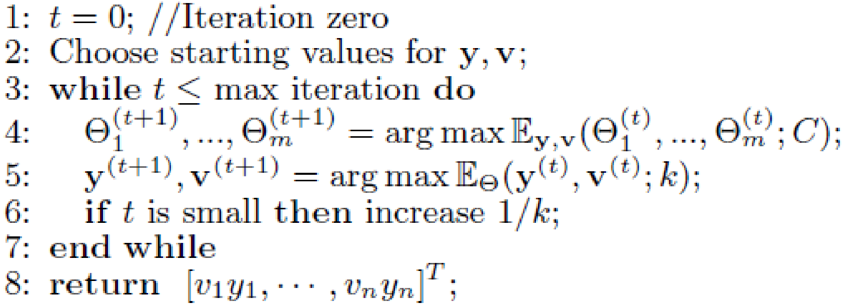
Q2: Does the process converge? If so, to where?

A: Yes, to the local optimum. See the theorem in our paper.

Q3: Does the arbitrarily predefined weighting scheme converge?

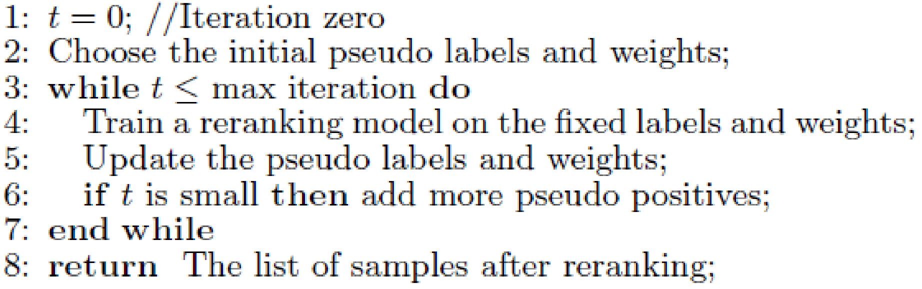
A: No, but the weights by self-paced function guarantees the convergence.

Reranking in Optimization and Conventional Perspective



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

Algorithm 1: Reranking in Optimization Perspective.



```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

Algorithm 2: Reranking in Conventional Perspective.

Q1: Why the reranking algorithm performs iteratively?

A: Self-paced learning mimicking human and animal learning process (from easy to complex examples).

Q2: Does the process converge? If so, to where?

A: Yes, to the local optimum. See the theorem in our paper.

Q3: Does the arbitrarily predefined weighting scheme converge?

A: No, but the weights by self-paced function guarantees the convergence.

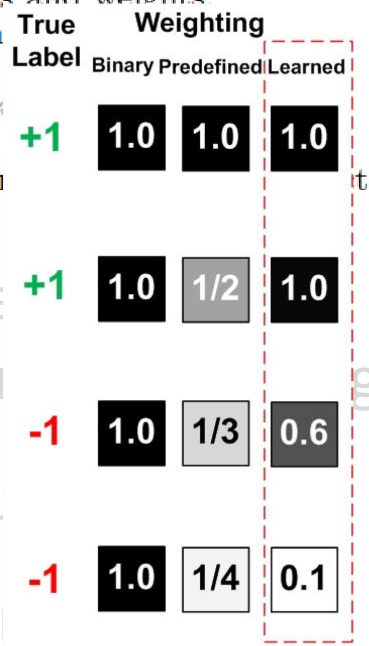
Reranking in Optimization and Conventional Perspective

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

Algorithm 1: Reranking in Optimization Perspective.

```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add m
end while
return The list of sample
```

Algorithm 2: Reranking i tive.



True Label	Weighting		
	Binary	Predefined	Learned
+1	1.0	1.0	1.0
+1	1.0	1/2	1.0
-1	1.0	1/3	0.6
-1	1.0	1/4	0.1

Q1: Why the reranking algorithm performs itera

A: Self-paced learning mimicking human and a g process (from easy to complex examples).

Q2: Does the process converge? If so, to where

A: Yes, to the local optimum. See the theorem i

Q3: Does the arbitrarily predefined weighting scheme converge?

A: No, but the weights by self-paced function guarantees the convergence.

Outline

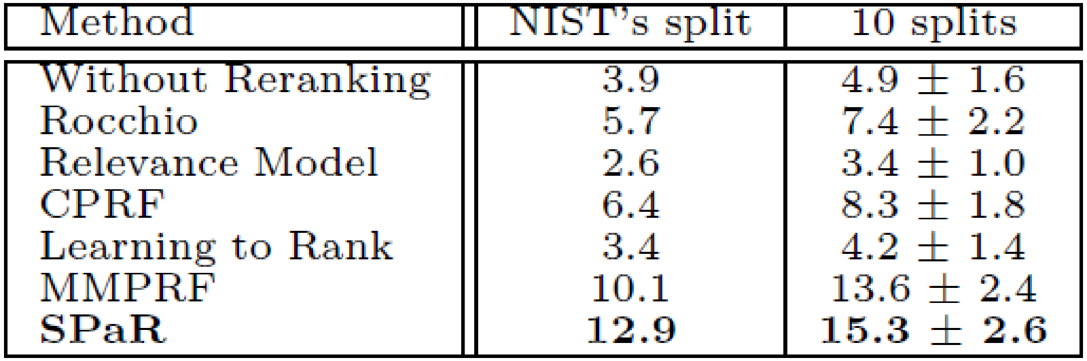
- Background
- Related Work
- Self-Paced Reranking (SPaR)
- **Experiment Results**
- Conclusions

TRECVID Multimedia Event Detection

- Dataset: MED13Test (around 34,000 videos) on 20 predefined events.
- Test on the NIST's split (25,000 videos).
- Evaluated by Mean Average Precision.
- Four types of high-level features:
  - ASR, OCR, SIN, and ImageNet DCNN
- Two types of low-level features:
  - Dense trajectory and MFCC
- Configurations:
  - Mixture self-paced function
  - Starting values obtained by MMPRF
  - Setting age parameter to include certain number of samples.

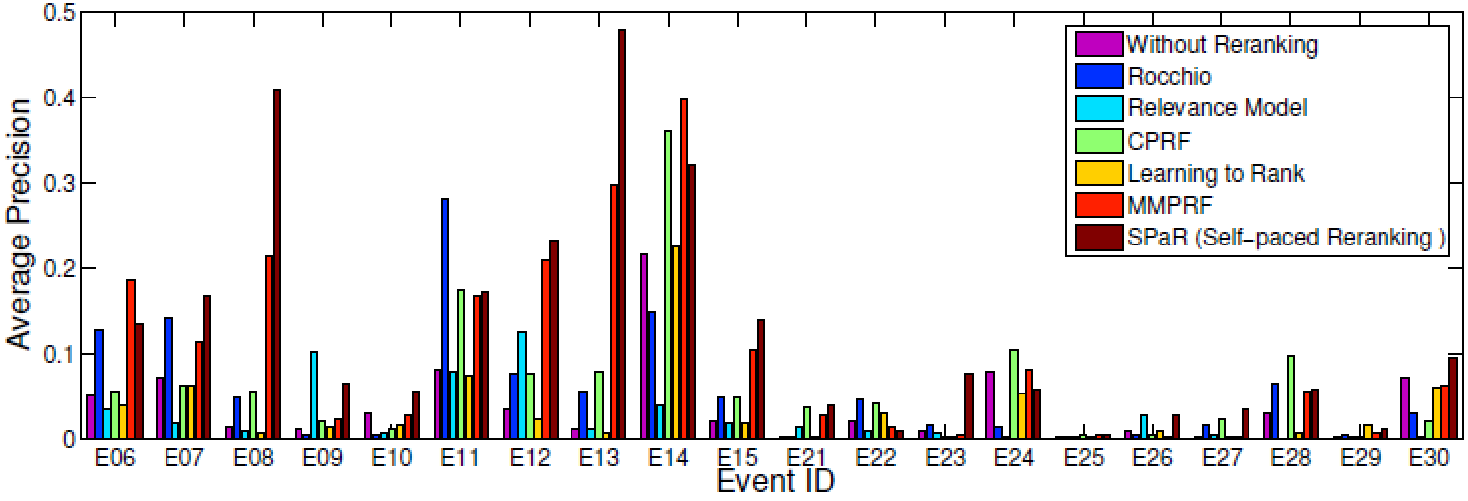
Results on MED13Test

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



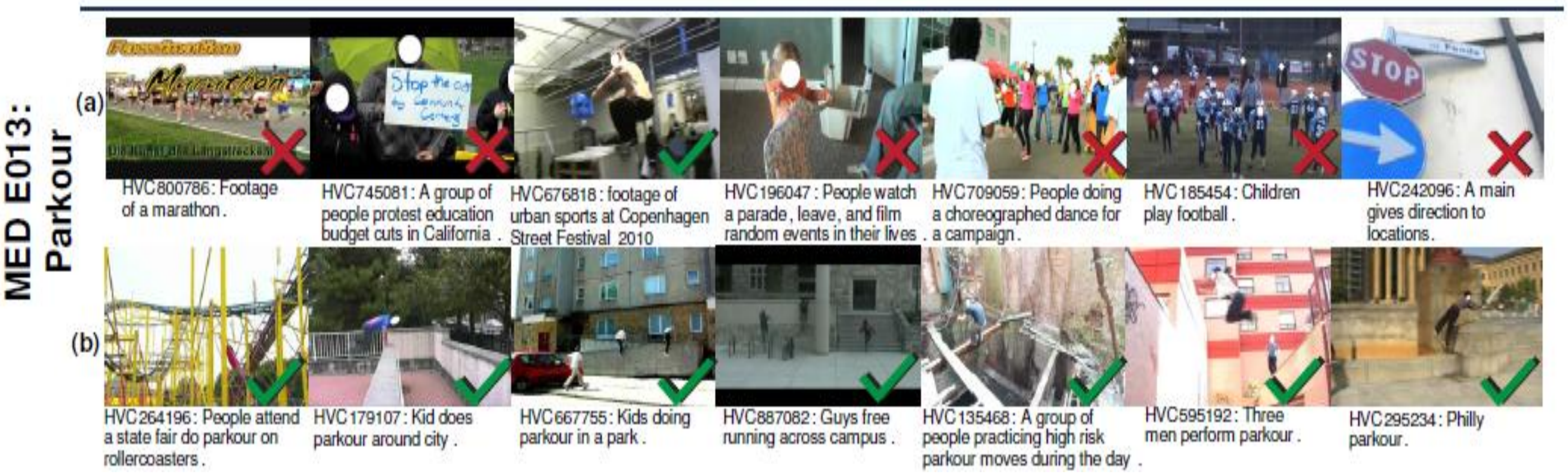
Method	NIST's split	10 splits
Without Reranking	3.9	4.9 ± 1.6
Rocchio	5.7	7.4 ± 2.2
Relevance Model	2.6	3.4 ± 1.0
CPRF	6.4	8.3 ± 1.8
Learning to Rank	3.4	4.2 ± 1.4
MMPRF	10.1	13.6 ± 2.4
SPaR	**12.9**	**15.3 ± 2.6**

By far the best MAP of the 0Ex task reported on the dataset!

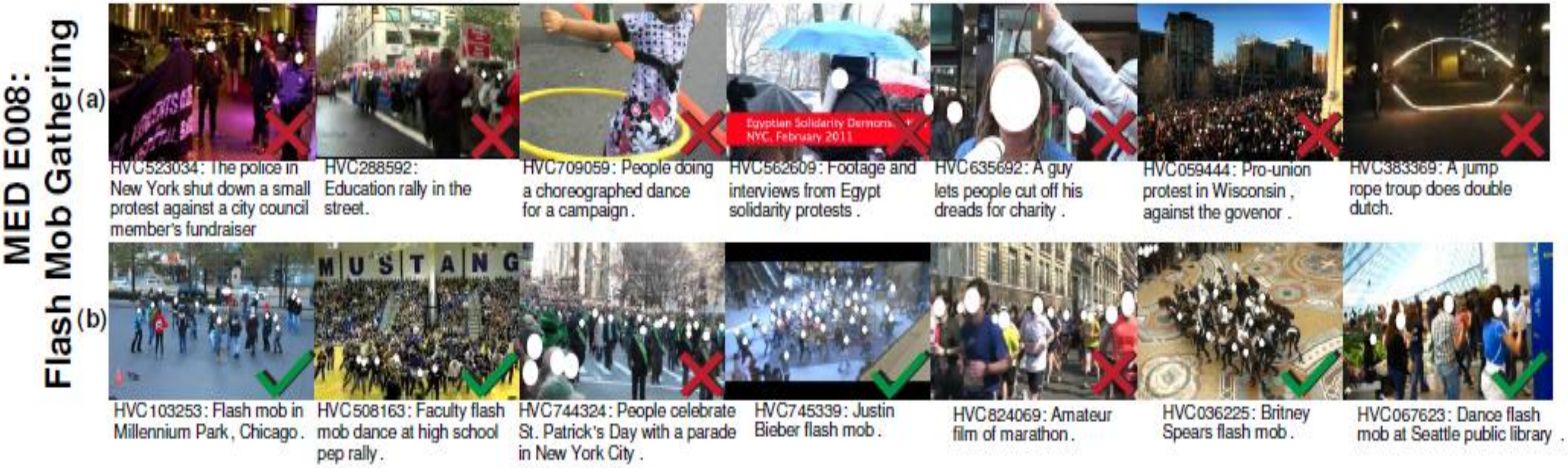


Outperforms MMPRF on 15/20 events.

Comparison of top-ranked videos


MED E013: Parkour
(a)
HVC800786: Footage of a marathon.
HVC745081: A group of people protest education budget cuts in California.
HVC676818: footage of urban sports at Copenhagen Street Festival 2010
HVC196047: People watch a parade, leave, and film random events in their lives.
HVC709059: People doing a choreographed dance for a campaign.
HVC185454: Children play football.
HVC242096: A main gives direction to locations.
(b)
HVC264196: People attend a state fair do parkour on rollercoasters.
HVC179107: Kid does parkour around city.
HVC667755: Kids doing parkour in a park.
HVC887082: Guys free running across campus.
HVC135468: A group of people practicing high risk parkour moves during the day.
HVC595192: Three men perform parkour.
HVC295234: Philly parkour.

Comparison of top-ranked videos



(a)

HVC523034: The police in New York shut down a small protest against a city council member's fundraiser

HVC288592: Education rally in the street.

HVC709059: People doing a choreographed dance for a campaign.

HVC562609: Footage and interviews from Egypt solidarity protests.

HVC635692: A guy lets people cut off his dreads for charity.

HVC059444: Pro-union protest in Wisconsin, against the govenor.

HVC383369: A jump rope troup does double dutch.

(b)

HVC103253: Flash mob in Millennium Park, Chicago.

HVC508163: Faculty flash mob dance at high school pep rally.

HVC744324: People celebrate St. Patrick's Day with a parade in New York City.

HVC745339: Justin Bieber flash mob.

HVC824069: Amateur film of marathon.

HVC036225: Britney Spears flash mob.

HVC067623: Dance flash mob at Seattle public library.

TRECVID MED 2014

- Very challenging task:
  - Search over MED14Eval Full (**200K videos**)
  - The ground-truth label is unavailable.
  - Can only submit one run.
  - Ad-Hoc queries (events) are unknown to the system.

- SPaR yields **outstanding improvements** for TRECVID MED14 **000Ex and 010Ex condition!**
- Take no more than **60 seconds/query** on a workstation.
- **Cost-effective Method!**

Web Query Dataset

- Web image (353 queries over 71,478 images)
- Densely sampled SIFT are extracted.
- Parameters are tuned on a validation set.
- Mixture self-paced function is used.

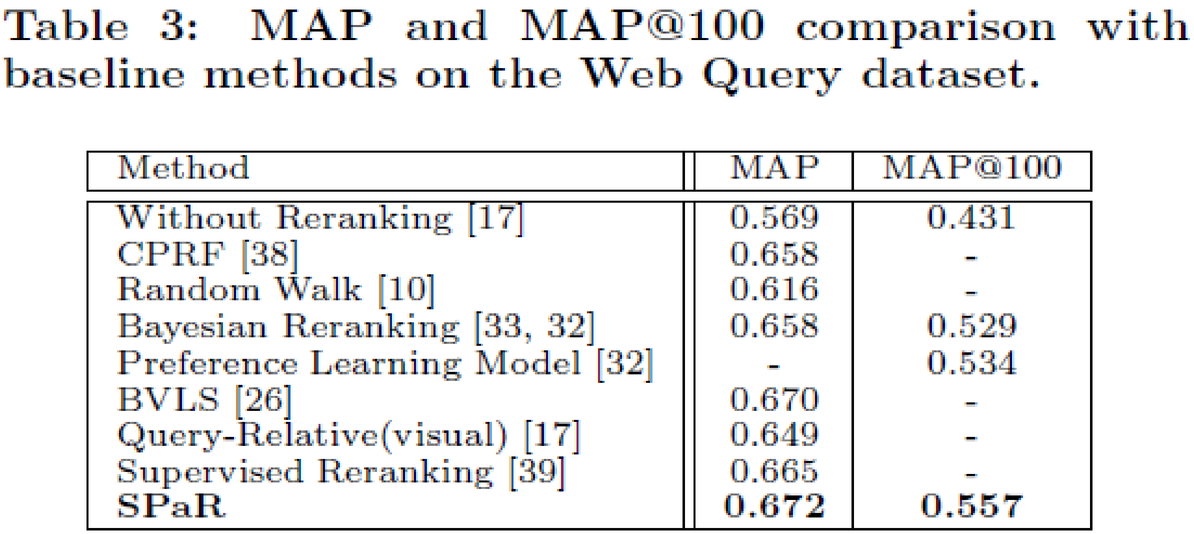


Table 3: MAP and MAP@100 comparison with baseline methods on the Web Query dataset.

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**

SPaR also works for image reranking (single modality)

Discussions

- Two scenarios where SPaR fails:
  - Initial top-ranked videos are completely off-topic.
  - Features used in reranking are not discriminative to the queries.
- Sensitive to random starting values
  - Initializing by existing reranking algorithms such as MMPRF/CPRF.
- Tuning the age parameter by the statistics collected from the ranked samples.
  - as opposed to absolute values.

Outline

- Motivation
- Related Work
- Jensen - Shannon Tiling
- Experiment Results
- **Conclusions**

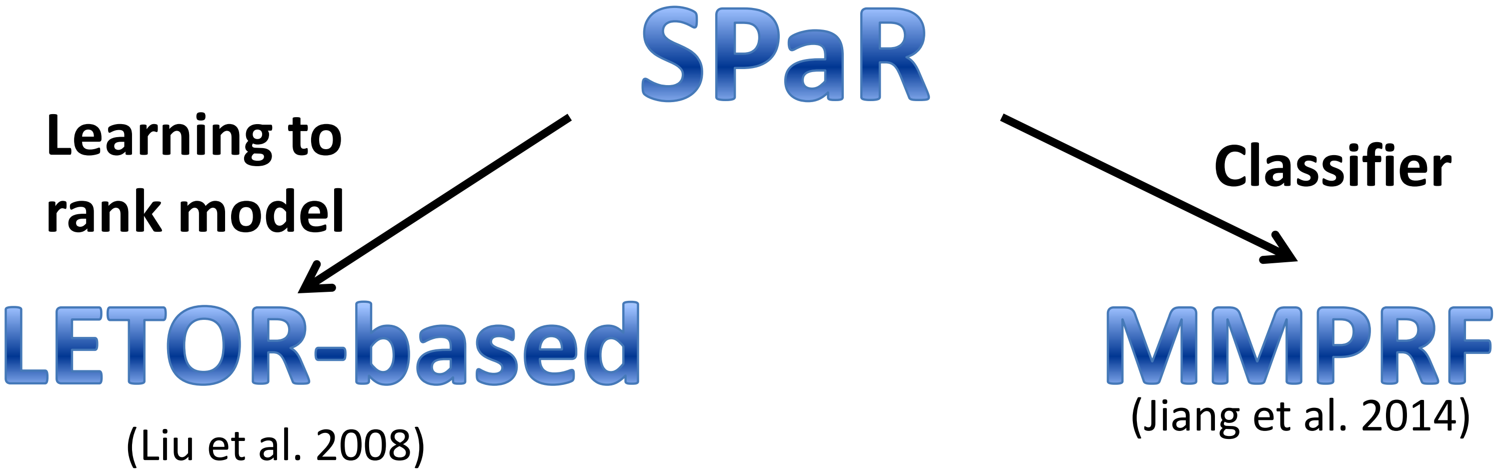
Summary

- A few messages to **take away from this talk**:
  - Reranking follows the self-paced learning process.
  - SPaR is a novel and general framework with theoretical backgrounds for multimodal reranking.
  - SPaR achieves by far the **best result** on the Multimedia Event Detection zero-example search.

THANK YOU.
Q&A?

APPENDIX

Relation to Existing Reranking Methods



SPaR

Learning to rank model

LETOR-based

(Liu et al. 2008)

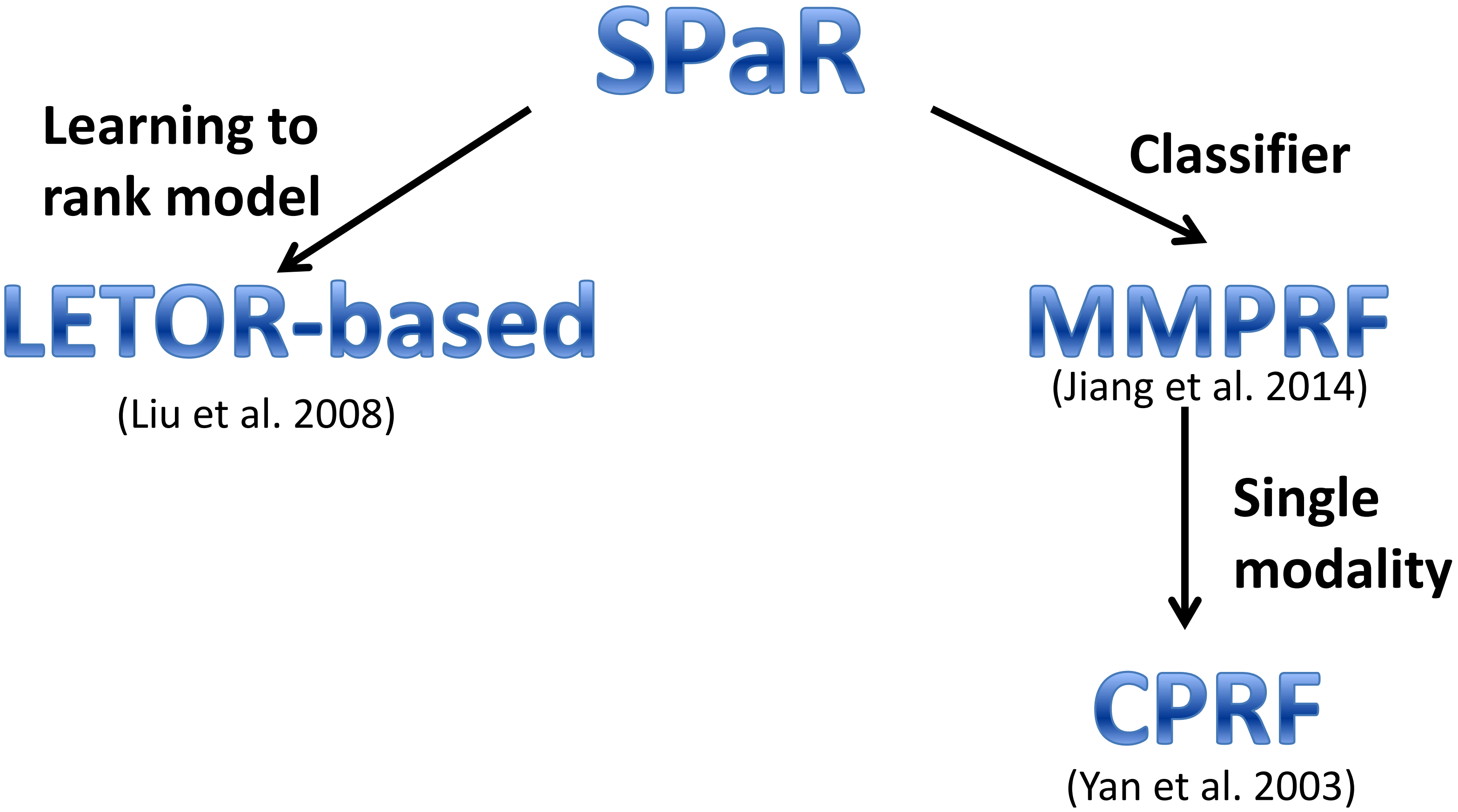
Classifier

MMPRF

(Jiang et al. 2014)

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, 2008.*
L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR, 2014*

Relation to Existing Reranking Methods



SPaR

Learning to rank model

LETOR-based

(Liu et al. 2008)

Classifier

MMPRF

(Jiang et al. 2014)

Single modality

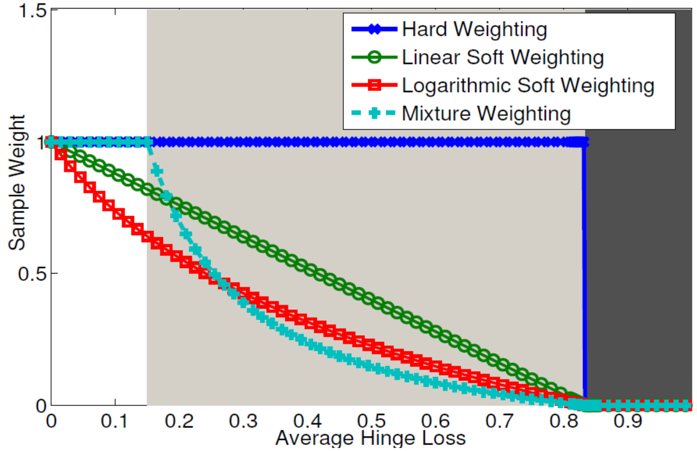
CPRF

(Yan et al. 2003)

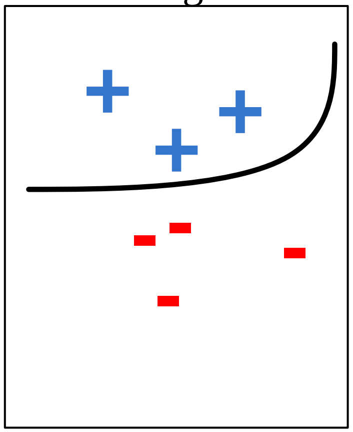
Y. Liu, T. Mei, X.-S. Hua, J. Tang, X. Wu, and S. Li. Learning to video search rerank via pseudo preference feedback. In *ICME, 2008.*
L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR, 2014*
R. Yan, A. G. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *CVIR, 2003.*

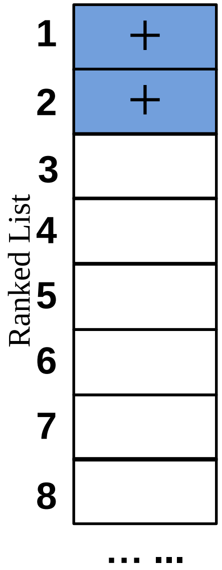
SPaR for Practitioners

1. Pick a self-paced function
   - Binary/Linear/Logarithmic/Mixture weighting .
2. Pick a favorite reranking model
   - SVM*/Logistic Regression/Learning to Rank.
3. Get reasonable starting values
   - Initializing by existing reranking algorithms.



Reranking Model





*weighted sample LibSVM http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#weights_for_data_instances

SPaR for Practitioners

Iterate the following steps:

- Training a reranking model using the pseudo samples.
- Selecting pseudo positive samples and their weights by self-paced function. Selecting some pseudo negative samples randomly.
- Changing the model age $1/k$ to include more positive samples for the next iteration (setting to include certain number of examples).

Cyclic Coordinate Algorithm

- The propose model:

$$\min_{\Theta_1,\ldots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\ldots,\Theta_m,\mathbf{v},\mathbf{y};C,k)$$

$\Theta_1,\ldots,\Theta_m$ Reranking models for each modality.

$\mathbf{y}\in\{-1,1\}^n$ The pseudo label.

$\mathbf{v}\in[0,1]^n$ The weight for each sample.

Algorithm (Cyclic Coordinate Method):

1. Fix $\mathbf{v},\mathbf{y}$, optimize $\Theta_1,\ldots,\Theta_m$

   Using the existing off-the-shelf algorithm.

2. Fix $\Theta_1,\ldots,\Theta_m$ $\mathbf{v}$ optimize $\mathbf{y}$

   Enumerating binary labels.

3. Fix $\Theta_1,\ldots,\Theta_m$ $\mathbf{y}$ optimize $\mathbf{v}$

   Selecting samples and their weights for the next iteration

4. Change the age parameter to include more samples.