Temporal-Relational Classifiers for Prediction in Evolving Domains

IEEE ICDM, Pisa, Italy

December 18th, 2008

Authors: Umang Sharan, Jennifer Neville

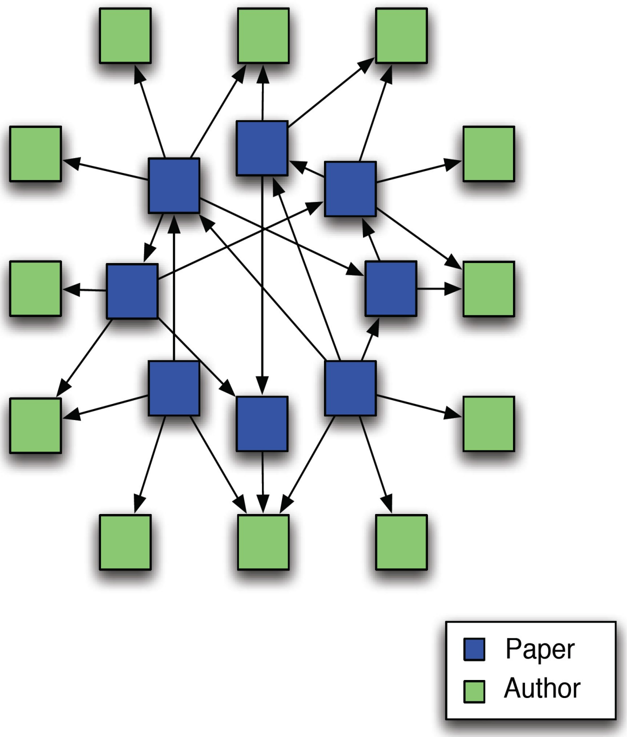
Network Learning and Discovery Lab

Purdue University, West Lafayette

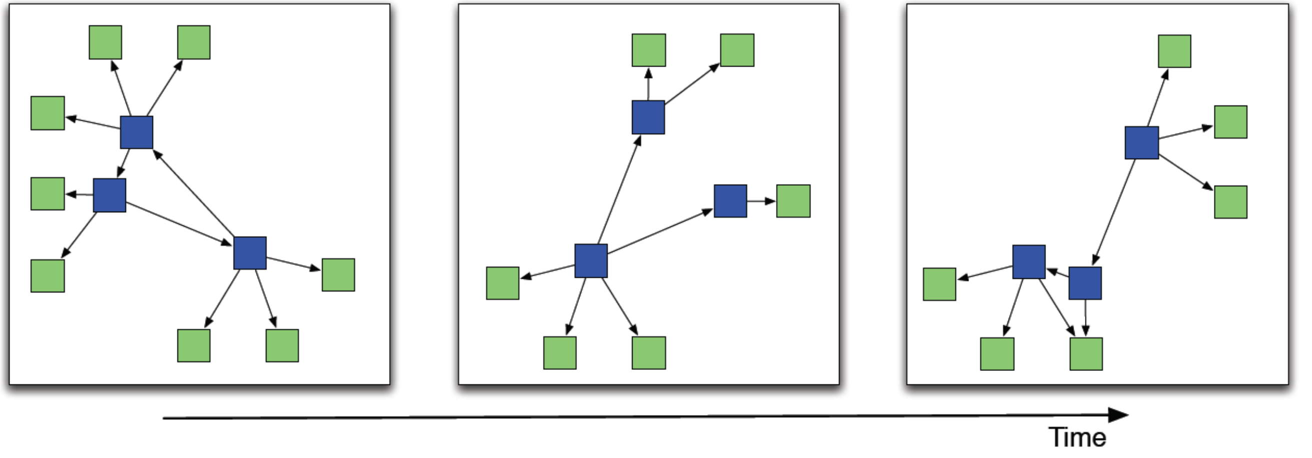
Motivation

Example Task:

- Relational learning has achieved significant performance gains by exploiting homophily/autocorrelation.
- Improved performance depends on level of autocorrelation in data
  - Stronger relationships are more likely to exhibit high levels of autocorrelation.
  - Spurious or fleeting relationships are more likely to exhibit low levels of autocorrelation.
- Current modeling techniques generally treat all links equally.

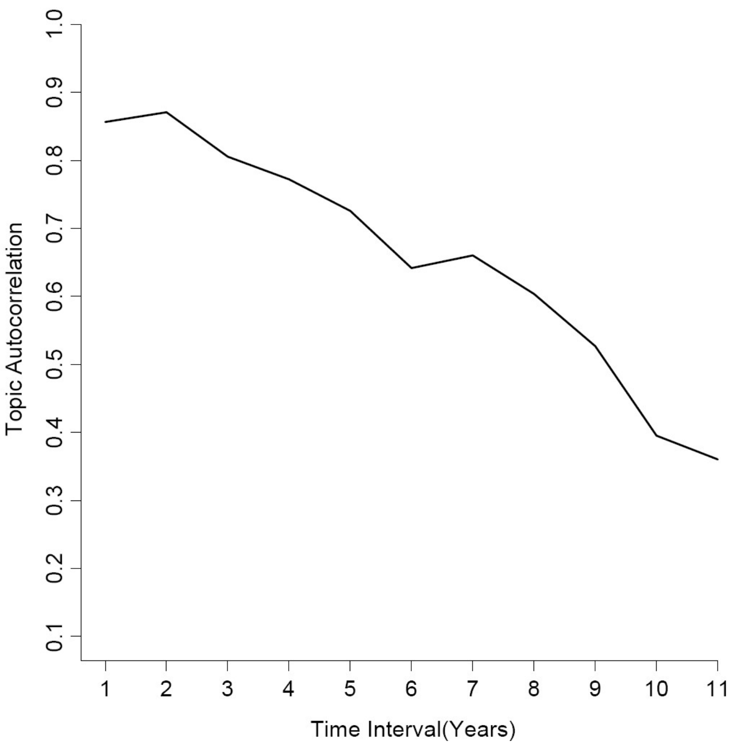


Temporal Relational Domains



- Temporal aspects contain information that is important to model.
    - Exploiting **temporal locality** and **temporal recurrence**.
- Many relational domains have temporal dynamics (e.g. Fraud Detection, Web Analysis, Bioinformatics, etc).

Temporal Aspects → Identify Stronger Relationships



Topic autocorrelation between papers published in 1996 with their references published in the past.

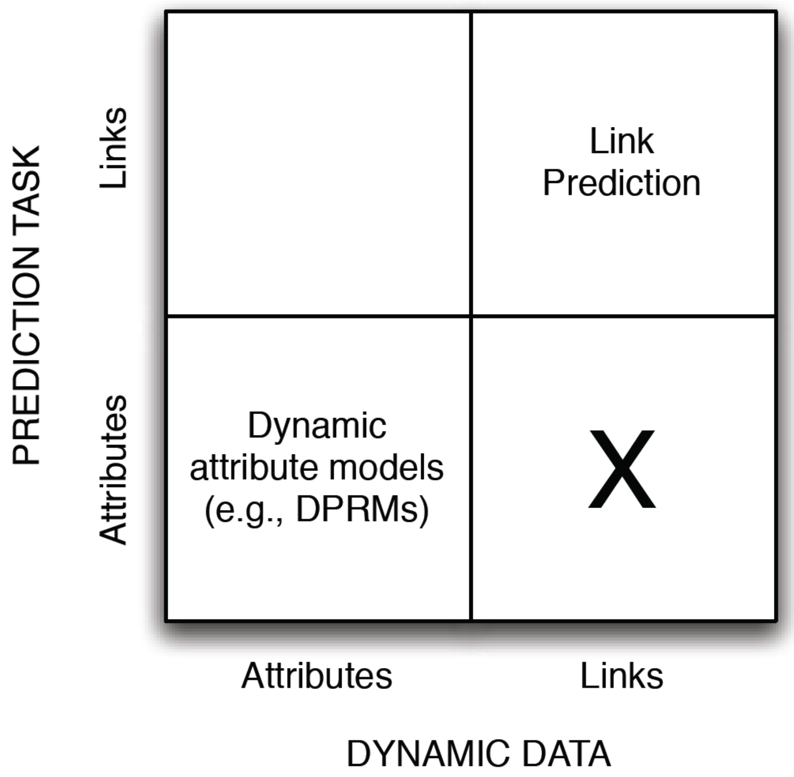
Movie	Release Year	Earnings ($ million)
Zoolander	2001	54
The Royal Tanenbaums	2001	62
Starsky and Hutch	2004	99
Meet The Fockers	2004	315
Night At The Museum	2006	264

List of movies that Owen Wilson and Ben Stiller have costarred in the past 7 years.

Both examples support Temporal locality and Temporal recurrence.

Related Work

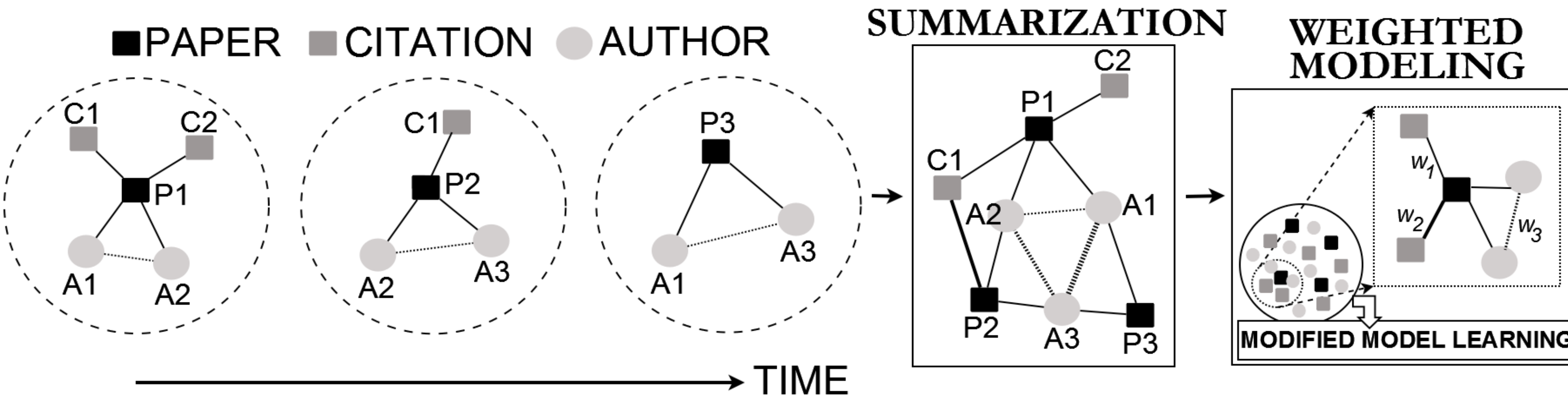
- Prediction models for relational domains – link structure and attributes varying with time.
- Spatio-Temporal RPTs (McGovern et al. '08): Adding Temporal features.
- Discrete Temporal Models (Hanneke et al. '06): Link prediction models based on first order Markov assumption.



Main Findings

- We describe a new framework for improving prediction models for temporally varying relational domains.
- TVRC or Time Varying Relational Classifier leads to improvements in prediction accuracy.
- TVRC better than baseline models on all real world datasets evaluated.

Time Varying Relational Classifier



- Two step approach:
  - Graph Summarization through Kernel Smoothing.
  - Weighted Model Learning for classification.

TVRC – Graph Summarization

- Represent the relationships at each time step *t* with graph G_t.
  - Temporal Events – A relationship occurring at a particular time step *t*. For e.g., Citing a particular paper.
  - Temporally Recurring - A relationship occurring periodically. For e.g., Co-authors publishing jointly.

- Use kernel smoothing to summarize (G_1, G_2, G_3, … , G_t) into G^S_t and estimate relationship strength.

Graph Summarization: Example

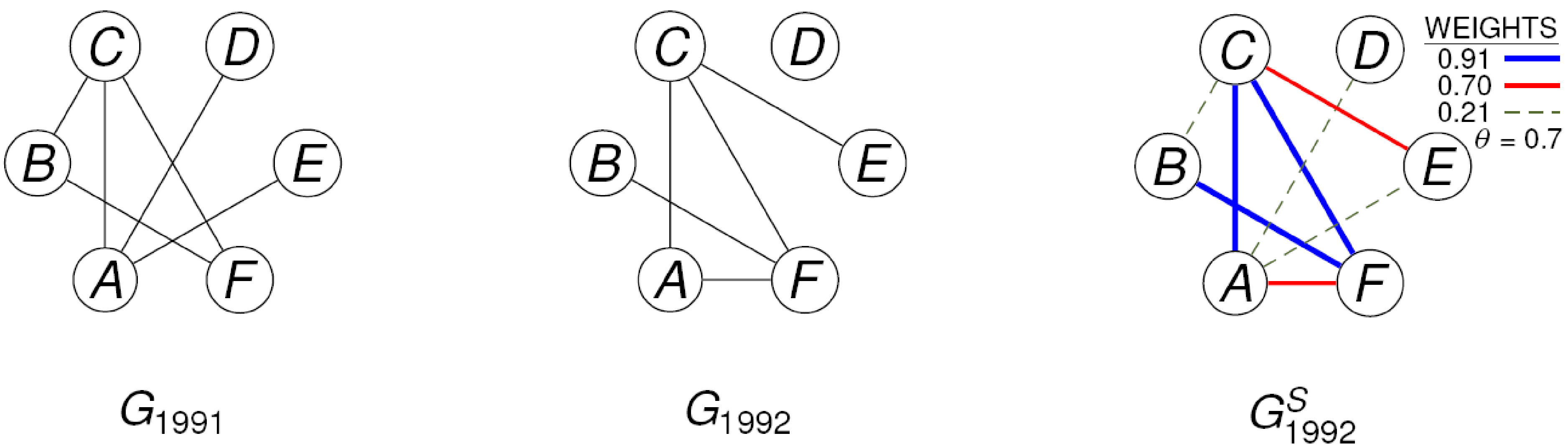
$$G_1=(V_1,E_1),\ldots,G_t=(V_t,E_t)$$

$$G_t^S=(V_t^S,E_t^S,W_t^S)$$

$$V_t^S=V_1\cup\ldots\cup V_t$$

$$E_t^S=E_1\cup\ldots\cup E_t$$

$$W_t^S=\alpha_1 W_1+\ldots+\alpha_t W_t=\sum_{i=1}^{t} K(G_i;t,\theta)$$



Kernel Smoothing

- <u>Exponential Kernel</u>

$$G_t^S = (1-\theta)G_{t-1}^S + \theta G_t$$

$$K_E(G_i;t,\theta) = (1-\theta)^{(t-i)}\theta W_i$$

- <u>Inverse Linear Kernel</u>

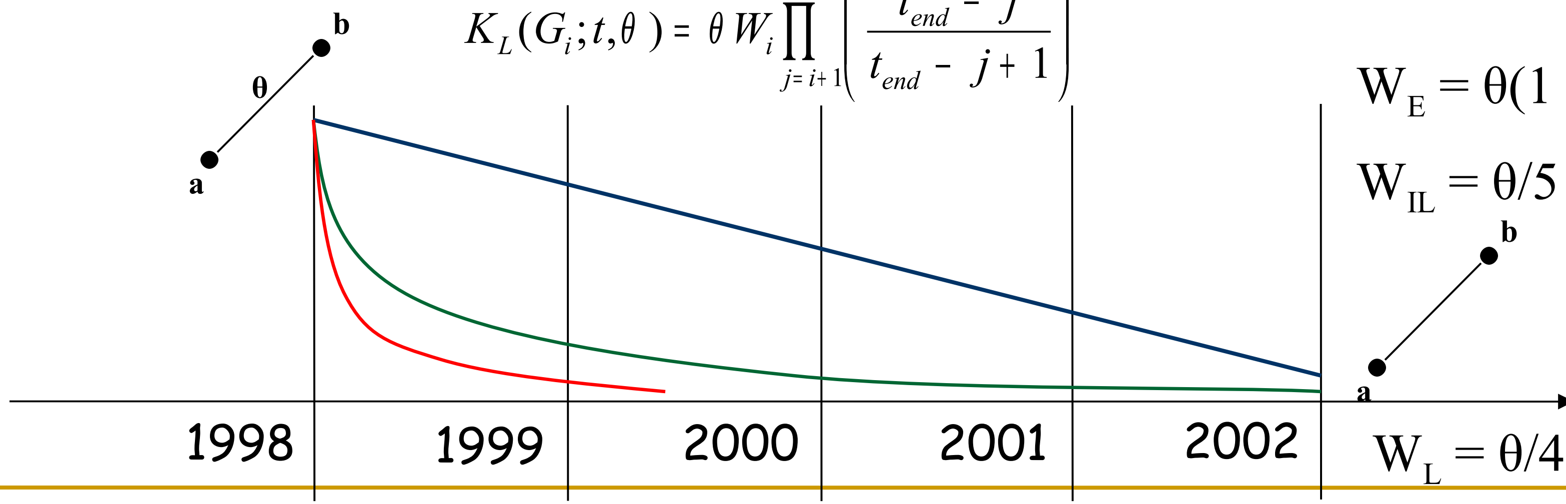
$$G_t^S = \left(\frac{t-t_{start}}{t-t_{start}+1}\right)G_{t-1}^S + \theta G_t$$

$$K_{IL}(G_i;t,\theta) = \theta W_i \prod_{j=i+1}^{t}\left(\frac{j-t_{start}}{j-t_{start}+1}\right)$$

- <u>Linear Kernel</u>

$$G_t^S = \left(\frac{t_{end}-t}{t_{end}-t+1}\right)G_{t-1}^S + \theta G_t$$

$$K_L(G_i;t,\theta) = \theta W_i \prod_{j=i+1}^{t}\left(\frac{t_{end}-j}{t_{end}-j+1}\right)$$



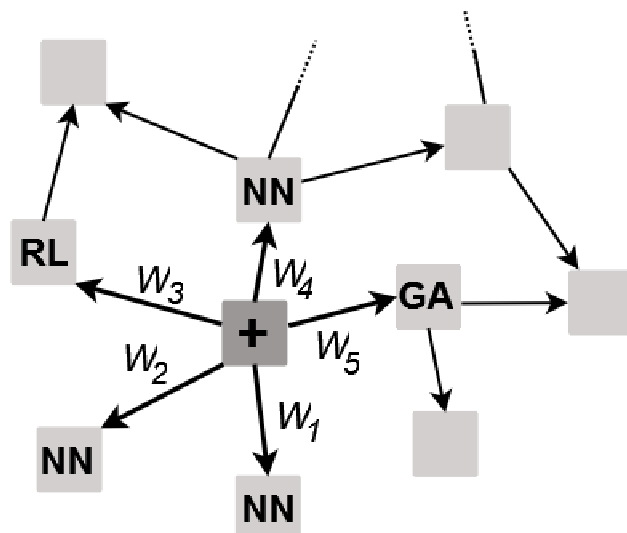
$W_E = \theta(1-\theta)^4$

$W_{IL} = \theta/5$

$W_L = \theta/4$

TVRC – Weighted Modeling

- Modify relational classifiers to incorporate link weights.
  - We consider two models – Relational Bayes Classifier (RBC) and Relational Probability Trees (RPT) (Neville et al. ’03).



RBC ⇨ {3:NN,1:RL,1:GA}

Weighted RBC ⇨ {(w_1+ w_2+ w_4):NN, w_3:RL, w_5:GA}

Weighted RBC

- More formally, for a class label **C**, attributes **X** and related items **R**, the Relational Bayes Classifier calculates the probability of **C** for an item **i** of type **G(i)** as:

$$P(C^i|\mathbf{X},R) \propto \prod_{X_m \in \mathbf{X}^{\mathbf{G(i)}}} P(X_m^i|C) \cdot \prod_{j \in R} \prod_{X_k \in \mathbf{X}^{\mathbf{G(j)}}} P(X_k^j|C) \cdot P(C)$$

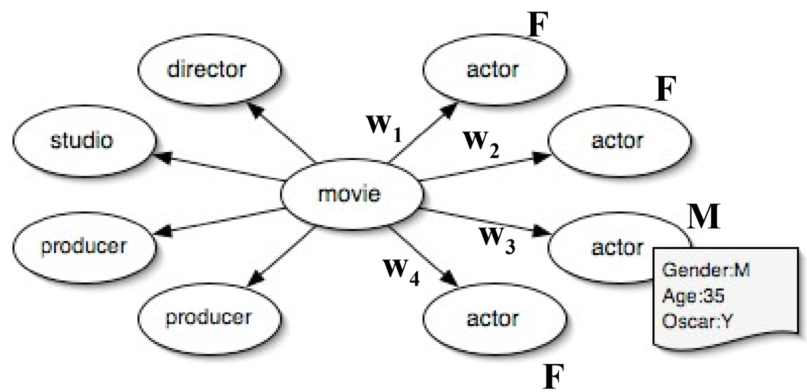
- We incorporate the weights from the summary graph (each edge (i,j) has weight w_{ij}) as follows:

$$P(C_t^i|\mathbf{X},R) \propto \prod_{X_m \in \mathbf{X}^{\mathbf{G(i)}}} P(X_m^i|C) \cdot \prod_{j \in R} \prod_{X_k \in \mathbf{X}^{\mathbf{G(j)}}} w_{ij}^t \cdot P(X_k^j|C) \cdot P(C)$$

Weighted RPT

- Aggregating relational neighborhood – Min/Max, Avg, Mode, Count, etc.
- For example,
  - Mode(Actor Gender) = F for RPTs
  - WMode(Actor Gender) = M for Weighted RPTs where w_1=0.1, w_2=0.2, w_3=0.8, w_4=0.3.

Receipt s>2m	Mode Actor Gender	Avg Actor Age	Mode Actor Oscar	Mode Studio Loc	…
+	F	28.50	N	USA	…
+	M	33.34	Y	USA	…
-	F	51.00	N	USA	…
+	M	22.79	N	Canada	…
…	…	…	…	…	…



Experimental Evaluation - Setup

Datasets Explored

IMDb	CORA	Reality Mining
Movies: 5,301 Actors: 126,641 Producers: 11,973 Studios: 391 Directors: 2,535 Editors: 2,186 Cinematgrs: 1,518	Papers:16,153 References: 29,603 Authors: 21,976	People: 97 Devices: 20,795 Telephone Call Edges: 443,553 Device Proximity Edges: 285,512
Time Window: 1981-2007	Time Window: 1981-1998	Time Window: May-Nov 2004

Kernels Explored

- Linear Kernel
- Inverse Linear Kernel
- Exponential Kernel

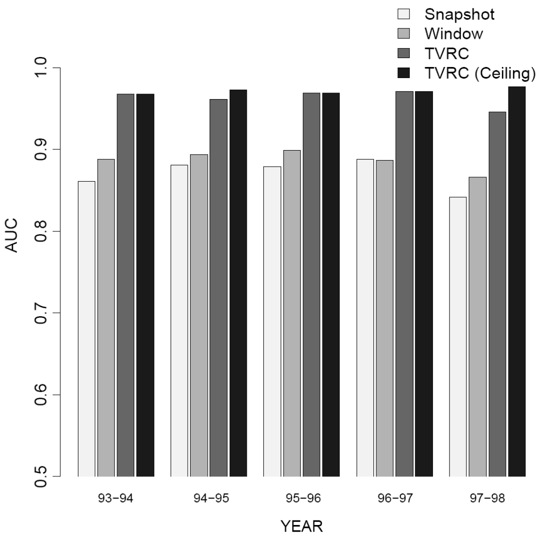
Models Explored

- Weighted RBC
- Weighted RPT

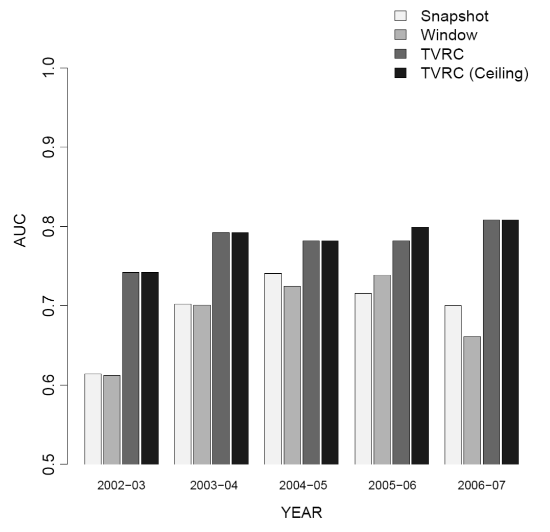
Models Compared

- **Snapshot Model:** Baseline model which uses all objects and links up to time ***t*** without weighing.
- **Window Model:** Baseline model which uses information from the previous time step.
- **TVRC:** Our model with graph summarization and weighted model learning. Parameter estimation through cross validation.
- **TVRC (Ceiling):** An 'optimal' TVRC model – included as a ceiling comparison for TVRC.

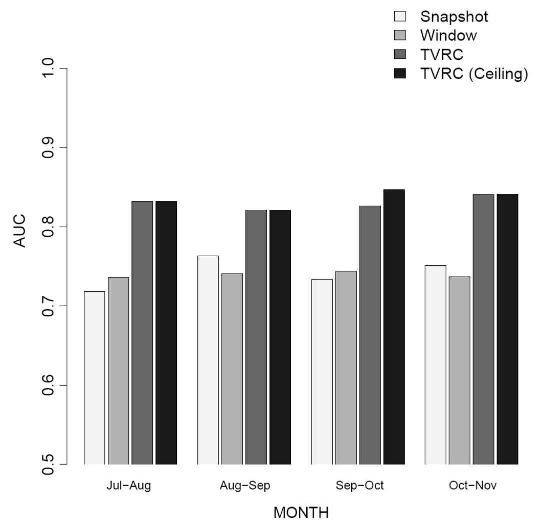
Exponential Kernel + Weighted RBC



CORA



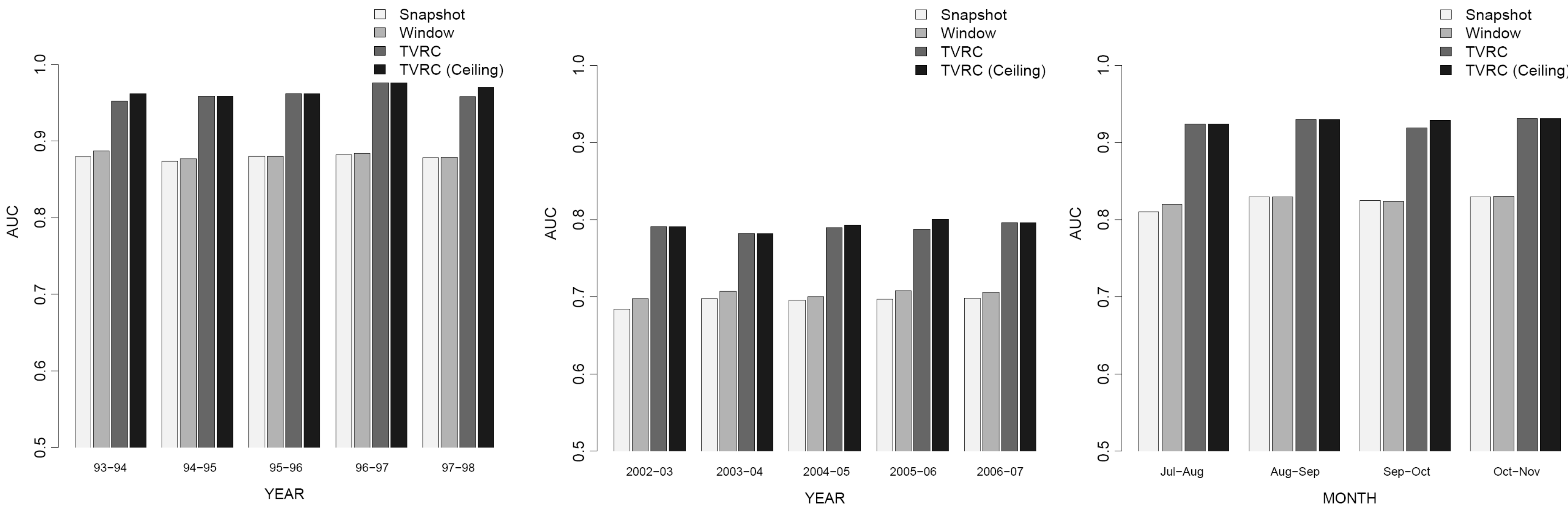
IMDb



Reality Mining

TVRC better than baselines: 10-12% on avg

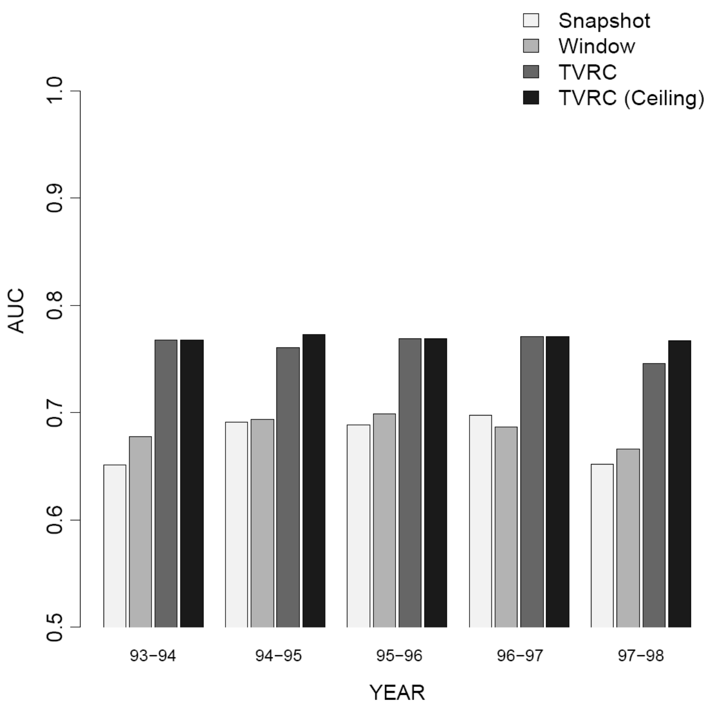
Exponential Kernel + Weighted RPT



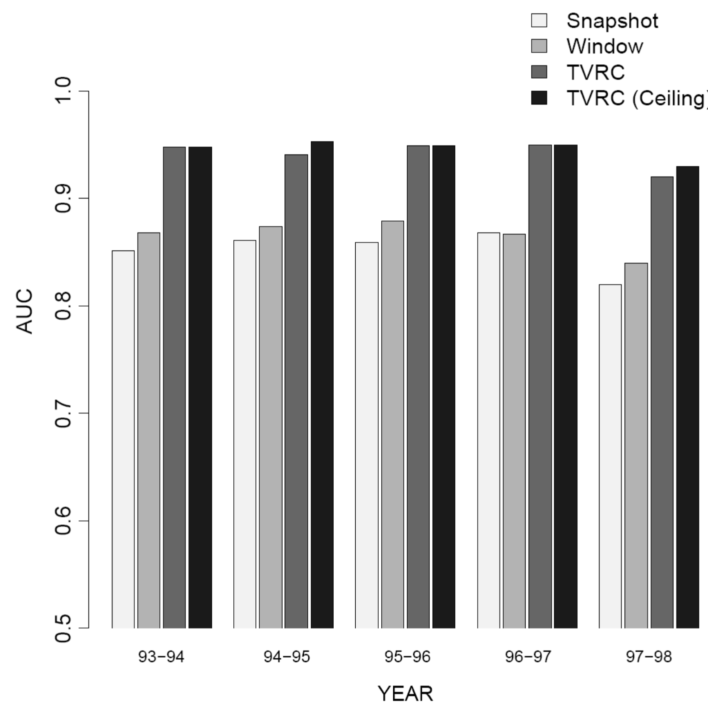
CORA IMDb Reality Mining

TVRC better than baselines: 10-15% on avg

Temporal Events Vs Temporally Recurring Relationships (CORA)



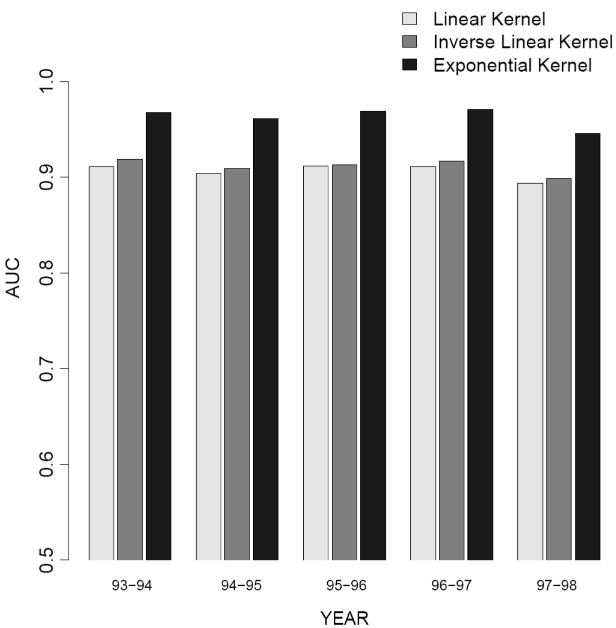
Recurring Relations (Co-authorship)



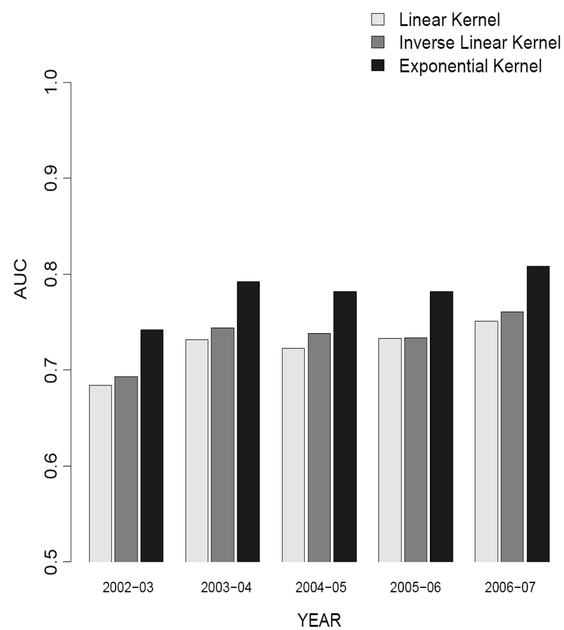
Events (Citations)

Exponential Kernel + Weighted RBC

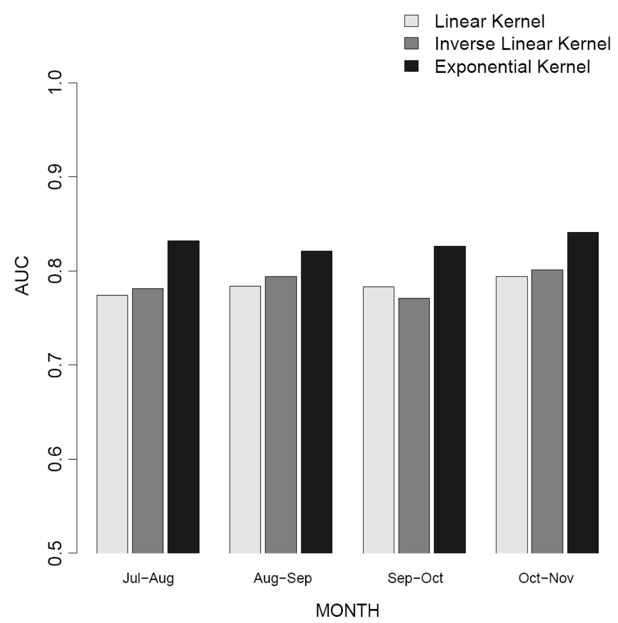
Comparing Kernel Performance



CORA



IMDb



Reality Mining

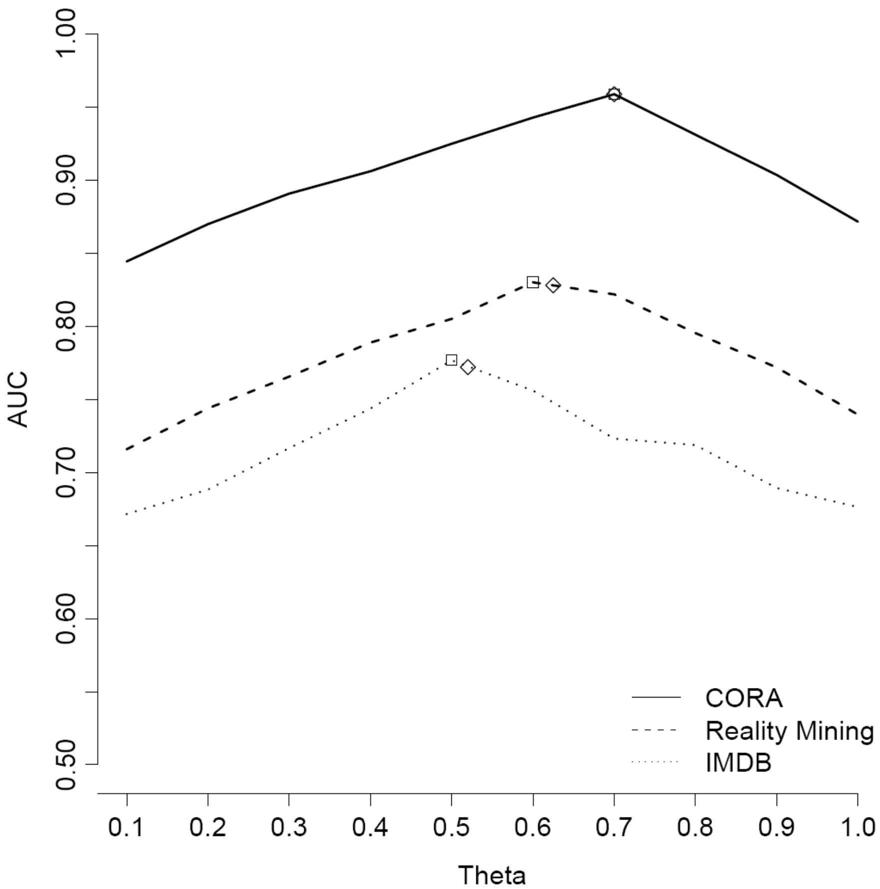
Weighted RBC: Exponential Kernel 5-8% better than other kernels

Cross Validation

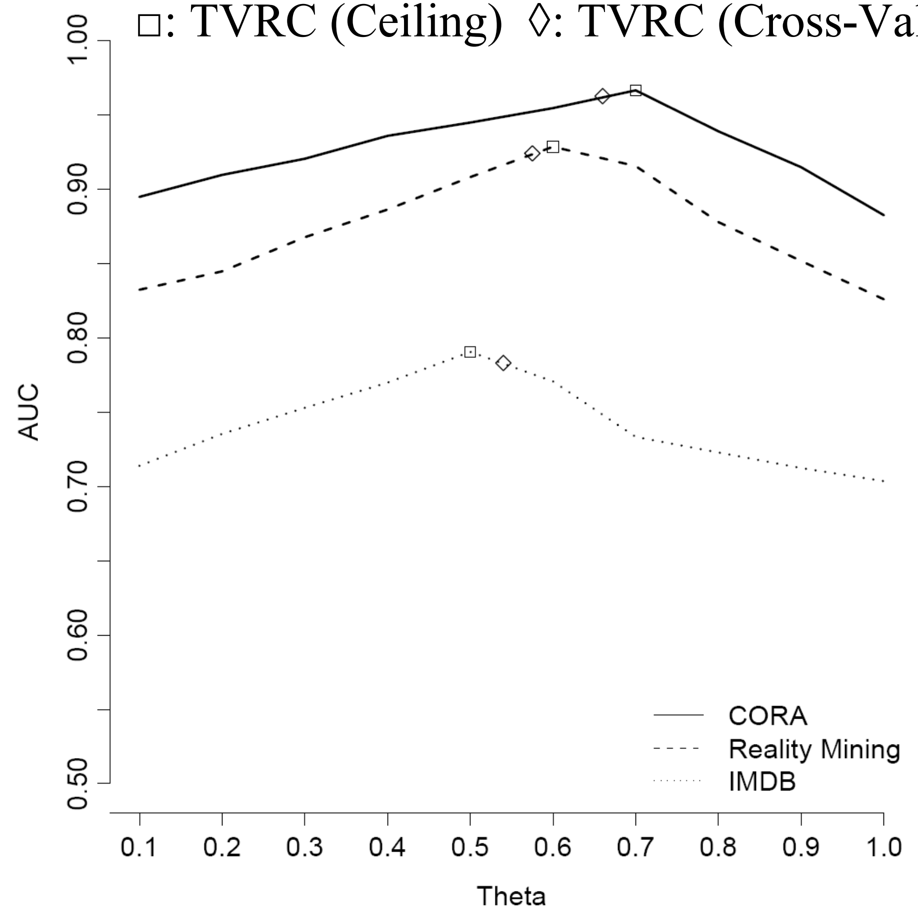
- TVRC does parameter estimation using *k-fold* cross validation.
- We do i.i.d cross validation that ignores the relational links.
- Biased estimates of error (Neville et al. ’01) do not effect parameter choice.
  - Optimal θ remains the same.

Cross-Validation Analysis

□: TVRC (Ceiling) ◊: TVRC (Cross-Val)



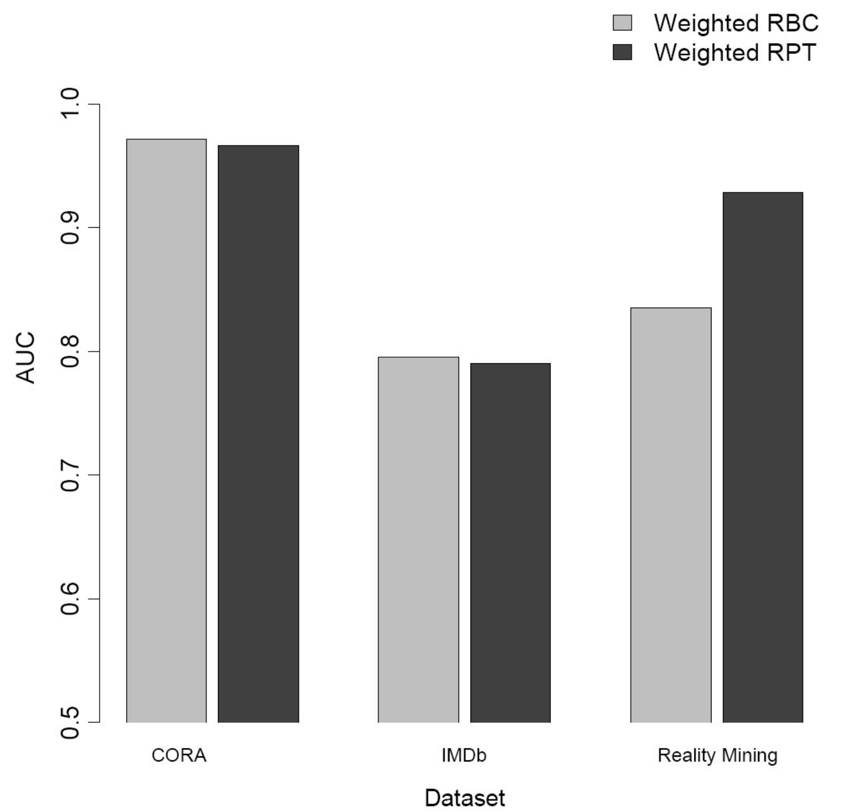
Weighted RBC



Weighted RPT

Weighted RBC vs Weighted RPT

- Weighted RBC and Weighted RPT performance comparison – exponential kernel summarization.
- Weighted RPT almost the same as weighted RBC on CORA and IMDb.
- Weighted RPT significantly better than Weighted RBC on Reality Mining – due to more prevalence of degree and count features.



Conclusions

- We presented a modular framework that exploits time-varying link structure.
- Evaluation on three real world datasets (CORA, IMDb and Reality Mining) using 3 kernels (Exponential, Linear and Inverse Linear) and 2 relational models (RBC, RPT).
- Incorporating Time improves prediction accuracy
  - TVRC significantly better than baseline models on all datasets regardless of kernel/model choice.

Conclusions…

- Exponential Kernel – the best choice amongst the kernels explored on all datasets.
- Automatic parameter estimation using cross validation not significantly different from TVRC (Ceiling) – based on paired t-test values.
- Weighted RPT as good as Weighted RBC on the datasets explored – better performance when the dataset has higher degree features (Reality Mining).

Future Directions

- Extend the approach to model temporally varying attributes.
  - Both relationships and attributes varying with time.
- Integrate information from multiple link types.
  - Different summary parameter for different types of links: Actor-Actor, Actor-Director, etc.
- Extending to a latent variable graphical model.
  - Summary weights are the latent variables learnt.

Questions?

- Contact Info:
  - Umang Sharan (usharan@cs.purdue.edu)
  - Jennifer Neville (neville@cs.purdue.edu)