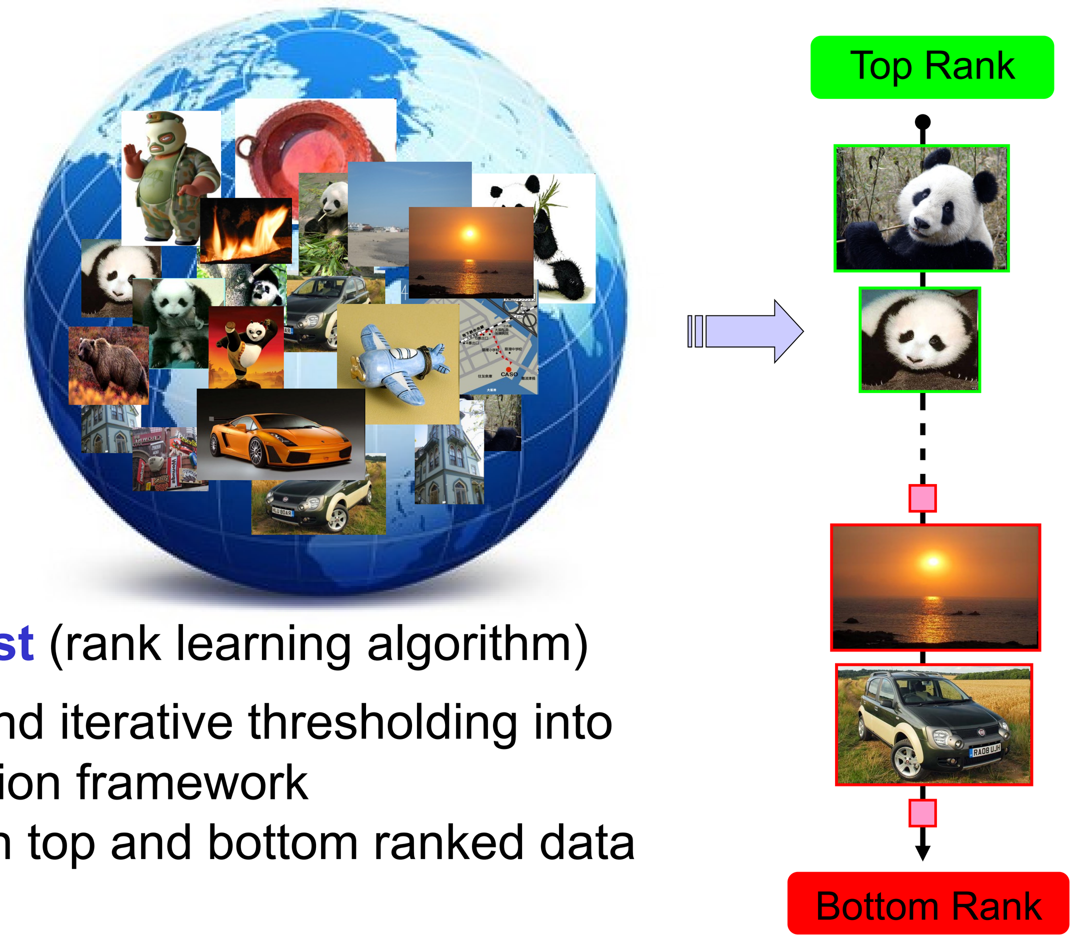
Imbalanced RankBoost for Efficiently Ranking Large-Scale Image/Video Collections

Michele Merler†, Rong Yan* and John R. Smith* † Columbia University, * IBM TJ Watson Research Center

Email: mmerler@cs.columbia.edu, {yanr, jsmith}@us.ibm.com

Introduction

- **Efficiently ranking** and retrieving relevant data is increasingly important for organizing / managing large-scale image/video collections
- Vast majority of users prefer to browse only a limited number of **top ranked examples**, while completely ignore the rest



- **Imbalanced RankBoost** (rank learning algorithm)
  - merges RankBoost and iterative thresholding into unified loss optimization framework
  - distinguishes between top and bottom ranked data

Learning to Rank

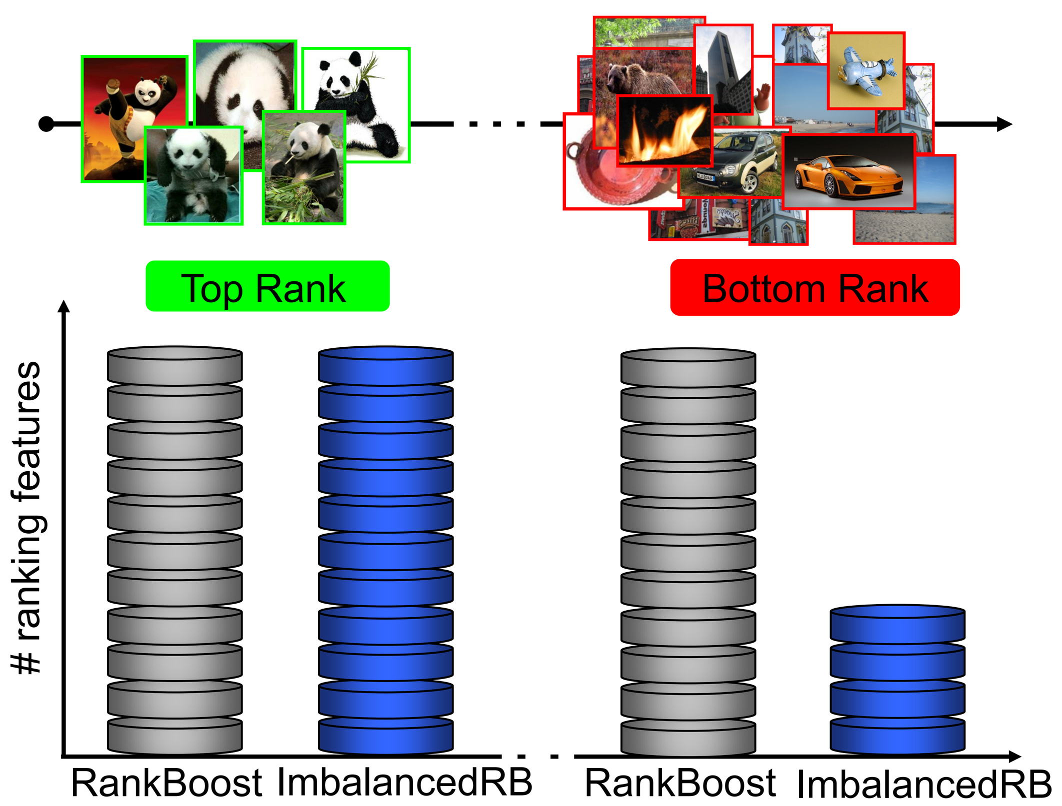
- **Goal:** produce a ranked list of originally unordered examples X, so that the relevant ones are placed as close as possible to the top
- RankBoost
  - Combine a pool of K "weak" ranking features $h_k(x)$ into composite ranking function F(x)
  - Exponential loss function minimization (~minimize # of ranking errors)
  - Joint selection of $h_k(x)$ and optimization of combination weights α_k

No distinction between top- and bottom-ranked examples → **Problem: Suboptimal Ranking Process**

Solution: Imbalanced RankBoost

- More ranking features for top ranked data
- Truncate ranking feature computation for the data ranked below learnt cutoff thresholds
- More efficient ranking process

RankBoost vs. Imbalanced RankBoost



Imbalanced RankBoost

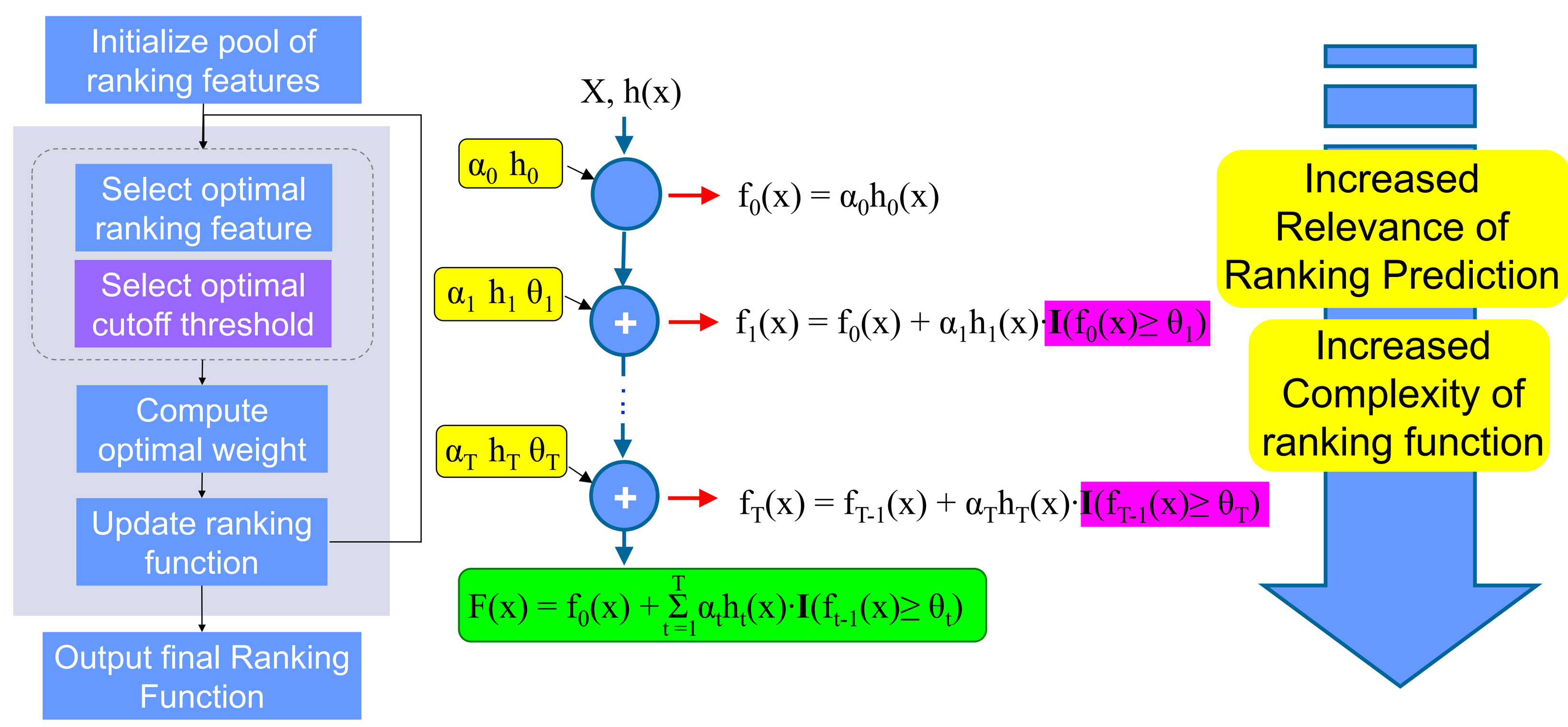
- Bipartite setting: relevant and non-relevant examples *x* are grouped in disjoint sets X_0 and X_1
- Ranking function F(x) contains **cut-off thresholds θ_t**

$$F(x) = f_0(x) + \sum_{t=1}^{T} \alpha_t h_t(x) \cdot \mathbf{I}(f_{t-1}(x) \geq \theta_t)$$

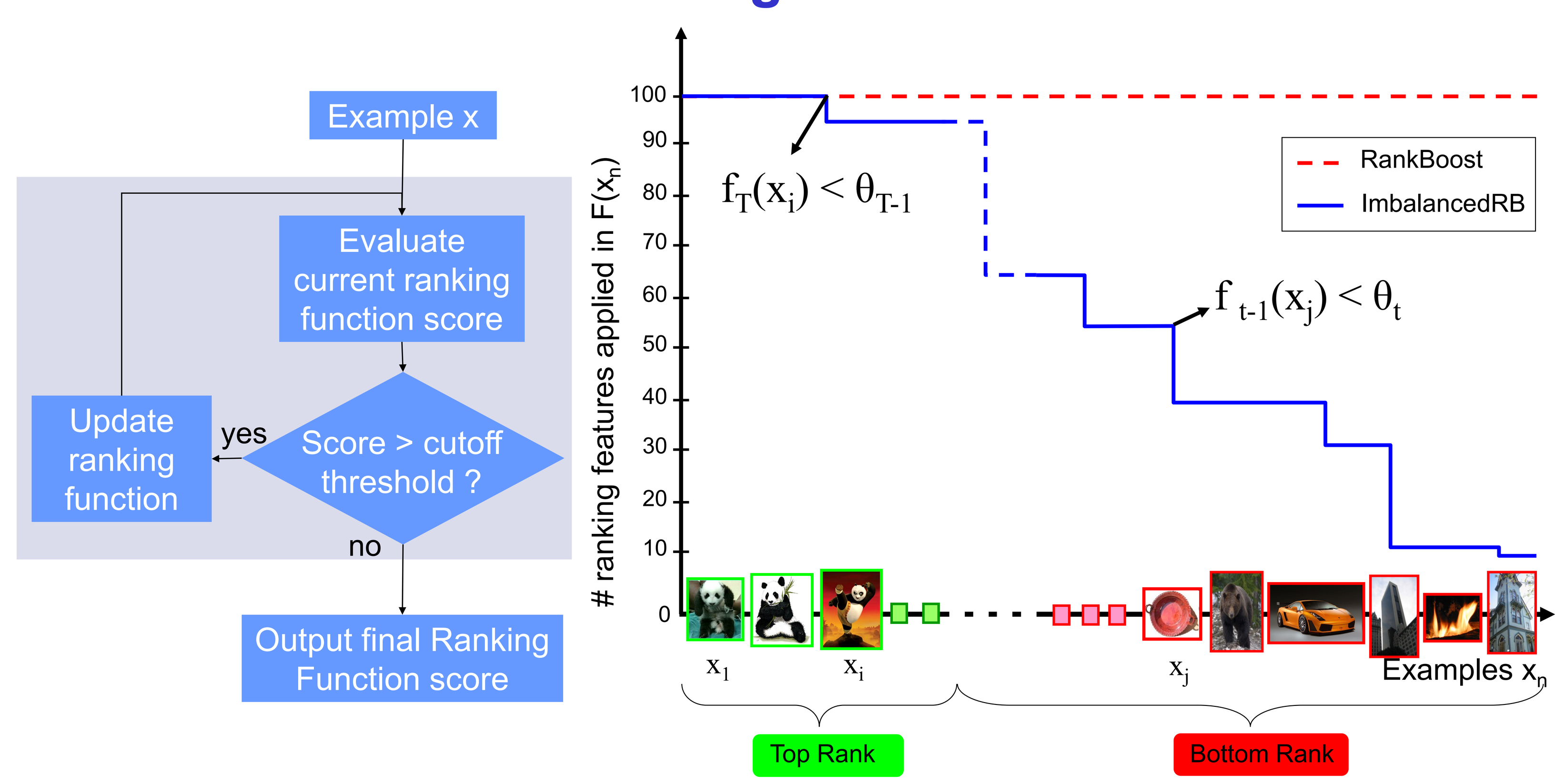
- Goal: minimize exponential rank misclassification error wrt $\alpha_t\ h_t\ \theta_t$

$$L_t = \sum_{X_0, X_1} \exp - \left(f_{t-1}(x_0) - f_{t-1}(x_1) + \alpha_t h_t(x_0) \mathbf{I}(f_{t-1}(x_0) \geq \theta_t) - \alpha_t h_t(x_1) \mathbf{I}(f_t\text{-}1(x_1) \geq \theta_t) \right) + \lambda \Omega(\theta)$$

Learning Process



Ranking Process



Conclusions

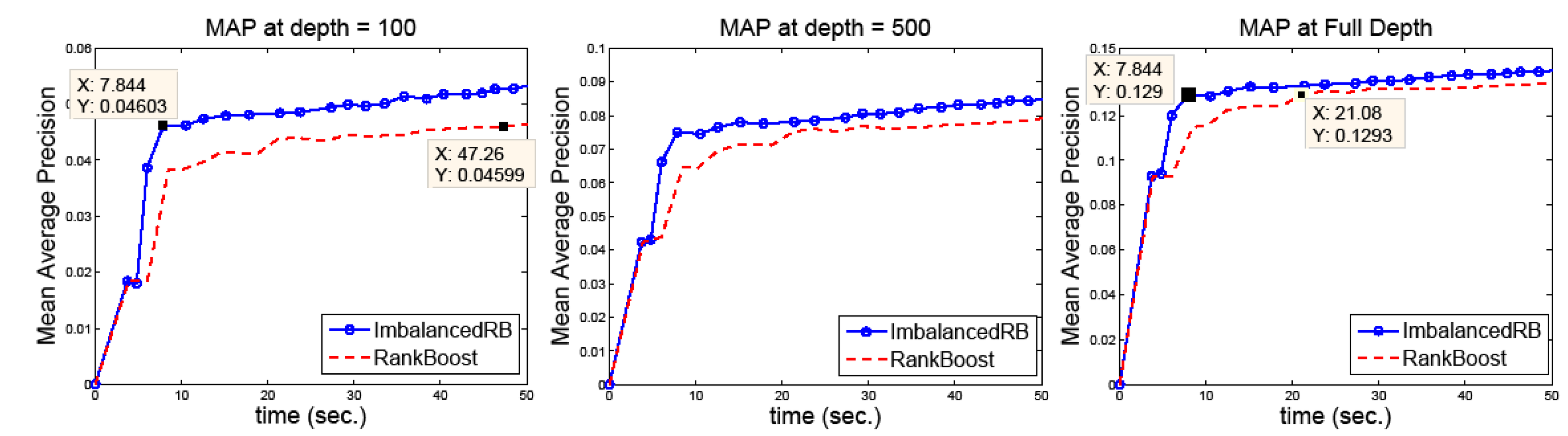
- Imbalanced RankBoost automatically emphasizes top ranked data and truncates computation for less important bottom-ranked ones
- **Ranking process** more **efficient** and **effective** wrt traditional RakBoost
- Future work: incorporate processing time of ranking features into learning process

Experiments

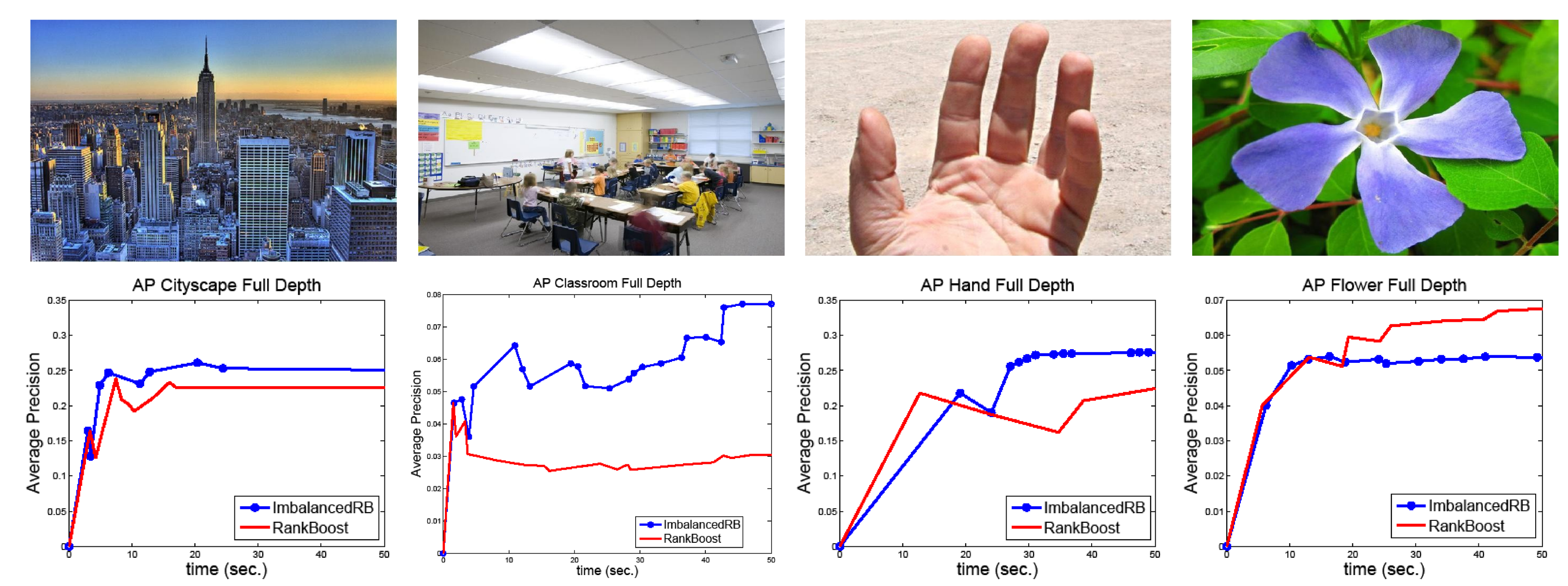
- TRECVID 2007 collection: 52,347 keyframes, 20 concepts
- 200 Ranking Features: RBF kernel SVM trained on bags of data and global features (color histogram, edge histogram, etc.)

RESULTS

- **6-fold speed up in ranking process**
- **7% to 21% relative MAP improvement**
- **Benefit more relevant on top ranked samples (Mean Average Precision computed at limited depth)**

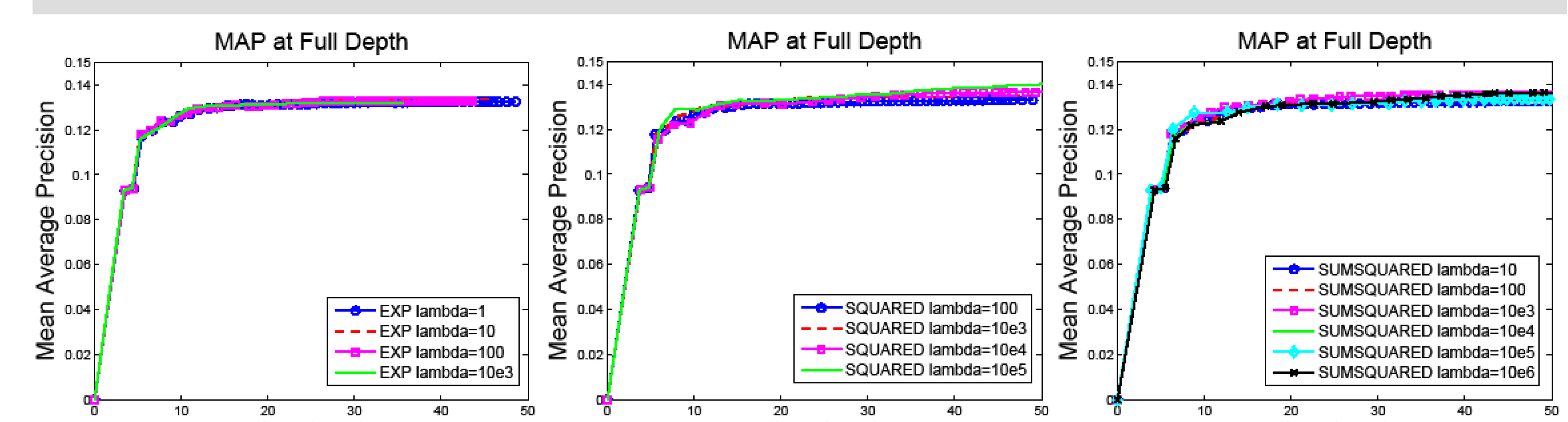


Mean average precision computed as function of the ranking time at top ranked 100 samples (left), top ranked 500 samples (center), full depth (right)



Average precision for single concepts computed as function of the ranking time at full depth

- Choices of reg. term Ω and parameter λ do not affect performances



Analysis of the influence of () to the mean average precision computed as function of the ranking time at full depth