Learning Recommendations in Social Media Systems By Weighting Multiple Relations

Boris Chidlovskii

Xerox Research Centre Europe

Grenoble, France

Outline

Introduction

Social media sharing

Tag recommendation

Relational Graphs

Random walks on relational

Relation weight learning

Evaluation

licensed under (cc) Attribution-NonCommercial-ShareAlike 2.0 Germany | Ludwig Gatzke | http://flickr.com/photos/stabilo-boss/

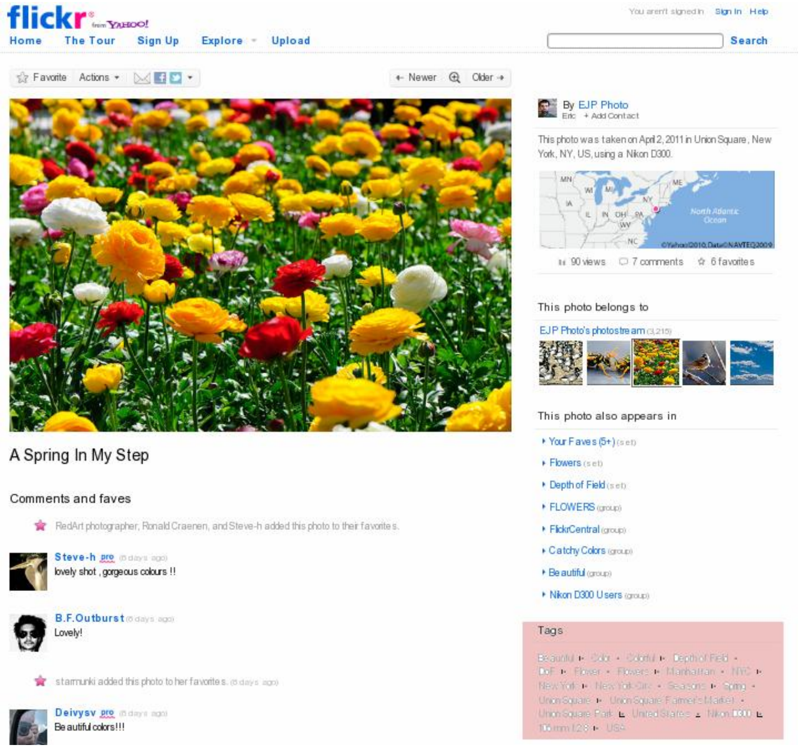
Social media sharing

Social image/video sharing sites

- Flickr: 6B images (Aug 2011)
- Youtube: 48 hours uploaded each minute (Dec 2010)

Each site is a environment with

- Multiple entities and actors
- Multiple activities
  - Navigation, search, tagging, commenting



By EJP Photo
Eric + Add Contact

This photo was taken on Apr 2, 2011 in Union Square, New York, NY, US, using a Nikon D300.

90 views 7 comments 6 favorites

This photo belongs to

EJP Photo's photostream (3,215)

This photo also appears in

- Your Faves (5+) (set)
- Flowers (set)
- Depth of Field (set)
- FLOWERS (group)
- FlickrCentral (group)
- Catchy Colors (group)
- Beautiful (group)
- Nikon D300 Users (group)

Tags

Beautiful • Color • Colorful • Depth of Field • DoF • Flower • Flowers • Manhattan • NYC • New York • New York City • Seasons • Spring • Union Square • Union Square Farmer's Market • Union Square Park • United States • Nikon D300 • 105 mm f2.8 • USA

A Spring In My Step

Comments and faves

RedArt photographer, Ronald Craenen, and Steve-h added this photo to their favorites.

Steve-h pro (6 days ago)
lovely shot , gorgeous colours !!

B.F.Outburst (6 days ago)
Lovely!

starmunki added this photo to her favorites. (6 days ago)

Deivysv pro (6 days ago)
Beautiful colors!!!

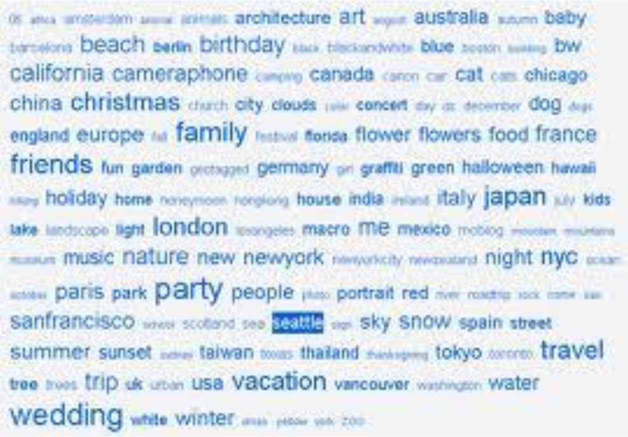
Image tagging and tag recommendation

Flickr tags are both predefined and free text

- Common topics emerge when people agree on the semantic description of objects

Image understanding by content analysis is hard

Tags are great help in image sharing and search



06 africa amsterdam animal animals architecture art august australia autumn baby
barcelona beach berlin birthday black blackandwhite blue boston building bw
california cameraphone camping canada canon car cat cats chicago
china christmas church city clouds color concert day dc december dog dogs
england europe fall family festival florida flower flowers food france
friends fun garden geotagged germany girl graffiti green halloween hawaii
hiking holiday home honeymoon hongkong house india ireland italy japan july kids
lake landscape light london losangeles macro me mexico moblog mountain mountains
museum music nature new newyork newyorkcity newzealand night nyc ocean
october paris park party people photo portrait red river roadtrip rock rome san
sanfrancisco school scotland sea seattle sign sky snow spain street
summer sunset sydney taiwan texas thailand thanksgiving tokyo toronto travel
tree trees trip uk urban usa vacation vancouver washington water
wedding white winter xmas yellow york zoo

Reliable tag recommendation

- Tag bootstrap for a new image
- Tag expansion for an existing image

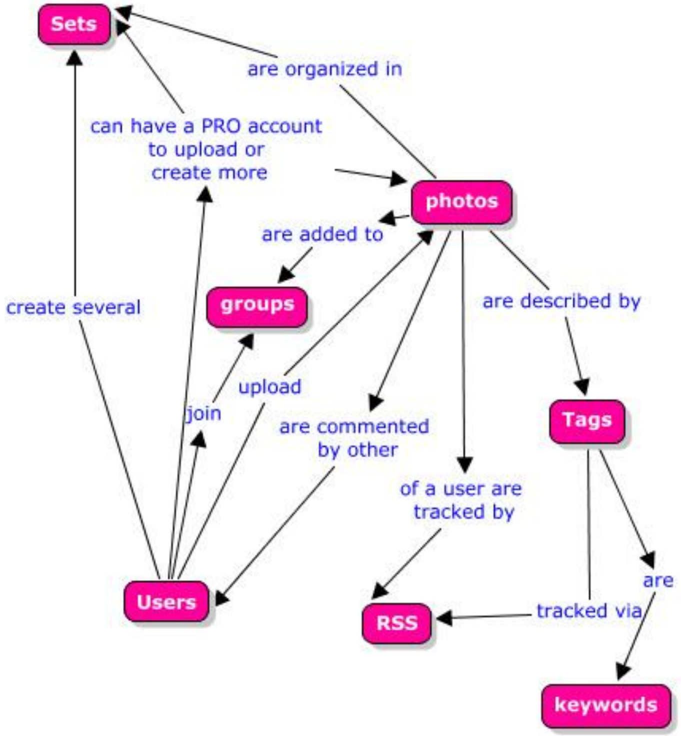
Flickr world

Entity types

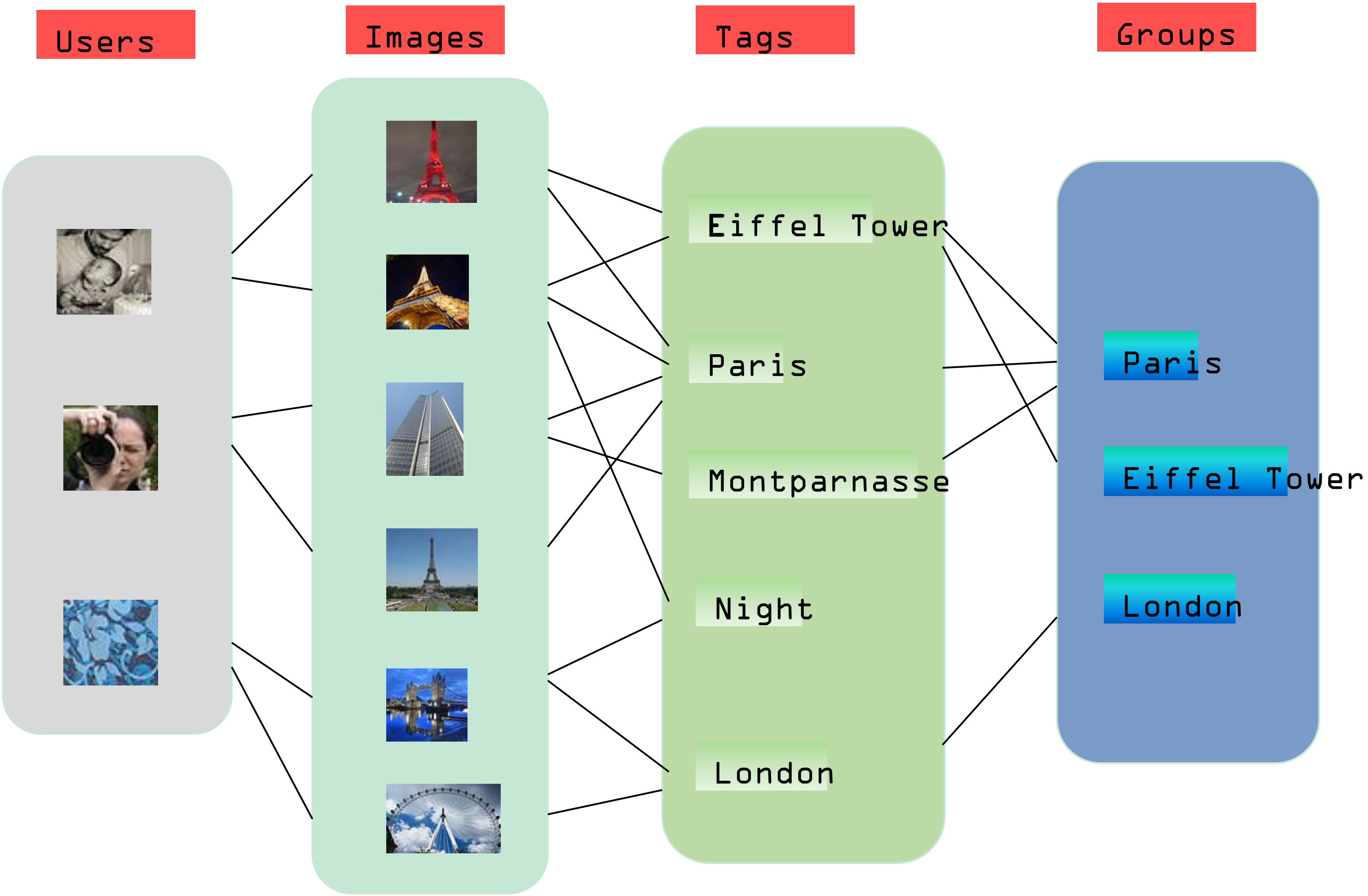
users, photos, tags, keywords, sets, groups, Rss

Multitude of relations

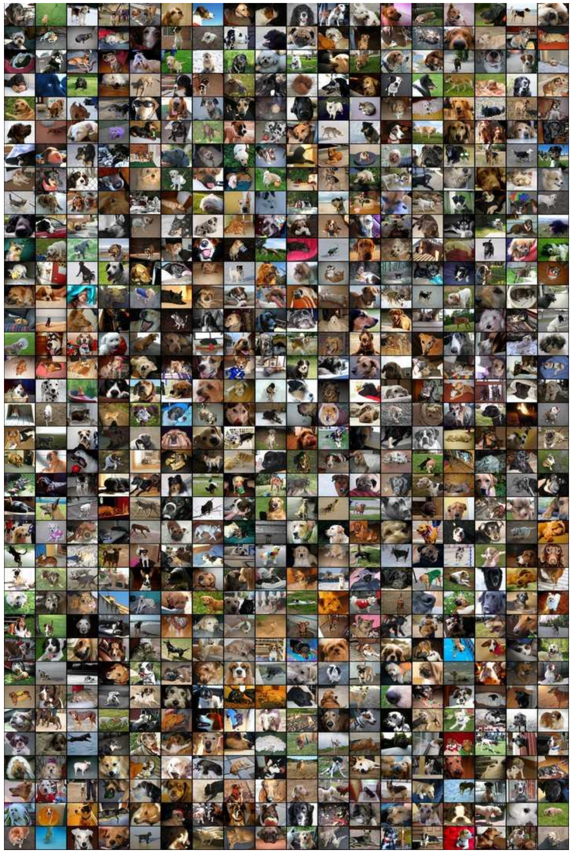
- Users oriented services around entities of different types



Flickr example



Tag recommendation



“Dog” tag images

Challange

- How to benefit from all available relations

Task

- given a recommendation task, determine the optimal combination of relations

Our approach

- We model it as a weighted random walk on the unfolded relational graph
- We minimize a loss function over the weights of relations contributing to the weighted random walk
- Scalability concern

Prior Art

- Recommender systems based on collective social knowledge [Garg08, Overell09, Sigurbj10]
- Random walks on weighted graphs [Toutanova04]
- Recommendation algorithms for the MovieLens collection and Netflix competition
- Label propagation [Zhou06] on the relational graph
- Supervised random walk [Backstrom11]

Outline

- Introduction
- Social media sharing
- Tag recommendation
- Relational Graphs
- Random walks on relational grap
- Relation weight learning
- Evaluation

Relational graph

Represents entity types and relations between them

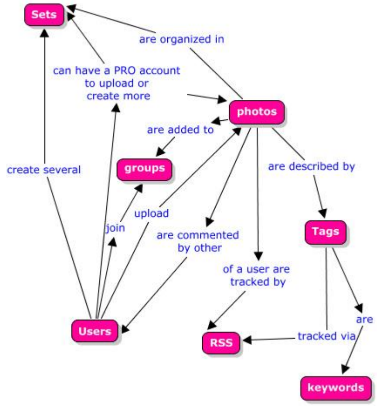
Graph *G* = *(E,R)*, where

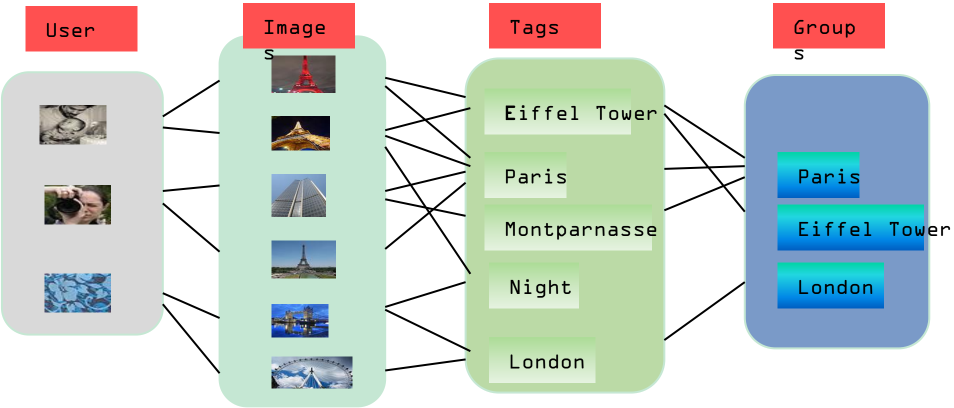
- Entity type $e_k \in E$ is represented as a node
- Relation $r_{kl} \in R$ between entities of types e_k and e_l

Flickr relational graph for 7 entity types:

- {photos, users, tags, groups, keywords, sets, rss}

Unfolded relational graph = instantiation of G





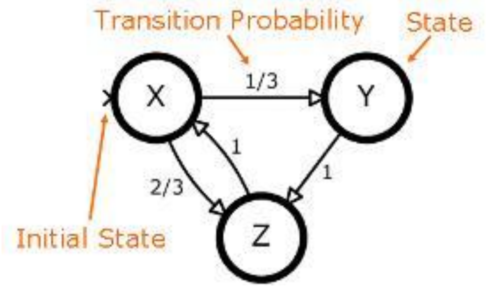
Outline

- Introduction
- Social media sharing
- Tag recommendation
- Relational Graphs
- Random walks on relational grap
- Relation weight learning
- Evaluation

Random walk model

A Markov chain is defined by

- a set of states *S*
- an initial distribution P_0 over *S*
- a set of state transition probabilities $A = P(S_t | S_{t-1})$
  - the Web graph example
- Reset probability α



Stationary distribution of the Markov chain

$$\pi = \alpha \sum_{t=0}^{\infty} (1-\alpha)^t P_0 A^t$$

Relational random walk model

A Relational Markov chain

- Node set *S* of the unfolded relational graph
- Initial distribution P_0 and resetting α
- Walk through the relation $r_{kl} \in R$ contributes with the weight $w_{kl}, 0 \leq w_{kl} \leq 1$

Random walk over the weighted sum of relation probability matrices A_{kl} :

$$A = \sum_{kl} w_{kl} A_{kl}$$

Transition probabilities $A = P(S_t | S_{t-1})$ w_{kl}
are a function of $\sum_k w_{kl} = 1$

Normalization constraint

Projection on tag nodes

Outline

- Introduction
- Social media sharing
- Tag recommendation
- Relational Graphs
- Random walks on relational grap
- Relation weight learning
- Evaluation

Relational weight learning

Approximation of the stationary distribution

- the truncated version, k steps of the random walk
- look for weights minimizing the prediction error on a training set T

Probability estimation loss

- square loss for true *y* and prediction *p*

$$l_{sq}(y,p)=(y-p)^2$$

- Multi-label square loss for true *Y* and prediction *P* vectors

$$L_{sq}(Y,P)=(l_{sq}(y_i,p_i)),i=1,\dots,L$$

Gradient

$$\nabla L_{sq}(Y,P)=(\frac{\partial}{\partial p_i}l_{sq}(y_i,p_i))_{i=1,\dots,L}=2(Y-P)$$

Loss function

- We dispose the training set *T*
  - the image tags are true vectors *Y*
- Define a scoring function to be the empirical loss over the set *T:*

$$Loss(H) = \frac{1}{|T|} \sum_{j \in T} L_{sq}(Y_j, P_j)$$

- Tag prediction vector for image *j*

$$P_j = \alpha \sum_{t=1}^{k} (1-\alpha)^t P_0^j A^t$$

- Loss gradient

$$\frac{\partial Loss(H)}{\partial w_{kl}} = \frac{1}{|T|} \sum_{j \in T} \nabla L_{sq}(Y_j, P_j) \frac{\partial P_j}{\partial w_{kl}}$$

- Chain rule for matrix derivatives

$$\frac{\partial A^t}{\partial w_{kl}} = \frac{\partial (A^{t-1} A)}{\partial w_{kl}} = \frac{\partial A^{t-1}}{\partial w_{kl}} A + A^{t-1} A_{kl}$$

Optimization problem

- The constrained optimization problem

$$\begin{gathered} min_{w_{kl}} Loss(H) \\ s.t. \\ 0 \leq w_{kl} \leq 1 \\ \sum_l w_{kl} = 1, k = 1, \ldots, b. \end{gathered}$$

- Converted into unconstrained one
- Quazi-Newton L-BFGS algorithm for gradient-based optimization
- Feed the derivatives of the loss function with respect to variables w_{kl}

Hessian matrix

- The Hessian matrix for the loss function may help the quasi-Newton method
- Faster convergence to a (generally local) optimum point

$$\frac{\partial Loss(H)}{\partial w_{kl}\partial w_{k'l'}} = \frac{1}{|T|}\sum_{j\in T}\frac{\partial}{\partial P_j}\nabla L_{sq}(Y_j, P_j)\frac{\partial^2 P_j}{\partial w_{kl}\partial w_{k'l'}}$$

- Recursive formula for the second derivatives of matrix series

Outline

- Introduction
- Social media sharing
- Tag recommendation
- Relational Graphs
- Random walks on relational grap
- Relation weight learning
- Evaluation

licensed under (cc) Attribution-NonCommercial-ShareAlike 2.0 Germany | Ludwig Gatzke | http://flickr.com/photos/stabilo-boss/

Evaluation

Python/SciPy/SpaMax implementation

Flickr API to download entities and relations

E={images, tags, users}

Core relations between them

- described_by (image, tag)
- upload (user, image)
- commented_by (user, image)

Inferred relations

- similarity (user, user)

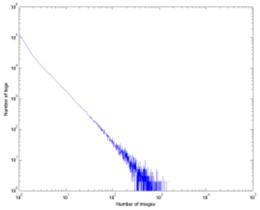
Flickr dataset

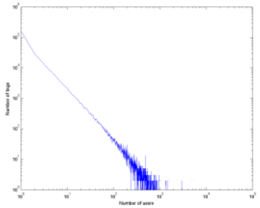
- 100,000 images by 1,951 users
- 127,182 different tags (113,426 after normalization)

Relation	**Min**	**Max**	**Mean**
Image2tag	1	132	5.65
User2image	1	384	27.32
User2user	0	43	1.24

Matrices are sparse and follow the power low distribution

- Preferential attachment in social networks
  - Number of tags per image
  - Number of tags used by a user





Experiment setting

Bootstrap mode

- predict tags for a newly uploaded image

Query mode

- Extend existing tags for an image

Cross-validation for the top 5 tag prediction

Evaluation metrics:

- precision, recall for the multi-label case

Baseline method

- the best un-weighted combination of relations

Alternative loss functions

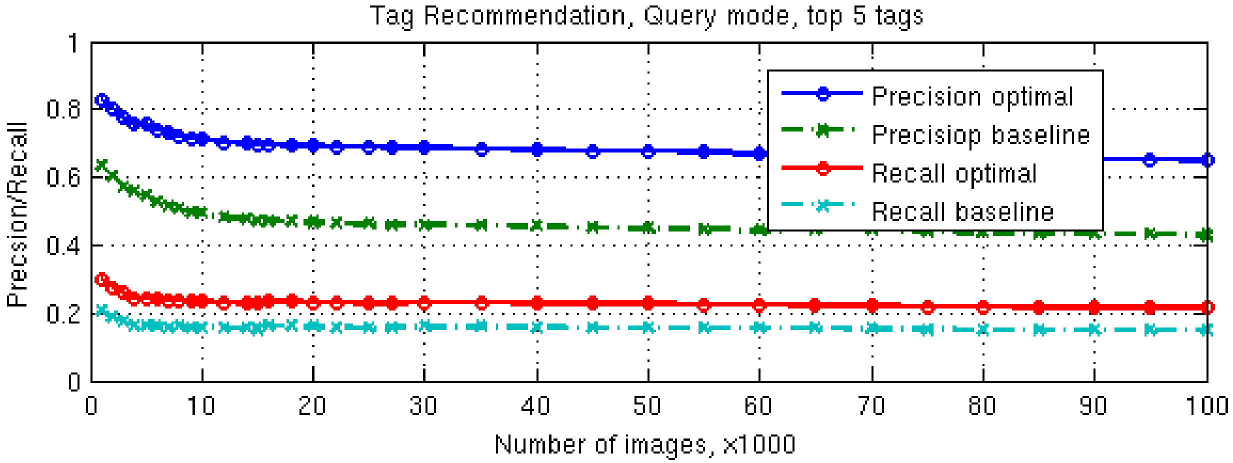
All satisfy the symmetry condition in *y* and *p*

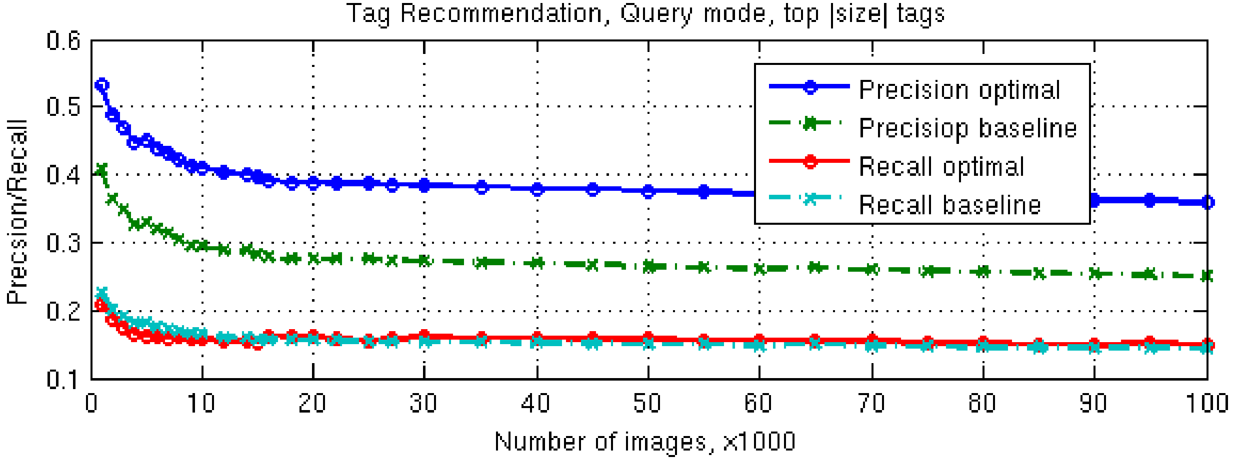
- Absolute loss $l_{abs}(y,p) = |y-p|$
- Exponential loss $l_{exp}(y,p) = exp^{|y-p|} - 1$
- Huber loss $$huber(y,p) = \begin{cases} \frac{|y-p|^2}{2}, & \text{if} |y-p| \leq 0.5 \\ |y-p| - \frac{1}{2}, & \text{otherwise}, \end{cases}$$

All are differentiable but derivatives are not continuous

Tag recommendation

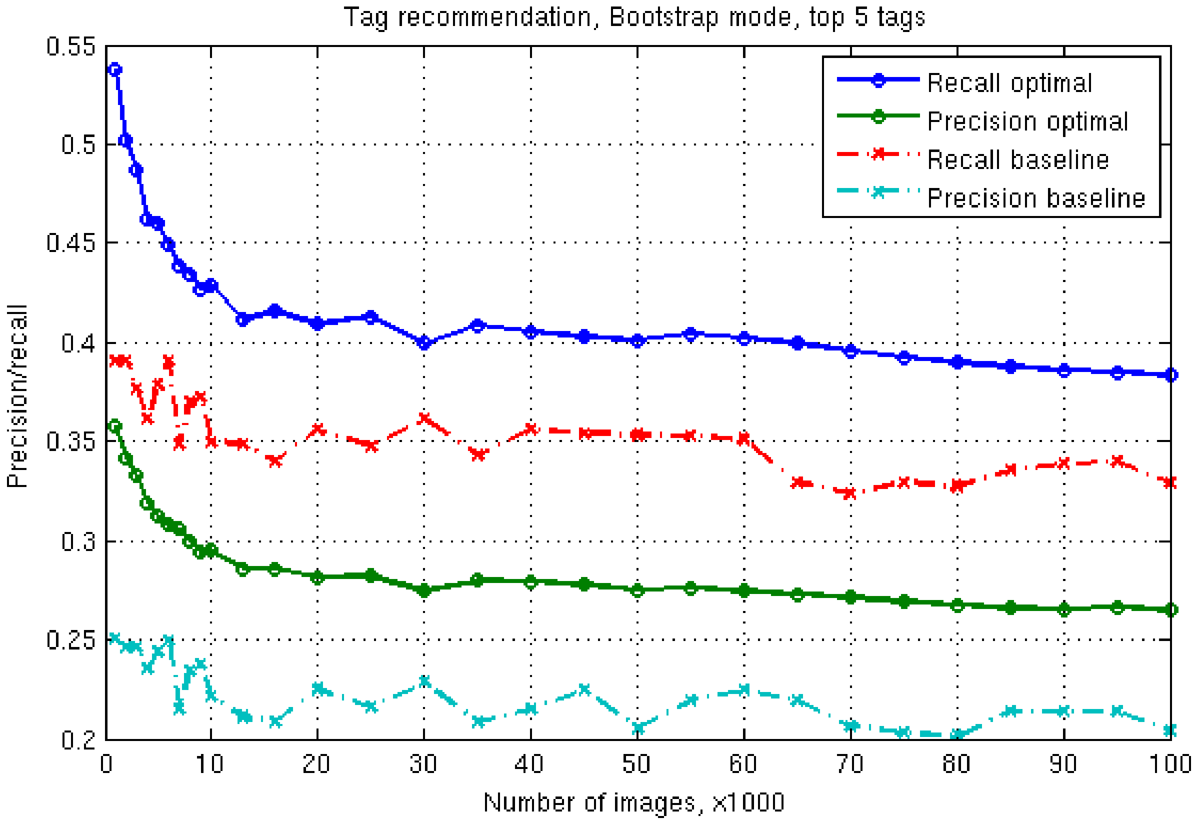
Querying mode





Tag recommendation

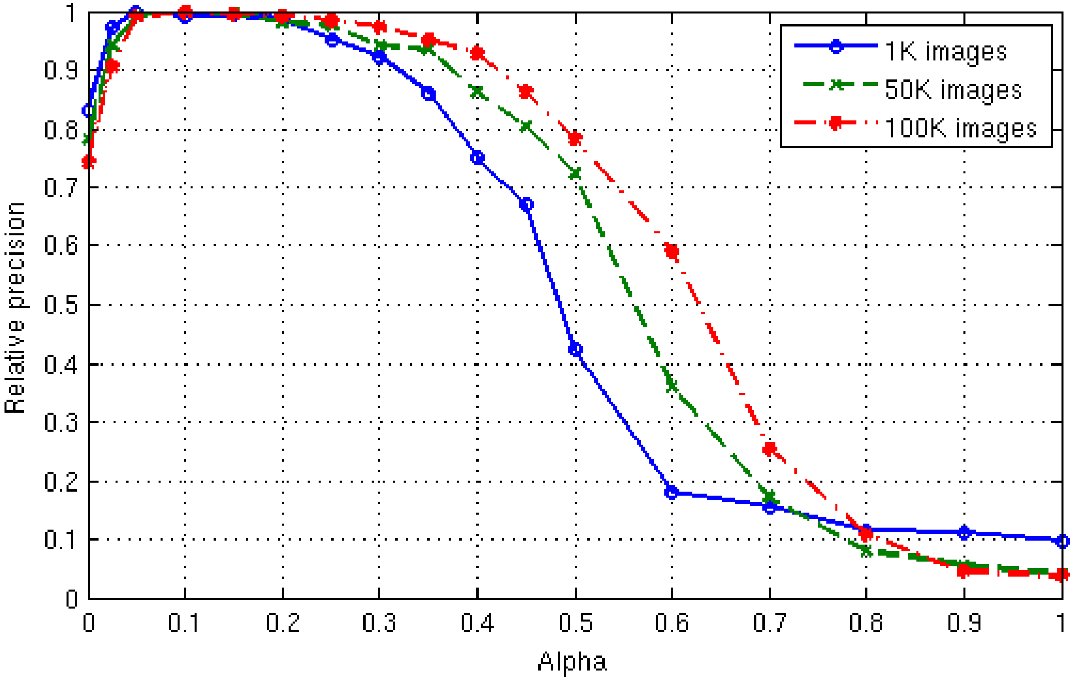
Bootstrap mode



Resetting coefficient effect

Tag recommendation for 1K/50K/100 images

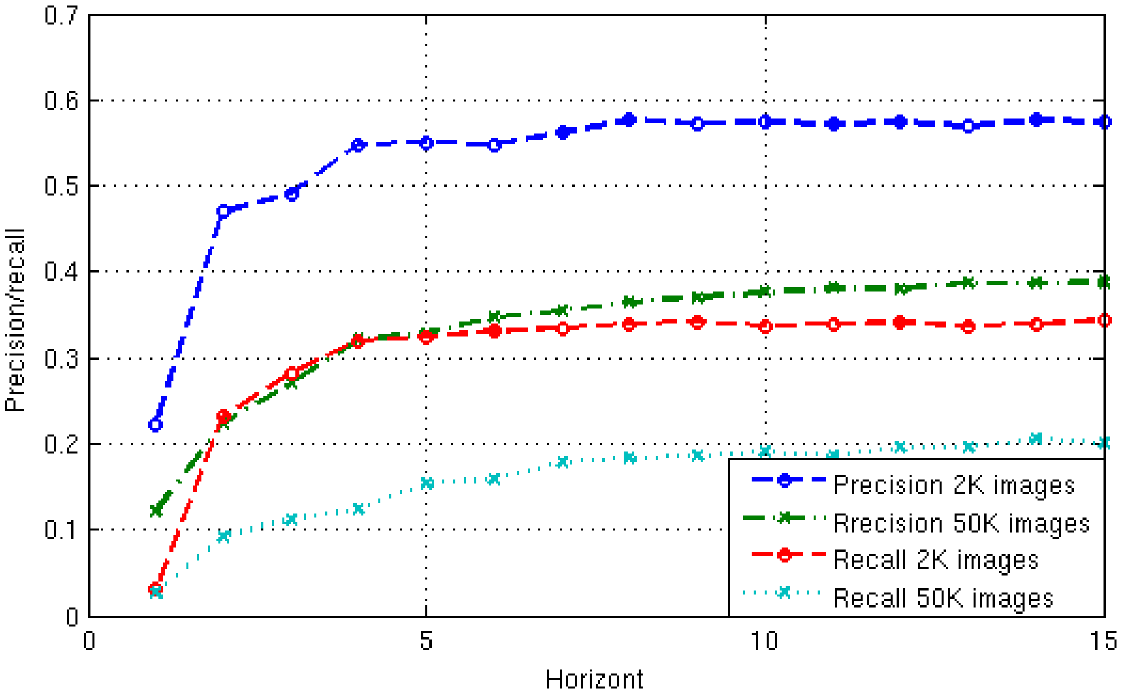
Querying mode



Impact of iteration number

Tag recommendation for 2K/50K images

- Querying mode



Lessons learned (Conclusion)

Relational random walk in the loss minimization form

Scalability is good (thank to the sparse matrices)

MapReduce version (64-cores parallel cluster) is for 12M images

- Flickr relations enriched with image features
- Local context

Hessian is of limited help (< 10K images)

Positioning w.r.t. matrix co-factorization

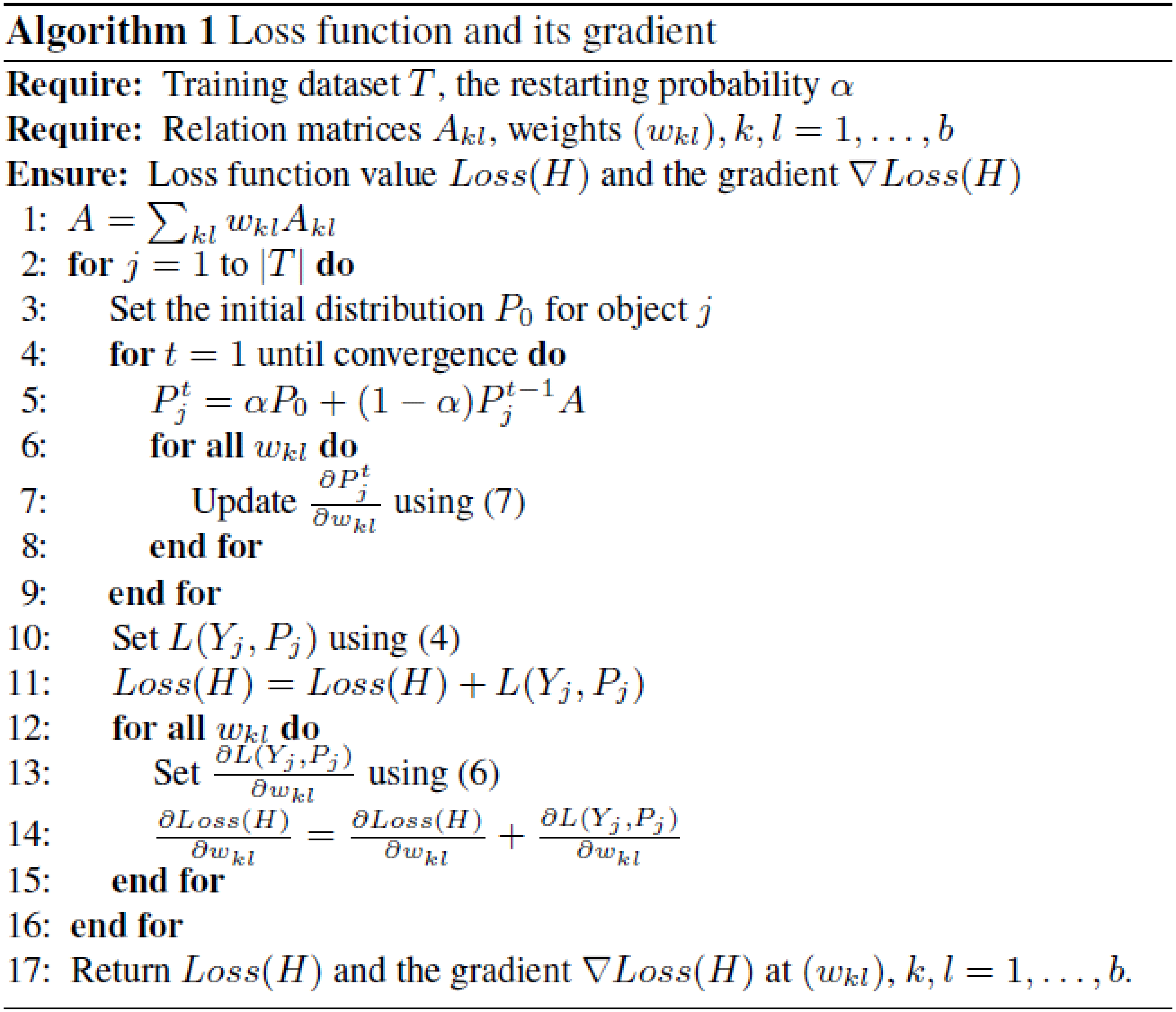
Interest in relation (in)dependence criteria

- Relations on groups and sets are of little

Thank you !

Questions ?

Algorithm 1



Algorithm 1 Loss function and its gradient

```
Require: Training dataset $T$, the restarting probability $\alpha$
Require: Relation matrices $A_{kl}$, weights $(w_{kl}), k, l = 1, \ldots, b$
Ensure: Loss function value $Loss(H)$ and the gradient $\nabla Loss(H)$
$A = \sum_{kl} w_{kl} A_{kl}$
for $j = 1$ to $|T|$ do
    Set the initial distribution $P_0$ for object $j$
    for $t = 1$ until convergence do
        $P_j^t = \alpha P_0 + (1 - \alpha) P_j^{t-1} A$
        for all $w_{kl}$ do
            Update $\frac{\partial P_j^t}{\partial w_{kl}}$ using (7)
        end for
    end for
    Set $L(Y_j, P_j)$ using (4)
    $Loss(H) = Loss(H) + L(Y_j, P_j)$
    for all $w_{kl}$ do
        Set $\frac{\partial L(Y_j, P_j)}{\partial w_{kl}}$ using (6)
        $\frac{\partial Loss(H)}{\partial w_{kl}} = \frac{\partial Loss(H)}{\partial w_{kl}} + \frac{\partial L(Y_j, P_j)}{\partial w_{kl}}$
    end for
end for
Return $Loss(H)$ and the gradient $\nabla Loss(H)$ at $(w_{kl}), k, l = 1, \ldots, b$.
```

Random Walk

The Markov chain model for walking on the relational graph

Defined as the supervised learning problem

- Define the loss function over a training set of examples
- Loss is a function of the weights of relations contributing to the weighted random walk
- Measure metric between estimated and true probability distributions on tags