Relation Regularized Matrix Factorization

Wu-Jun Li, Dit-Yan Yeung

Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China

IJCAI 2009

Contents

Matrix Factorization (MF)

To project instances into a lower-dimensional latent space.

$\mathbf{X} : n \times m$, with each row $\mathbf{X}_{i*}$ denoting an instance

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$$

$$\mathbf{U} : n \times D$$

$$\mathbf{V} : m \times D$$

$$D < m$$

$\mathbf{U}_{i*}$ is the lower-dimensional representation of $\mathbf{X}_{i*}$

- Objective:
  To get a $\mathbf{U}$ which can remove the noise in $\mathbf{X}$
- Example:
  Latent semantic indexing (LSI) for document analysis

Relational Data

Contain both content information and relation (link) structure.

Examples:

- Web pages: page content and hyperlinks
- Research papers: paper content and citations

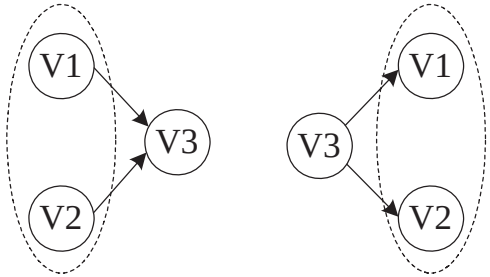
Representation: two matrices

- Content matrix
- Link matrix

Semantics of Relations

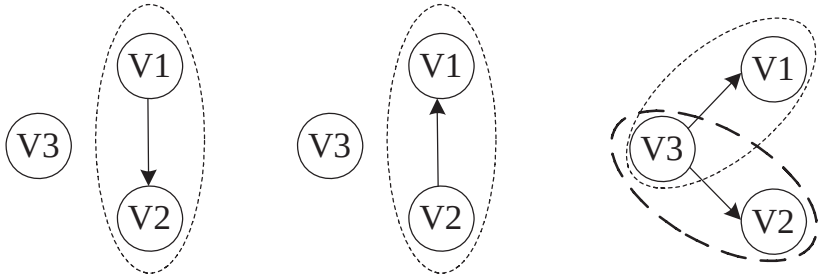
There exist at least two types of links with different semantics:

- Type I Links: If two instances link to or are linked by one common instance, they will be most likely to belong to the same class.



Example: Hyperlinks among web pages

- Type II Links: Two linked instances are most likely to belong to the same class.

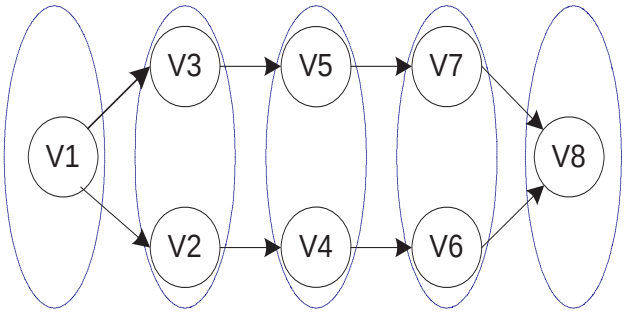


Example: Citations among research papers

Existing Work

- Traditional MF methods:
  - Can only model one matrix
  - Example: LSI
- Joint Link-Content MF (LCMF):
  - Can model both content and link matrices simultaneously
  - Can only model Type I links
  - Illustration:

Link structure



Result of LCMF

-.8	-.5	.3	-.1	-.0
-.0	.4	.6	-.1	-.4
-.0	.4	.6	-.1	-.4
.3	-.2	.3	-.4	.3
.3	-.2	.3	-.4	.3
-.4	.5	.0	-.2	.6
-.4	.5	.0	-.2	.6
-.1	.1	-.4	-.8	-.4

Our Contribution

Relation regularized matrix factorization (RRMF):

- To model Type II links
- Can also model Type I links by preprocessing the link structure
- Convergent
- Linear time-complexity

Notations

- Content matrix: $\mathbf{X} - n \times m$
  $\mathbf{X}_{i*}$: content feature vector for instance i
- Adjacency matrix: $\mathbf{A} - n \times n$
  $A_{ij} = 1$ if there is a relation between instances i and j, and otherwise $A_{ij} = 0$; $A_{ii} = 0$

Note: This specification of $\mathbf{A}$ is only suitable for Type II links. We will introduce the strategy to specify $\mathbf{A}$ for Type I links later.

Objective Function

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \frac{\alpha}{2}(\|\mathbf{U}\|^2 + \|\mathbf{V}\|^2) + \frac{\beta}{2}\mathrm{tr}(\mathbf{U}^T\mathcal{L}\mathbf{U})$$

where $\mathcal{L} = \mathbf{D} - \mathbf{A}$ and $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j A_{ij}$.

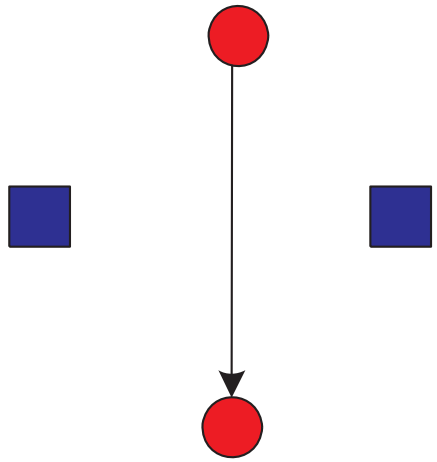
$$\mathrm{tr}(\mathbf{U}^T\mathcal{L}\mathbf{U}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\|\mathbf{U}_{i*} - \mathbf{U}_{j*}\|^2$$

The goal of $\mathrm{tr}(\mathbf{U}^T\mathcal{L}\mathbf{U})$ is to make the latent representations of two instances as close as possible if there exists a relation between them.

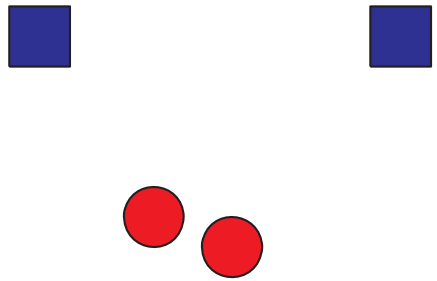
$\Rightarrow$ in line with the semantics of Type II links

Illustration

The original feature representation and link structure:



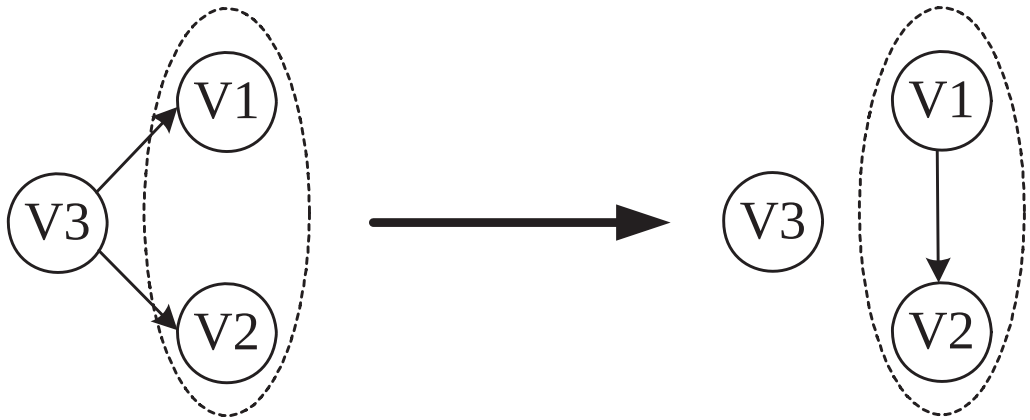
The goal to be achieved by RRMF:



Adapting RRMF for Type I Links

- Basic idea:
  To transform Type I links to Type II links

- Strategy:
  Artificially add a link between two instances if they link to or are linked by a common instance.



Type I links Type II links

Convexity of the Objective Function

Let $f = \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \frac{\alpha}{2}(\|\mathbf{U}\|^2 + \|\mathbf{V}\|^2) + \frac{\beta}{2}\mathrm{tr}(\mathbf{U}^T\mathcal{L}\mathbf{U})$.

Theorem

f is convex w.r.t. $\mathbf{U}$.

Theorem

f is convex w.r.t. $\mathbf{V}$.

Alternating Projection Method

Each time we fix one parameter and then update the other one.

Iterate until some termination condition is satisfied:

- Learn **U** with **V** fixed
- Learn **V** with **U** fixed

Learning U

To optimize one column $\mathbf{U}_{*d}$ at a time with the other columns fixed:

$$\mathbf{F}^{(d)}\mathbf{U}_{*d} = \mathbf{e}^{(d)}$$

$$\begin{aligned}
\mathbf{F}^{(d)} &= \mathbf{E}^{(d)} + \alpha\mathbf{I} + \beta\boldsymbol{\mathcal{L}} \\
\mathbf{E}^{(d)} &= \mathrm{diag}\left(\frac{\partial^2 g}{\partial U_{1d}\partial U_{1d}}, \frac{\partial^2 g}{\partial U_{2d}\partial U_{2d}}, \ldots, \frac{\partial^2 g}{\partial U_{nd}\partial U_{nd}}\right), \frac{\partial^2 g}{\partial U_{id}\partial U_{id}} = \sum_{j=1}^{m} V_{jd}^2 \\
\mathbf{e}^{(d)} &= (e_1^{(d)}, e_2^{(d)}, \ldots, e_n^{(d)})^T, e_i^{(d)} = \sum_{j=1}^{m} V_{jd}(X_{ij} - \mathbf{U}_{i*}\mathbf{V}_{j*}^T + U_{id}V_{jd})
\end{aligned}$$

Steepest descent to iteratively update $\mathbf{U}_{*d}$:

$$\begin{aligned}
r(t) &= \mathbf{e}^{(d)} - \mathbf{F}^{(d)}\mathbf{U}_{*d}(t) \\
\delta(t) &= \frac{r(t)^T r(t)}{r(t)^T \mathbf{F}^{(d)} r(t)} \\
\mathbf{U}_{*d}(t+1) &= \mathbf{U}_{*d}(t) + \delta(t) r(t)
\end{aligned}$$

Learning $\mathbf{V}$

The update of the whole matrix $\mathbf{V}$ can naturally be decomposed into the update of each row $\mathbf{V}_{j*}$:

$$\begin{aligned} \mathbf{V}_{j*} &= \left(\sum_{i=1}^{n} X_{ij} \mathbf{U}_{i*} \right) \mathbf{K}^{-1} \\ \mathbf{K} &= \sum_{i=1}^{n} \mathbf{U}_{i*}^{T} \mathbf{U}_{i*} + \alpha \mathbf{I} \end{aligned}$$

Convergence and Complexity

Theorem

The learning algorithm will converge.

The time complexity of the learning algorithm is $O(n)$.

Data Sets

- WebKB:
  - Web pages from the CS departments of 4 universities: Cornell, Texas, Washington, Wisconsin
  - 7-class problem: a page belongs to one of the {student, professor, course, project, staff, department or "others"}.
- Cora:
  - Research papers with their bibliographic citations
  - Each paper is labeled as one of the subfields of data structure (DS), hardware and architecture (HA), machine learning (ML), and programming language (PL).

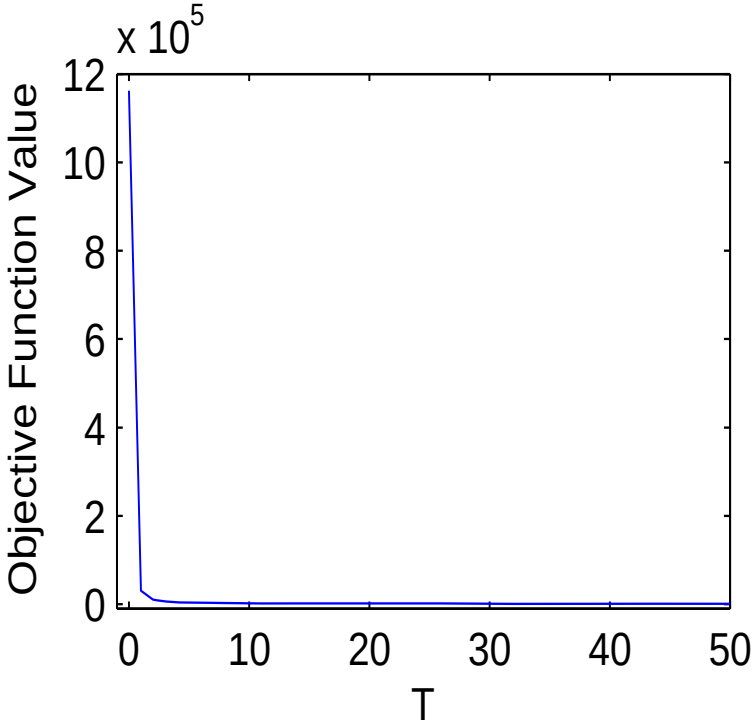
Characteristics of the WebKB data set

	#classes	#instances	#terms
Cornell	7	827	4,134
Texas	7	814	4,029
Washington	7	1,166	4,165
Wisconsin	6	1,210	4,189

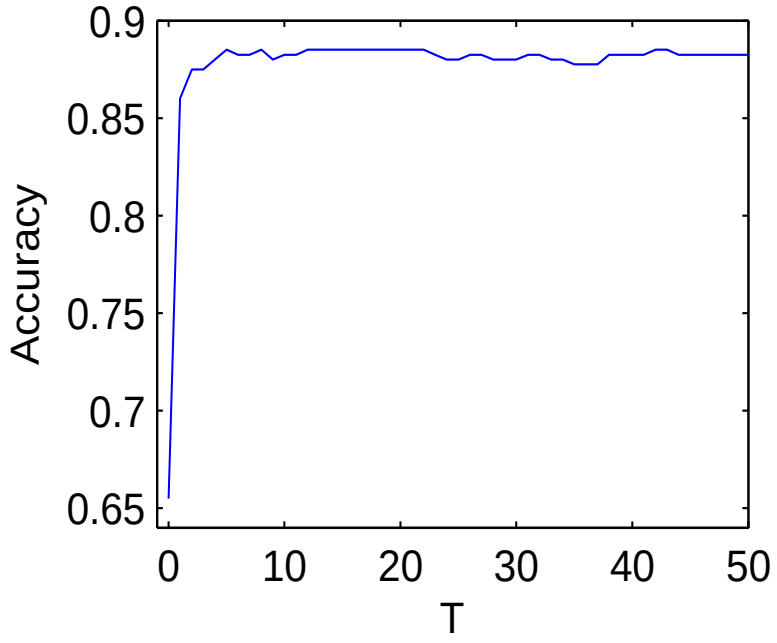
Characteristics of the Cora data set

	#classes	#instances	#terms
DS	9	751	6,234
HA	7	400	3,989
ML	7	1,617	8,329
PL	9	1,575	7,949

Convergence Speed

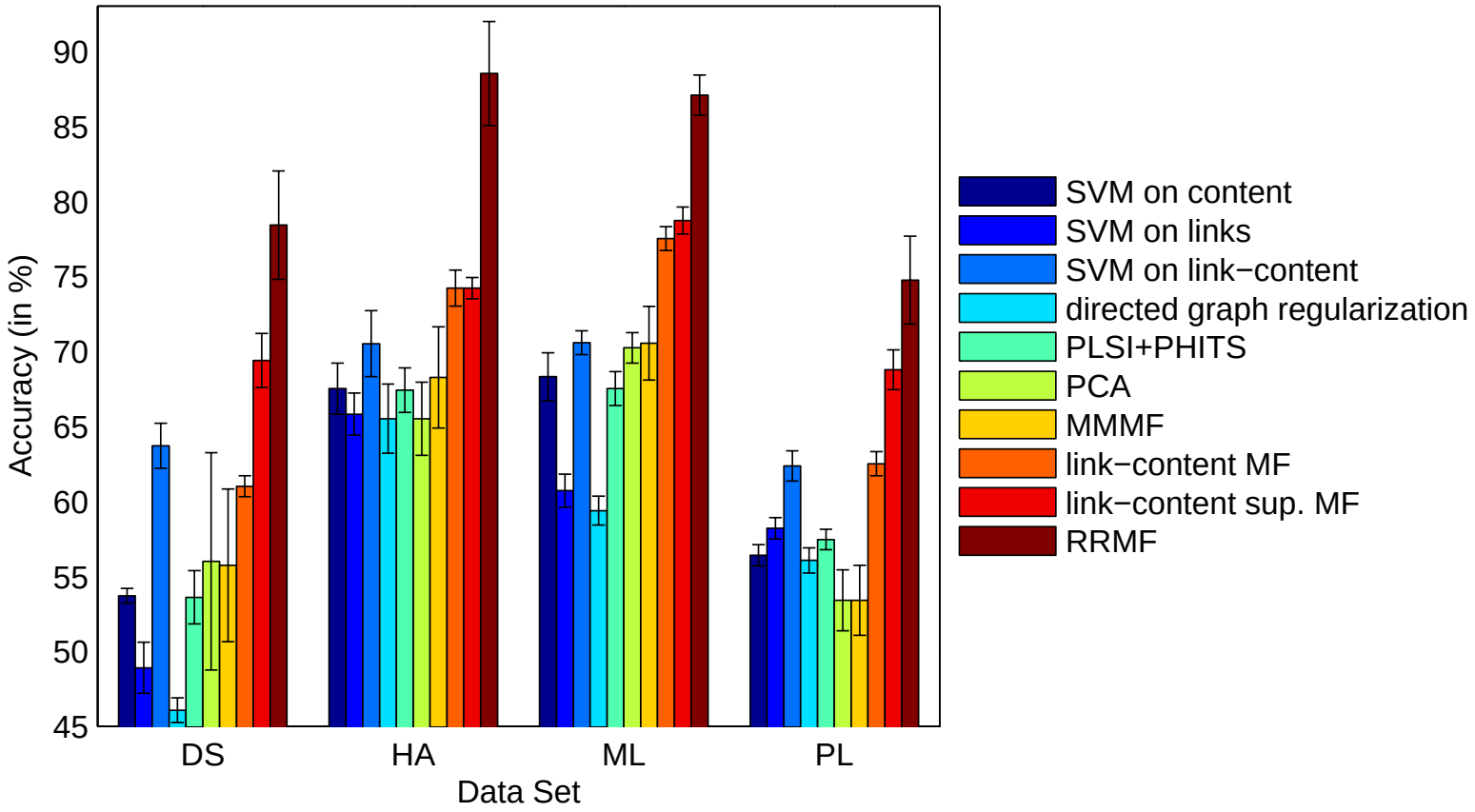


(a) Objective function

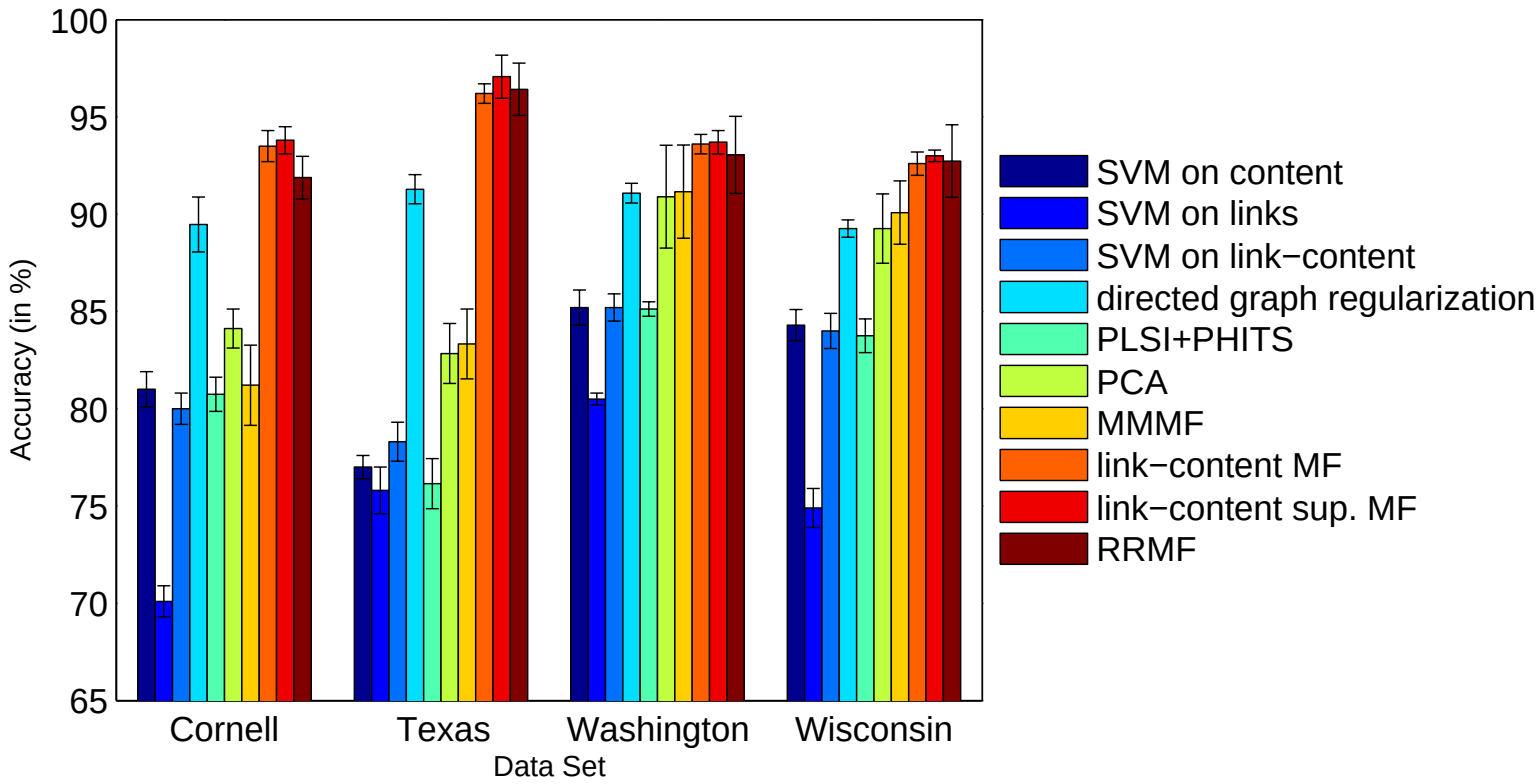


(b) Accuracy

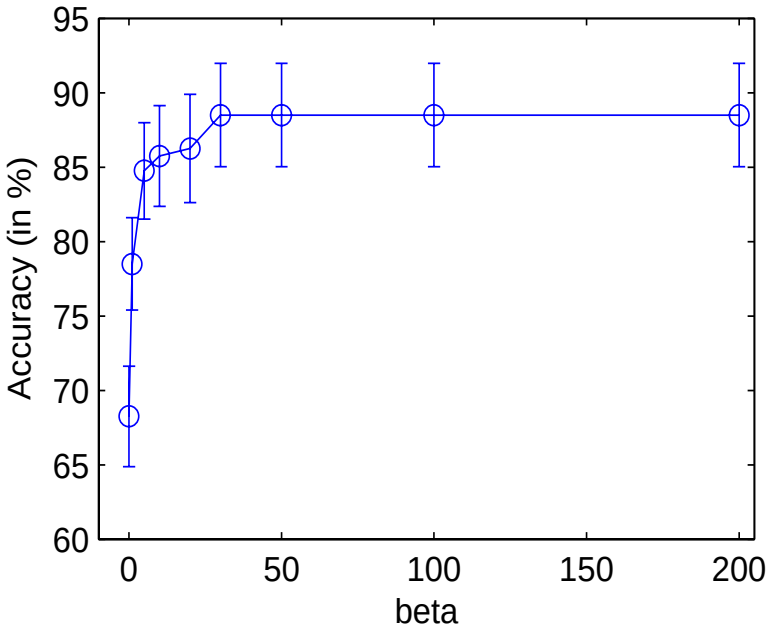
Performance on Cora



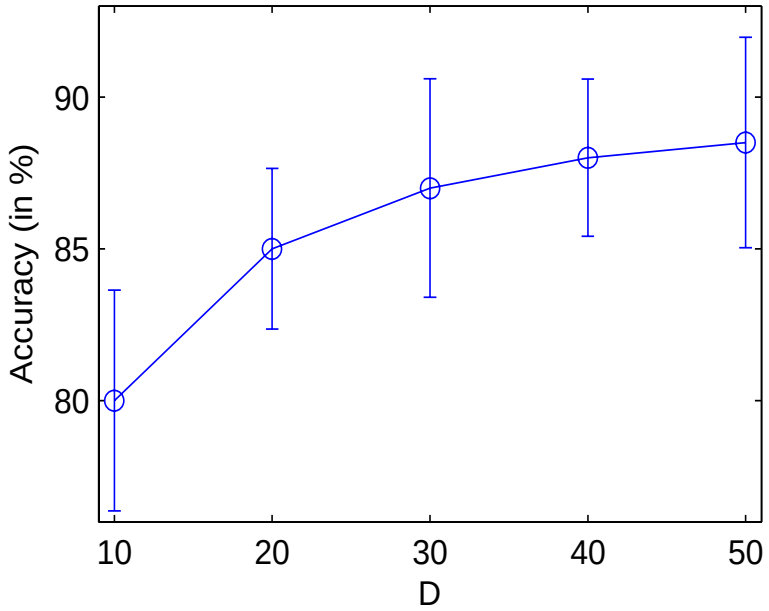
Performance on WebKB



Sensitivity to Parameters



(a) Effect of β



(b) Effect of D

Main Contributions

- RRMF seamlessly integrates both relation and content information
- RRMF achieves state-of-the-art performance.
- RRMF is scalable to large-scale problems.
- RRMF is convergent and very stable.

Code

MATLAB code and data can be downloaded from:

http://www.cse.ust.hk/~liwujun