Latent Wishart Processes for Relational Kernel Learning

Wu-Jun Li

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

Joint work with Zhihua Zhang and Dit-Yan Yeung

Contents

Relational Learning

- Traditional machine learning models:
  - Assumption: i.i.d.
  - Advantage: simple
- Many real-world applications:
  - Relational: instances are related (linked) to each other
  - Autocorrelation: statistical dependency between the values of a random variable on related objects (non i.i.d.)
  - E.g., web pages, protein-protein interaction data
- Relational learning:
  - An emerging research area attempting to represent, reason, and learn in domains with complex relational structure [Getoor & Taskar, 2007].
- Application areas:
  - Web mining, social network analysis, bioinformatics, marketing, etc.

Relational Kernel Learning

- Kernel function:
  To characterize the similarity between data instances:
  - $K(\mathbf{x}_i, \mathbf{x}_j)$
    e.g., $K(\text{cat}, \text{tiger}) > K(\text{cat}, \text{elephant})$
  - Positive semidefiniteness (p.s.d.)

- Kernel learning:
  To learn an appropriate kernel matrix or kernel function for a kernel-based learning method.

- Relational kernel learning (RKL):
  To learn an appropriate kernel matrix or kernel function for relational data by incorporating relational information between instances into the learning process.

Stochastic Processes and Gaussian Processes

- Stochastic processes:
  A stochastic process (or random process) $y(\mathbf{x})$ is specified by giving the joint distribution for any finite set of instances $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in a consistent manner.

- Gaussian processes:
  - A Gaussian process is a distribution over functions $y(\mathbf{x})$ s.t. the values of $y(\mathbf{x})$ evaluated at an arbitrary set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ jointly have a Gaussian distribution.
  - Assuming $y(\mathbf{x})$ has zero mean, the specification of a Gaussian process is completed by giving the covariance function of $y(\mathbf{x})$ evaluated at any two values of $\mathbf{x}$, given by the kernel function $K(\cdot, \cdot)$:

$$\mathbb{E}[y(\mathbf{x}_i)y(\mathbf{x}_j)] = K(\mathbf{x}_i, \mathbf{x}_j).$$

Wishart Processes

- Wishart distribution:
  An $n \times n$ random symmetric positive definite matrix $\mathbf{A}$ is said to have a Wishart distribution with parameters n, q, and $n \times n$ scale matrix $\mathbf{\Sigma} \succ 0$, written as $\mathbf{A} \sim W_n(q, \mathbf{\Sigma})$, if its p.d.f. is given by

$$\frac{|\mathbf{A}|^{(q-n-1)/2}}{2^{qn/2}\,\Gamma_n(q/2)\,|\mathbf{\Sigma}|^{q/2}} \exp\Big(-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{A})\Big),\ q \geq n.$$

  Here $\mathbf{\Sigma} \succ 0$ means that $\mathbf{\Sigma}$ is positive definite (p.d.).

- Wishart processes:
  Given an input space $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$, the kernel function $\{A(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}\}$ is said to be a Wishart process (WP) if for any $n \in \mathbb{N}$ and $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathcal{X}$, the $n \times n$ random matrix $\mathbf{A} = [A(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^n$ follows a Wishart distribution.

Relationship between GP and WP

- For any kernel function $A : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a function $B : \mathcal{X} \to \mathcal{F}$ s.t. $A(\mathbf{x}_i, \mathbf{x}_j) = B(\mathbf{x}_i)'B(\mathbf{x}_j)$, where $\mathcal{X}$ is the input space and $\mathcal{F} \subset \mathbb{R}^q$ is some latent (feature) space (in general the feature space may also be infinite-dimensional).
- Our previous result: $A(\mathbf{x}_i, \mathbf{x}_j)$ is a Wishart process iff $\{B_k(\mathbf{x})\}_{k=1}^q$ are q mutually independent Gaussian processes.
- Let $\mathbf{A} = [A(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^n$ and $\mathbf{B} = [B(\mathbf{x}_1), \ldots, B(\mathbf{x}_n)]' = [\mathbf{b}_1, \ldots, \mathbf{b}_n]'$. Then $\mathbf{b}_i$ are the latent vectors, and $\mathbf{A} = \mathbf{B}\mathbf{B}'$ is a linear kernel in the latent space but is a nonlinear kernel w.r.t. the input space.

Theorem

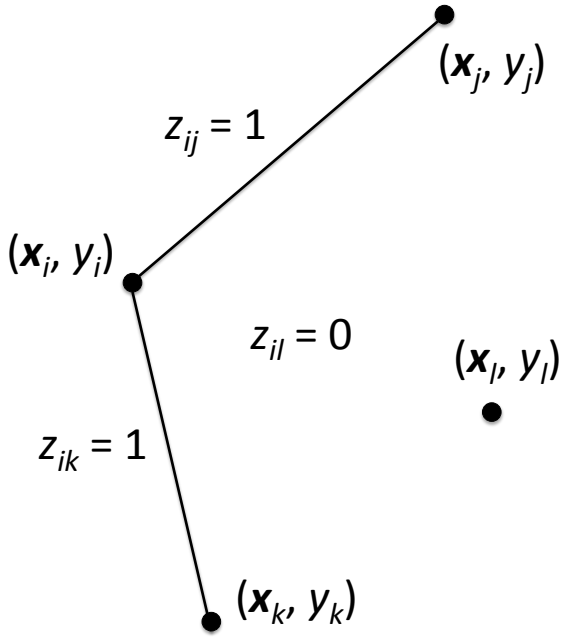
Let $\mathbf{\Sigma}$ be an $n \times n$ positive definite matrix. Then $\mathbf{A}$ is distributed according to the Wishart distribution $W_n(q, \mathbf{\Sigma})$ if and only if $\mathbf{B}$ is distributed according to the (matrix-variate) Gaussian distribution $N_{n,q}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{I}_q)$.

GP and WP in a Nutshell

- Gaussian distribution:
  Each sampled instance is a finite-dimensional vector, $\mathbf{v} = (v_1, \ldots, v_d)'$.
- Wishart distribution:
  Each sampled instance is a finite-dimensional p.s.d. matrix, $\mathbf{M} \succeq 0$.
- Gaussian process:
  Each sampled instance is an infinite-dimensional function, $f(\cdot)$.
- Wishart process:
  Each sampled instance is an infinite-dimensional p.s.d. function, $g(\cdot, \cdot)$.

Relational Data

- $\{(\mathbf{x}_i, y_i, z_{ik}) \mid i, k = 1, \ldots, n\}$



- $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$: input feature vector for instance i
- y_i: label for instance i
- $z_{ik} = 1$ if there exists a link between $\mathbf{x}_i$ and $\mathbf{x}_k$; 0 otherwise. $z_{ik} = z_{ki}$ and $z_{ii} = 0$. $\mathbf{Z} = [z_{ik}]_{i,k=1}^{n}$.

LWP Model

- Goal:
  To learn a target kernel function $A(\mathbf{x}_i, \mathbf{x}_k)$ which takes both the input attributes and the relational information into consideration.

- LWP:
  Let $a_{ik} = A(\mathbf{x}_i, \mathbf{x}_k)$.
  Then $\mathbf{A} = [a_{ik}]_{i,k=1}^{n}$ is a latent p.s.d. matrix.
  We model $\mathbf{A}$ by a Wishart distribution $W_n(q, \mathbf{\Sigma})$, which implies that $A(\mathbf{x}_i, \mathbf{x}_k)$ follows a Wishart process.

LWP Model

- Prior:

$$p(\mathbf{A}) = W_n(q, \beta(\mathbf{K} + \lambda \mathbf{I})),$$

where $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_k)]_{i,k=1}^n$ with $K(\mathbf{x}_i, \mathbf{x}_k)$ being a kernel function defined on the input attributes, $\beta > 0$, and λ is a very small number to make $\mathbf{\Sigma} \succ 0$.

- Likelihood:

$$p(\mathbf{Z}|\mathbf{A}) = \prod_{i=1}^{n} \prod_{k=i+1}^{n} s_{ik}^{z_{ik}} (1 - s_{ik})^{1 - z_{ik}} \quad \text{with} \quad s_{ik} = \frac{\exp(a_{ik}/2)}{1 + \exp(a_{ik}/2)}.$$

- Posterior:

$$p(\mathbf{A}|\mathbf{Z}) \propto p(\mathbf{Z}|\mathbf{A}) p(\mathbf{A})$$

The input attributes and relational information are seamlessly integrated via the Bayesian approach.

Maximum A Posteriori (MAP) Estimation

- Optimization via MAP estimation:

$$\underset{\mathbf{A}}{\operatorname{argmax}} \log\left[p(\mathbf{Z}|\mathbf{A})p(\mathbf{A})\right]$$

- The theorem shows that finding the MAP estimate of $\mathbf{A}$ is equivalent to finding the MAP estimate of $\mathbf{B}$. Hence, we maximize the following:

$$\begin{aligned} L(\mathbf{B}) &= \log\{p(\mathbf{Z}|\mathbf{B})p(\mathbf{B})\} = \sum_{i\neq k} \log p(z_{ik}|\mathbf{b}_i,\mathbf{b}_k) + \log p(\mathbf{B}) \\ &= \sum_{i\neq k}\left[\frac{z_{ik}\mathbf{b}_i'\mathbf{b}_k}{2} - \log(1+\exp(\frac{\mathbf{b}_i'\mathbf{b}_k}{2}))\right] - \frac{1}{2}\mathrm{tr}\left[\frac{(\mathbf{K}+\lambda\mathbf{I})^{-1}}{\beta}\mathbf{B}\mathbf{B}'\right] + C \\ &= \sum_{i\neq k}\left[z_{ik}\mathbf{b}_i'\mathbf{b}_k/2 - \log(1+\exp(\mathbf{b}_i'\mathbf{b}_k/2))\right] - \frac{1}{2}\sum_{i,k}\sigma_{ik}\mathbf{b}_i'\mathbf{b}_k + C, \end{aligned}$$

where $[\sigma_{ik}]_{i,k=1}^{n} = \frac{(\mathbf{K}+\lambda\mathbf{I})^{-1}}{\beta}$ and C is a constant independent of $\mathbf{B}$.

MAP Estimation

Block quasi-Newton method to solve the maximization of $L(\mathbf{B})$ w.r.t. $\mathbf{B}$:

- Fisher score vector and Hessian matrix of L w.r.t. $\mathbf{b}_i$:

$$\frac{\partial L}{\partial \mathbf{b}_i} = \sum_{j \neq i} (z_{ij} - s_{ij} - \sigma_{ij}) \mathbf{b}_j - \sigma_{ii} \mathbf{b}_i$$

$$\frac{\partial^2 L}{\partial \mathbf{b}_i \partial \mathbf{b}_i'} = -\frac{1}{2} \sum_{j \neq i} s_{ij}(1 - s_{ij}) \mathbf{b}_j \mathbf{b}_j' - \sigma_{ii} \mathbf{I}_q \triangleq -\mathbf{H}_i.$$

- Update equations:

$$\mathbf{b}_i(t+1) = \mathbf{b}_i(t) + \gamma \, \mathbf{H}_i(t)^{-1} \left. \frac{\partial L}{\partial \mathbf{b}_i} \right|_{\mathbf{B} = \mathbf{B}(t)}, \quad i = 1, \ldots, n,$$

where γ is the step size.

Embedding for Test Data

- Let $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} \\ \mathbf{Z}_{21} & \mathbf{Z}_{22} \end{bmatrix}$ and $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix}$, where $\mathbf{Z}_{11}, \mathbf{\Sigma}_{11}$ are $n_1 \times n_1$ matrices and $\mathbf{Z}_{22}, \mathbf{\Sigma}_{22}$ are $n_2 \times n_2$ matrices.

  The n_1 instances corresponding to $\mathbf{Z}_{11}, \mathbf{\Sigma}_{11}$ are training data and the n_2 instances corresponding to $\mathbf{Z}_{22}, \mathbf{\Sigma}_{22}$ are new test data.

- Similarly, we partition $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1\mathbf{B}_1' & \mathbf{B}_1\mathbf{B}_2' \\ \mathbf{B}_2\mathbf{B}_1' & \mathbf{B}_2\mathbf{B}_2' \end{bmatrix}$,

  $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$.

- Because $\mathbf{B} \sim N_{n,q}(\mathbf{0},\ \mathbf{\Sigma} \otimes \mathbf{I}_q)$, we have $\mathbf{B}_1 \sim N_{n_1,q}(\mathbf{0},\ \mathbf{\Sigma}_{11} \otimes \mathbf{I}_q)$ and

$$\mathbf{B}_2 \,|\, \mathbf{B}_1 \sim N_{n_2,q}\left(\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{B}_1,\ \mathbf{\Sigma}_{22\cdot 1} \otimes \mathbf{I}_q\right),$$

  where $\mathbf{\Sigma}_{22\cdot 1} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}$.

Comparison with RGP [Chu *et al.*, 2007] and XGP [Silva *et al.*, 2008]

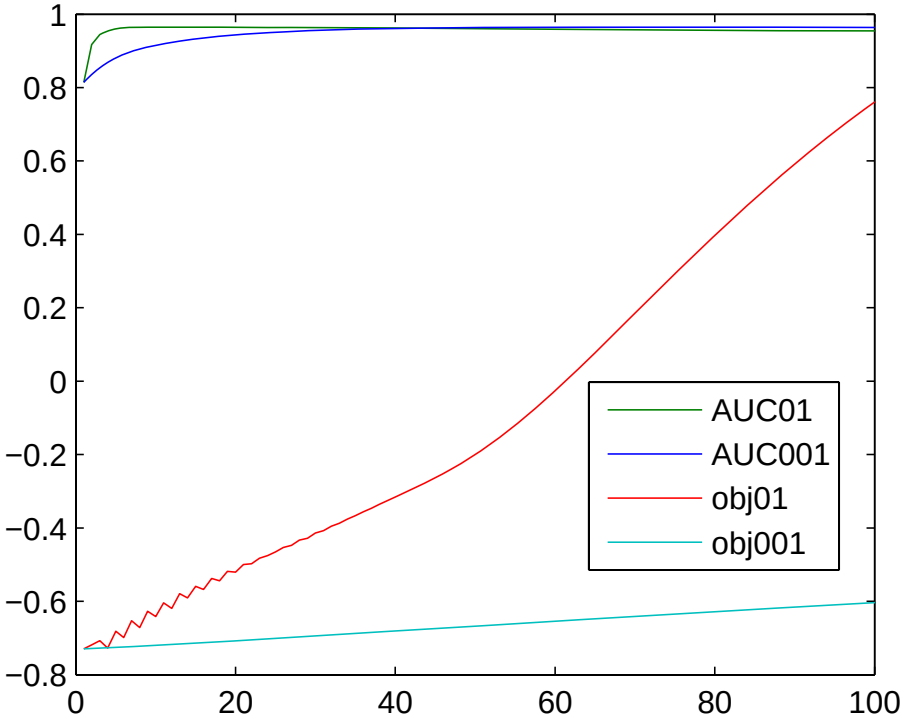
- RGP and XGP:
  - Learn only one GP.
  - $p(\mathbf{B}|\mathbf{Z})$ is itself a prediction function with $\mathbf{B}$ being a vector of function values for all input points.
  - The learned kernel, which is the covariance matrix of the posterior distribution $p(\mathbf{B}|\mathbf{Z})$, is $(\mathbf{K}^{-1} + \mathbf{\Pi}^{-1})^{-1}$ in RGP and $(\mathbf{K} + \mathbf{\Pi})$ in XGP, where $\mathbf{\Pi}$ is a kernel matrix capturing the link information.
- LWP:
  - Learn multiple (q) GPs.
  - Treat $\mathbf{A} = \mathbf{B}\mathbf{B}'$ as the learned kernel matrix.

Data Sets

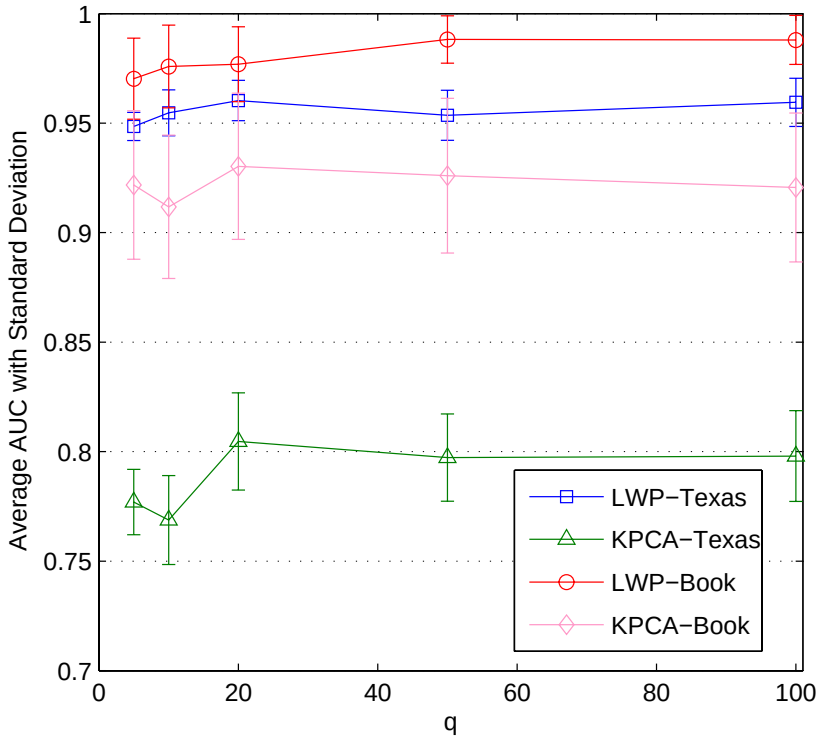
- WebKB:
  - Web pages from the CS departments of 4 universities: Cornell, Texas, Washington, Wisconsin
  - 4160 pages, 66249 links
  - 2-class problem: a page is either for {student, professor, course, project, staff, department} or for "others".
- Cora:
  - 4285 machine learning papers with their bibliographic citations
  - Each paper is labeled as one of 7 subareas of machine learning.
- Political Books:
  - 105 books, 43 of which are labeled as liberal ones
  - Pairs of books frequently bought together by the same customer are used to represent the relationship between them
  - 2-class problem: liberal or not.

Sensitivity to Parameters

Step size γ and iteration number T (X-axis denotes T, $\gamma = 0.01, 0.001$)

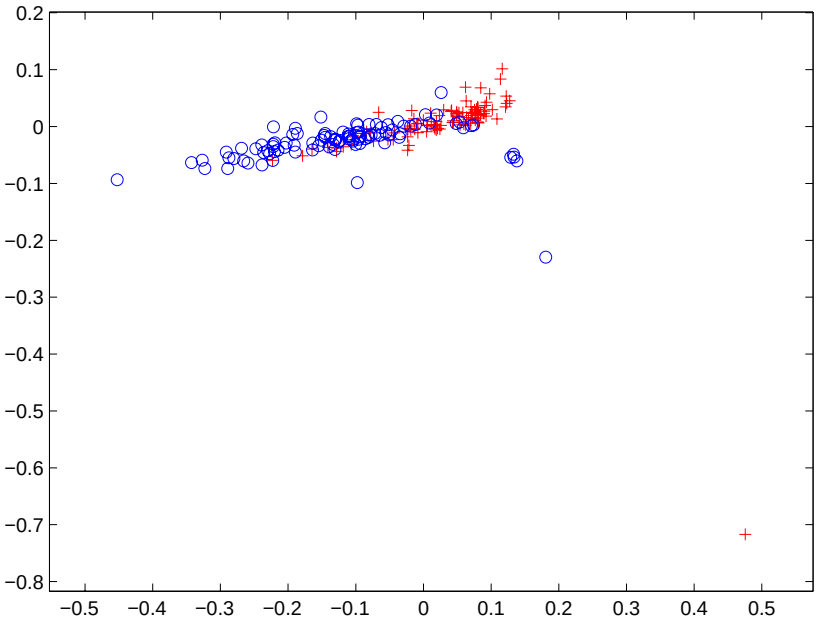


Dimensionality of latent space q (KPCA is used for initializing LWP)

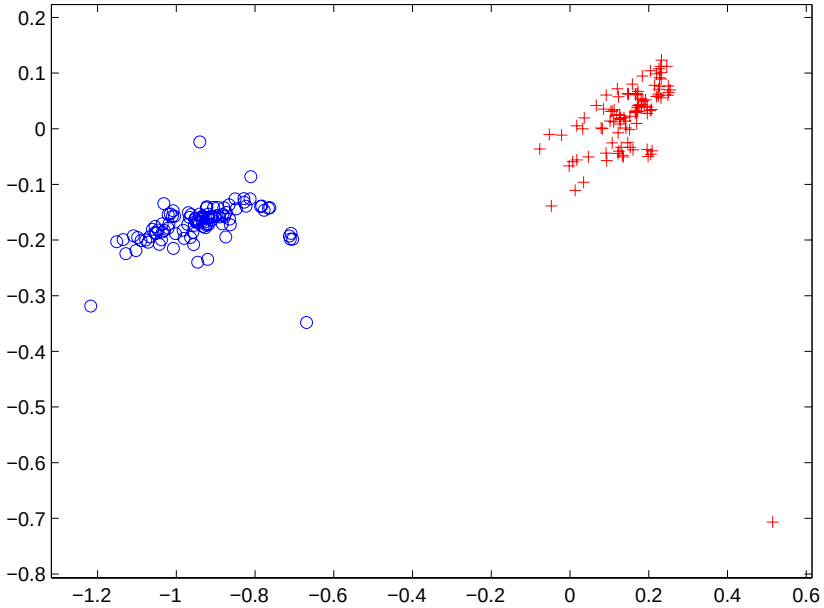


Visualization

KPCA



LWP



Performance on WebKB

Table: Mean and SD of AUC over 100 rounds of test on WebKB.

University	#Other/#All/#Links	GPC	RGP	XGP	LWP
Cornell	617 / 865 / 13177	0.708 ± 0.021	0.884 ± 0.025	0.917 ± 0.022	0.932 ± 0.019
Texas	571 / 827 / 16090	0.799 ± 0.021	0.906 ± 0.026	0.949 ± 0.015	0.960 ± 0.009
Washington	939 / 1205 / 15388	0.782 ± 0.023	0.877 ± 0.024	0.923 ± 0.016	0.935 ± 0.010
Wisconsin	942 / 1263 / 21594	0.839 ± 0.014	0.899 ± 0.015	0.941 ± 0.018	0.940 ± 0.012

- All methods are based on the same data partitions for both training and testing.
- #Other: number of positive examples
- #All: number of all examples
- #Links: number of links

Performance on Cora

Table: Mean and SD of AUC over 100 rounds of test on Cora.

Group	#Pos/#Neg/#Citations	GPC	GPC with Citation	XGP	LWP
5vs1	346 / 488 / 2466	0.905 ± 0.031	0.891 ± 0.022	0.945 ± 0.053	0.990 ± 0.000
5vs2	346 / 619 / 3417	0.900 ± 0.032	0.905 ± 0.044	0.933 ± 0.059	0.991 ± 0.001
5vs3	346 / 1376 / 3905	0.863 ± 0.040	0.893 ± 0.017	0.883 ± 0.013	0.986 ± 0.001
5vs4	346 / 646 / 2858	0.916 ± 0.030	0.887 ± 0.018	0.951 ± 0.042	0.997 ± 0.000
5vs6	346 / 281 / 1968	0.887 ± 0.054	0.843 ± 0.076	0.955 ± 0.041	0.998 ± 0.000
5vs7	346 / 529 / 2948	0.869 ± 0.045	0.867 ± 0.041	0.926 ± 0.076	0.992 ± 0.002

- All methods are based on the same data partitions for both training and testing.
- #Pos: number of positive examples
- #Neg: number of negative examples
- #Citations: number of links

Performance on Political Books

Table: Experiment on political books data set.

GPC	RGP	XGP	KPCA	LWP
0.92	0.98	0.98	0.93 ± 0.03	0.98 ± 0.02

Main Contributions

- LWP achieves state-of-the-art performance in diverse applications.
- LWP is the first model that employs WP for relational learning.
- LWP is naturally applicable for inductive inference over test data.
- LWP is unsupervised in nature.
  So it can be used for visualization or clustering of relational data.

Future Work

- Inductive inference experiments
- Social network analysis