Algorithms for Evolving Networks and Structured Predictions

Ricardo Silva

Department of Statistical Science, UCL and the Alan Turing Institute

Joint work with
Yin Cheng Ng

Setup

- Part I: Semi-supervised learning using graph-structured flexible predictors
- Part II: A dynamic edge-exchangeable model for sparse temporal networks

Part I:
Semi-supervised Learning in Networks

Data and Problem

- Suppose we have data on *citation networks*:
  - Each data point is a representation of the text containing in a document, plus a *label* indicating its category
  - On top of it, we have a network indicating which document cites which document
- The problem is to classify documents into this pre-defined set of classes assuming we know a subset of the labels (*training set*) and the network.
- This is an instance of a problem known in machine learning as *semi-supervised learning*.

Main Assumption

- “Guilty by association”: documents cite documents of the same category more often than an uniform distribution.

- Network information comes without cost: it is a by-product of an information system such as conference/journal management databases.
  - Measurement error is possible (documents may be misidentified), but we will ignore it as it is a problem for which much work already exists.

- Label information does have a cost: human labor necessary to provide a “seed” set of labels.

Outline

- Build upon a pre-existing method for supervised learning
  - Gaussian processes in this case
- Describe how the predictor of any given output should be “blurred” with the predictors of neighbors in the graph.
- Provide some intuition on why this modelling choice can capture real-world data.

A Quick Overview of Gaussian Processes

- The main idea is the definition of a black-box function mapping input vector **x** to output *y*.

$$y_n \mid f(\mathbf{x}_n) \sim p(y_n|f(\mathbf{x}_n)) \quad \forall n \in \{1, \ldots, N\}.$$

- One useful building block is the linear combination of a "dictionary" of transformations of the input data.

$$f(\mathbf{x}) = \phi(\mathbf{x})^{\mathsf{T}}\mathbf{w}$$

A Quick Overview of Gaussian Processes

- Interestingly, it is possible to choose a very large dictionary, even *infinite in size* (think of a Fourier basis, for instance), as long as we penalize overly complex ("wiggly") combinations of weights.

- One possibility is to put a Gaussian prior over these weights (usually) centered at zero and use the data to get a posterior using Bayes' rule.

A Quick Overview of Gaussian Processes

- It is a standard result of multivariate statistics this is the same as having a multivariate Gaussian given by some *covariance function k*.

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k_\theta(\mathbf{x}, \mathbf{x}')\big).$$

- This is very handy if available, as some infinite dimensional dictionaries then pose no problem in terms of computation (theoretically).

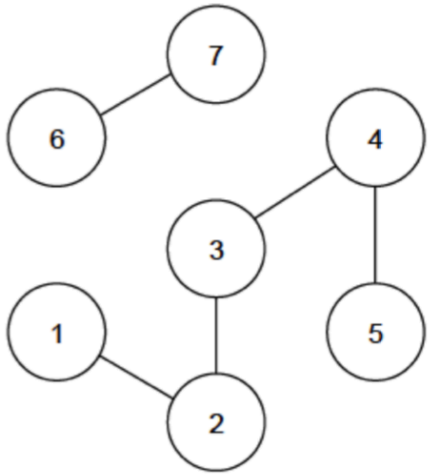
- It also gives a way of thinking about how to modify a model by modifying its covariance function.

The Graph Gaussian Process

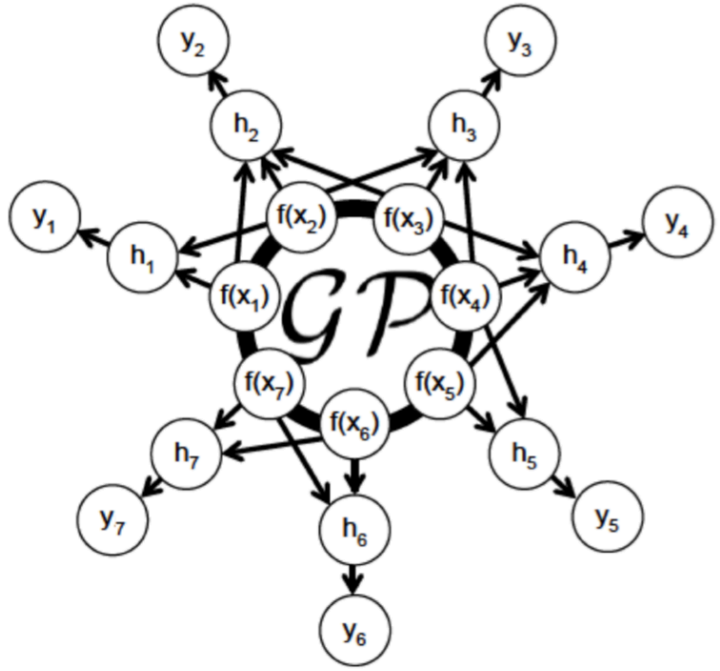
- Given adjacency matrix **A**, “blur” the function around each node, borrowing from its neighbors.

$$p_\theta(\mathbf{Y}, \mathbf{h}|\mathbf{X}, \mathbf{A}) = p_\theta(\mathbf{h}|\mathbf{X}, \mathbf{A}) \prod_{n=1}^{N} p(y_n|h_n),$$

$$h_n = \frac{f(\mathbf{x}_n) + \sum_{l \in Ne(n)} f(\mathbf{x}_l)}{1 + D_n}$$



Observed Graph



Implied Kernel

$$p_\theta(\mathbf{h}|\mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{0}, \mathbf{P}\mathbf{K}_{\mathbf{X}\mathbf{X}}\mathbf{P}^\mathsf{T})$$

where adjacency matrix **A** and diagonal degree matrix **D** provide

$$\mathbf{P} = (\mathbf{I} + \mathbf{D})^{-1}(\mathbf{I} + \mathbf{A})$$

Interpretation

- Covariance as similarity between empirical distributions

$$Cov(h_m, h_n) = \frac{1}{(1+D_m)(1+D_n)} \sum_{i \in \{m \cup Ne(m)\}} \sum_{j \in \{n \cup Ne(n)\}} k_\theta(\mathbf{x}_i, \mathbf{x}_j)$$

$$= \langle \frac{1}{1+D_m} \sum_{i \in \{m \cup Ne(m)\}} \phi(\mathbf{x}_i), \frac{1}{1+D_n} \sum_{j \in \{n \cup Ne(n)\}} \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$$

where we have an “empirical kernel embedding” of a neighborhood distribution

$$\hat{\mu}_n = \frac{1}{1+D_n} \sum_{j \in \{n \cup Ne(n)\}} \phi(\mathbf{x}_j)$$

Inference

- In nonparametric models, the predictive distribution is represented as an explicit function of the entire training data.
- A common approximation is to use “representative points” instead of the entire data, with some further independence assumptions.
  - In a Gaussian process, this boils down to cross-covariances between training points and the “representative” ones, plus covariance among the latter.
- We can show that such pseudopoints $\mathbf{z}_m$ can be defined so that graph-bas$Cov(h_n, f(\mathbf{z}_m)) = \frac{1}{D_n + 1}\left[k_\theta(\mathbf{x}_n, \mathbf{z}_m) + \sum_{l \in Ne(n)} k_\theta(\mathbf{x}_l, \mathbf{z}_m)\right]$ way:

Experiments

- Three datasets of citation networks (CORA, Citeseer, Pubmed)

	Type	N_{nodes}	N_{edges}	$N_{label\_cat.}$	$D_{features}$	Label Rate
Cora	Citation	2,708	5,429	7	1,433	0.052
Citeseer	Citation	3,327	4,732	6	3,703	0.036
Pubmed	Citation	19,717	44,338	3	500	0.003

- The input feature vector of each document is TFIDF processing of bag of words.

Classification Results with Small Number of Seeds

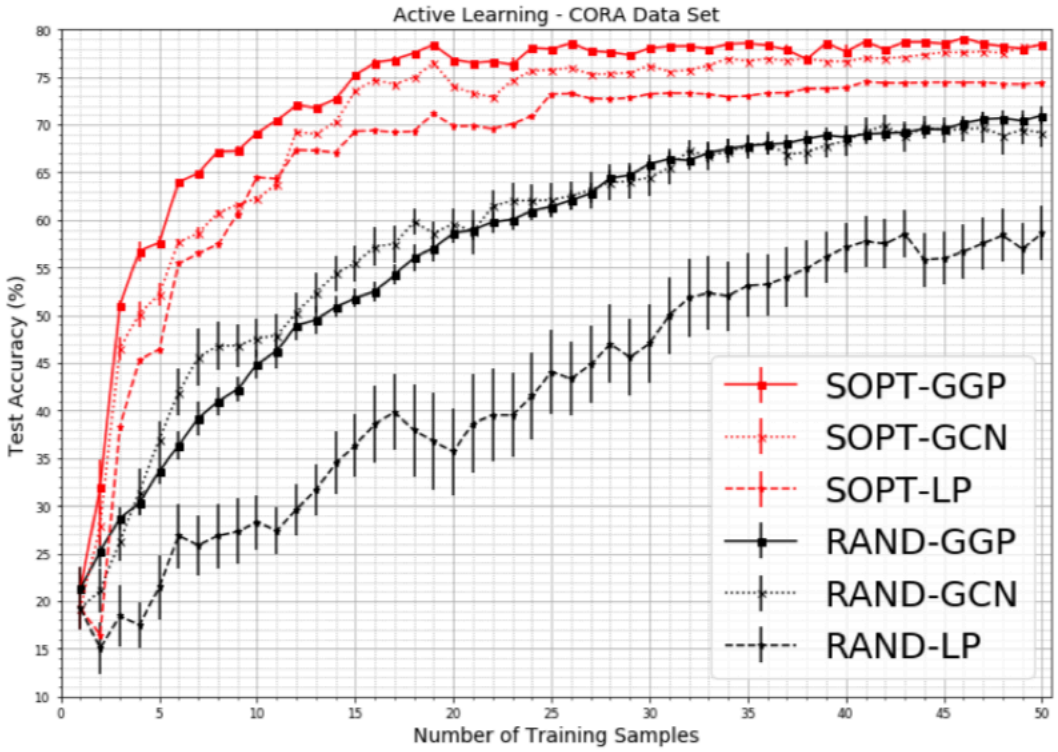
	Cora	**Citeseer**	**Pubmed**
GGP	80.9%	69.7%	77.1%
GGP-X	84.7%	75.6%	82.4%
GCN[25]	81.5%	70.3%	79.0%
DCNN[1]	76.8%	-	73.0%
MoNet[31]	81.7%	-	78.8%
DeepWalk[33]	67.2%	43.2%	65.3%
Planetoid[46]	75.7%	64.7%	77.2%
ICA[27]	75.1%	69.1%	73.9%
LP[48]	68.0%	45.3%	63.0%
SemiEmb[44]	59.0%	59.6%	71.1%
ManiReg[3]	59.5%	60.1%	70.7%

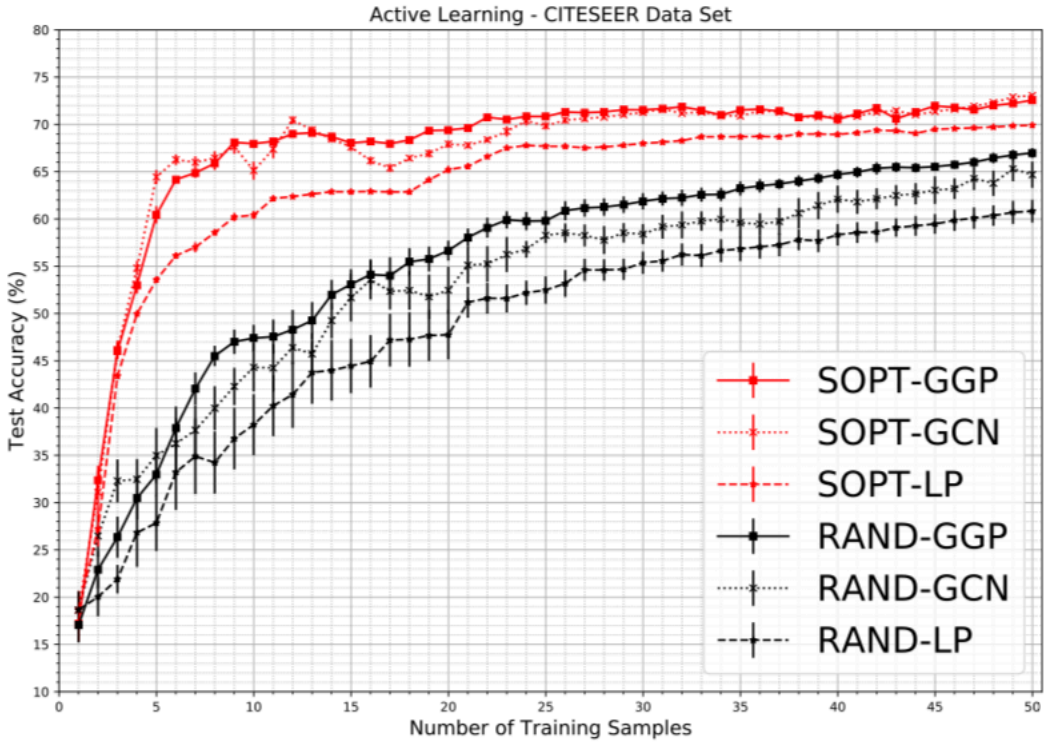
Active Learning Experiments

- Up to 50 points used, labelling one at a time while evaluating results on all remaining points.

	Cora	**Citeseer**
SOPT-GGP	0.733 ± 0.001	0.678 ± 0.002
SOPT-GCN	0.706 ± 0.001	0.675 ± 0.002
SOPT-LP	0.672 ± 0.001	0.638 ± 0.001
RAND-GGP	0.575 ± 0.007	0.557 ± 0.008
RAND-GCN	0.584 ± 0.011	0.533 ± 0.008
RAND-LP	0.424 ± 0.020	0.490 ± 0.011

Active Learning Experiments





Part II:
Temporal Network Evolution

Motivation

- Consider a network formed by participants and their communication patterns.
  - We will exemplify it using the ENRON dataset. Participants are employees, and a “link” is whether they exchanged emails at least once with a time period of 1 month.

- Over time, they might communicate differently due to change of ”topics” to be discussed.

- We would like to have models that can handle this data so that: i) it allows for *sparsity* in a particular technical sense; ii) it reveals *community structure* in a probabilistic sense and iii) it provides some evidence of *social influence* among participants.

Framework

- We will represent networks (and their temporal evolution) probabilistically.
- We will exploit the concept of *edge exchangeability* from probability theory to achieve sparsity.
- We will exploit *approximate inference* tools for the probabilistic AI literature to make community detection tractable.
- We will exploit *attention models* from the neural network literature for explain social influence and temporal changes

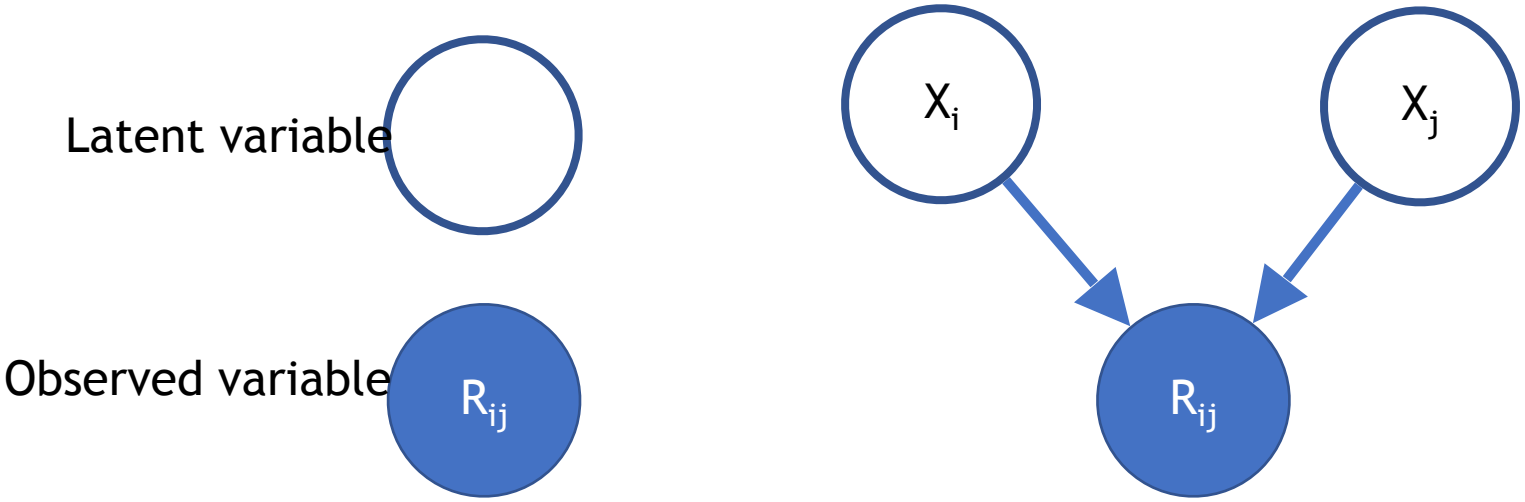
Sparsity in Real Networks

- Real networks are typically sparse, in the sense that vertices have “few” neighbors.
  - In a mathematical sense, we can say each vertex should have $o(n^2)$ neighbors (as opposed to $O(n^2)$), where n is the number of vertices, for some notion of “network growth”.

- At the same time, when we build probabilistic models, we like the concept of *exchangeability*.
  - Roughly speaking, the probability of observing a set of data points should not depend on the indexing of the data points.

Exchangeability in Networks

- Networks have multiple indices. We can index nodes and index edges. So we need to specify what we mean by exchangeability more clearly.
- A template for *node exchangeable* networks can be described by the following *generative model*:



$$X_i \sim F(\bullet)$$

Exchangeability in Networks

- Although popular, this type of model is not realistic. *We can't create sparse graphs with it*.

- The reason is that each entry in the respective adjacency matrix has, individually, O(1) chance appearing, so in expectation the number of edges is $O(n^2)$.

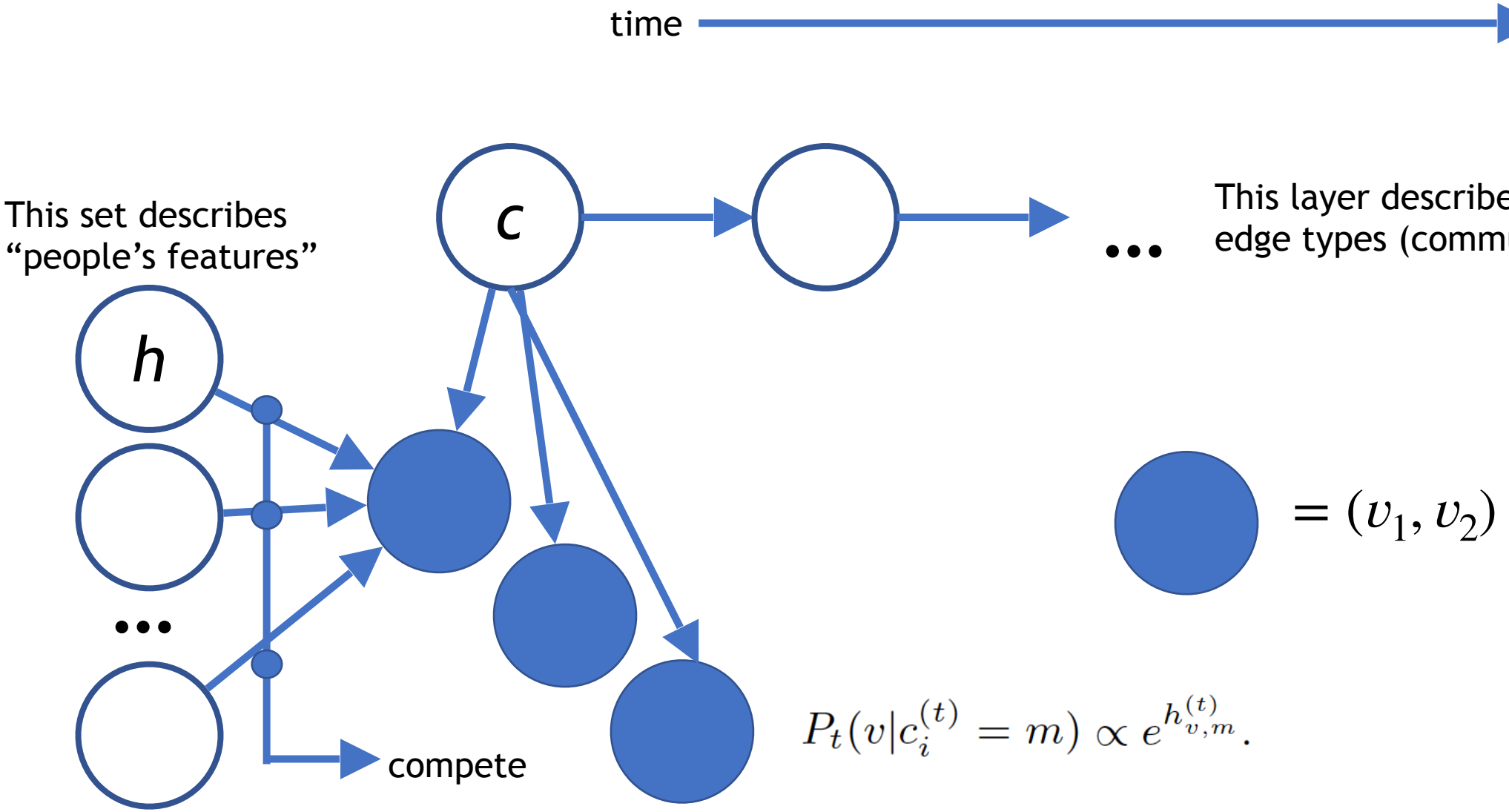
A Solution: *Edge-exchangeable* Models

- There is a rich theory about it. Here we adopt a plain model that will allows us to integrate it with temporal information and “influence” models.

- It boils down to something very straightforward: treat each edge as a tuple of two entries, with a discrete probability model of the two people linked.

- The number of people n_t at each time varies, so the model normalizes a distribution conditioned on n_t.

First Graphical Sketch



Community Structure

- We describe here the generative model that also include *latent community classes* that will create dependency among the latent features of each node.

$$P_t(e = (v_i^{(t)}, v_i'^{(t)})) = \sum_{m=1}^{M} [P_t(c_i^{(t)} = m) P_t(v_i^{(t)} | c_i^{(t)} = m) P_t(v_i'^{(t)} | c_i^{(t)} = m)]$$

Time-evolution of Communities

- A third layer of latent variables **k** provides a constructive definition of the model of latent communities.

$$P_t(c_i^{(t)} = m|\mathbf{k}^{(t)}) \propto e^{k_m^{(t)}}$$

$$p(\mathbf{k}^{(1:T)}) = \mathcal{N}(\mathbf{k}^{(1)}; \boldsymbol{\mu}_k, \mathbf{B}_k\mathbf{B}_k^\intercal) \prod_{t=2}^{T} \mathcal{N}(\mathbf{k}^{(t)}; \mathbf{A}_k\mathbf{k}^{(t-1)}, \mathbf{B}_k\mathbf{B}_k^\intercal)$$

Introducing Social Influence

- We incorporate a feedback mechanism, mapping the previous graph directly to the vertex latent space of the next time point.

$$p(\mathbf{h}_v^{(t+1)}|G^{(t)}, \{\mathbf{h}_i^{(t)}|i \in V^{(t)}\}) = \mathcal{N}(\mathbf{h}_v^{(t+1)}; \mathbf{f}(v, G^{(t)}, \{\mathbf{h}_i^{(t)}|i \in V^{(t)}\}), \mathbf{B}\mathbf{B}^\intercal)$$

$$\mathbf{f}_{v,t} = w_{vv}^{(t)}\mathbf{h}_v^{(t)} + \sum_{i \in ne(v,t)} w_{vi}^{(t)}\mathbf{h}_i^{(t)}$$

$$w_{vi}^{(t)} = \frac{e^{\mathbf{h}_v^{(t)} \cdot \mathbf{h}_i^{(t)}}}{e^{\mathbf{h}_v^{(t)} \cdot \mathbf{h}_v^{(t)}} + \sum_{j \in ne(v,t)} e^{\mathbf{h}_v^{(t)} \cdot \mathbf{h}_j^{(t)}}}$$

Introducing Social Influence

- The interpretation is similar to “attention mechanisms” in recurrent neural networks.

- As such, we call this model the Attention Augmented State-Space model (**ATTAS**).

Finally: Birth/Death Mechanism

- New vertices might join or drop the network at any moment in time.

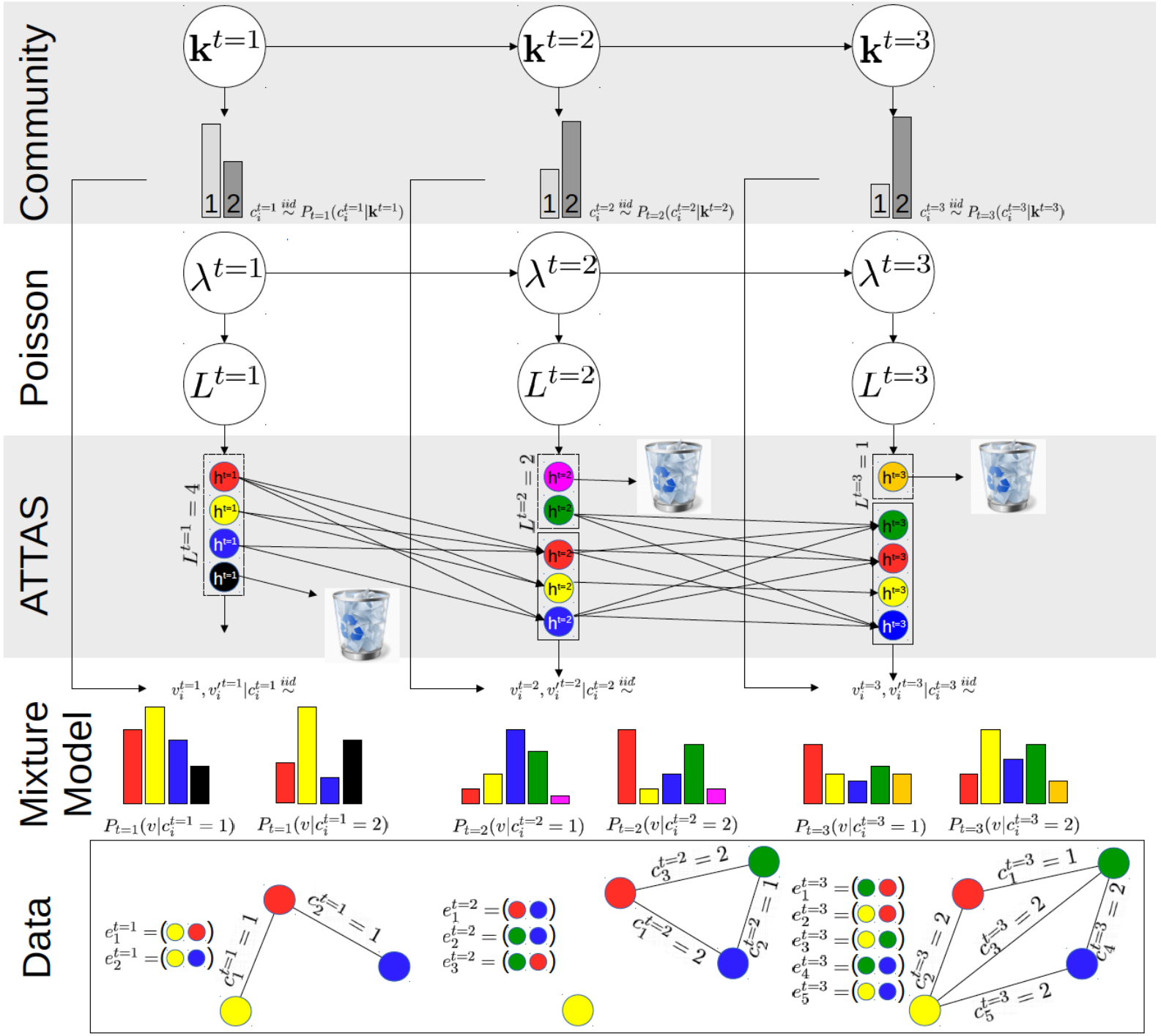
- The number of new vertices is generated by a standard Gaussian-Poisson process.

$$P(L^{(t)}|\lambda^{(t)}) = Poisson(e^{\lambda^{(t)}}).$$

$$p(\lambda^{(1:T)}) = \mathcal{N}(\lambda^{(1)}; \mu_\lambda; \sigma_\lambda^2) \prod_{t=2}^{T} \mathcal{N}(\lambda^{(t)}; a_\lambda \lambda^{(t-1)}, \sigma_\lambda^2).$$

- Existing vertices eventually “die” as a byproduct of not being linked to any existing vertex.

Final Diagram



Learning

- We infer all latent variables and posterior distributions of the models using *variational inference*, an approximation technique for probabilistic models with latent variables.

- Details are complicated (but standard).

- The main message is: unlike vertex-exchangeable models where we need to represent explicitly the “non-edges” (always $O(n^2)$), in edge-exchangeable method can be computed in time *linear on the number of edges*.

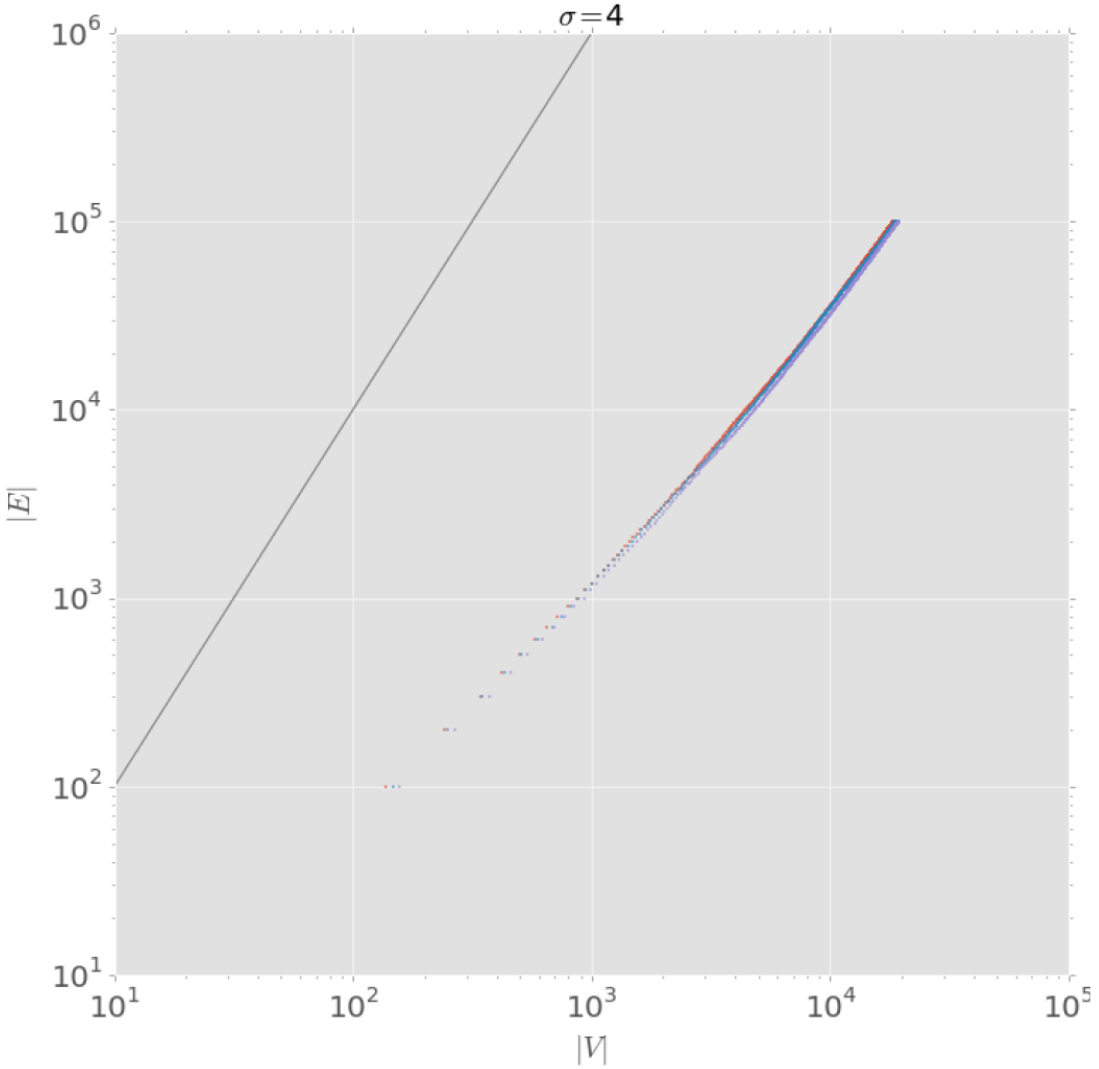
Experiments

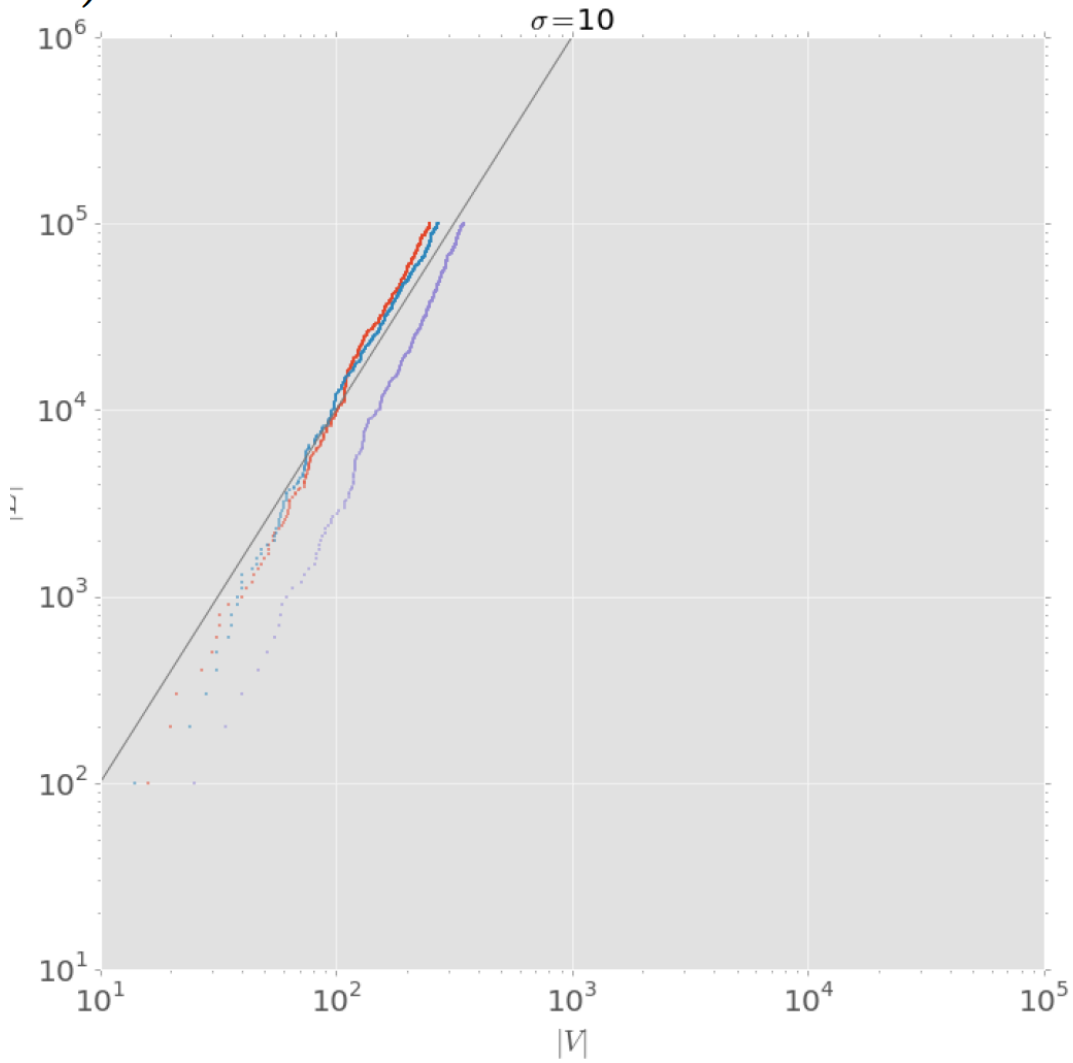
- We conducted 3 experiments with the following goals.

1. Investigate sparsity under various hyper-parameter settings.
2. Benchmark the model's link prediction powers.
3. Investigate the model's capacity to capture community structures.

Sparsity Assessment

$$h_i \sim \mathcal{N}(0, \sigma^2)$$





Link Prediction Assessment

- Setup: three to four time points. At each time point, we are given a partial view of it. Predict the rest.

- The problem can be relatively easy for some domains where link history is highly predictive of the majority of future links. Hence, we assess a variation of the prediction problem, where we classify whether a previously unseen pair will be linked at the new time point.

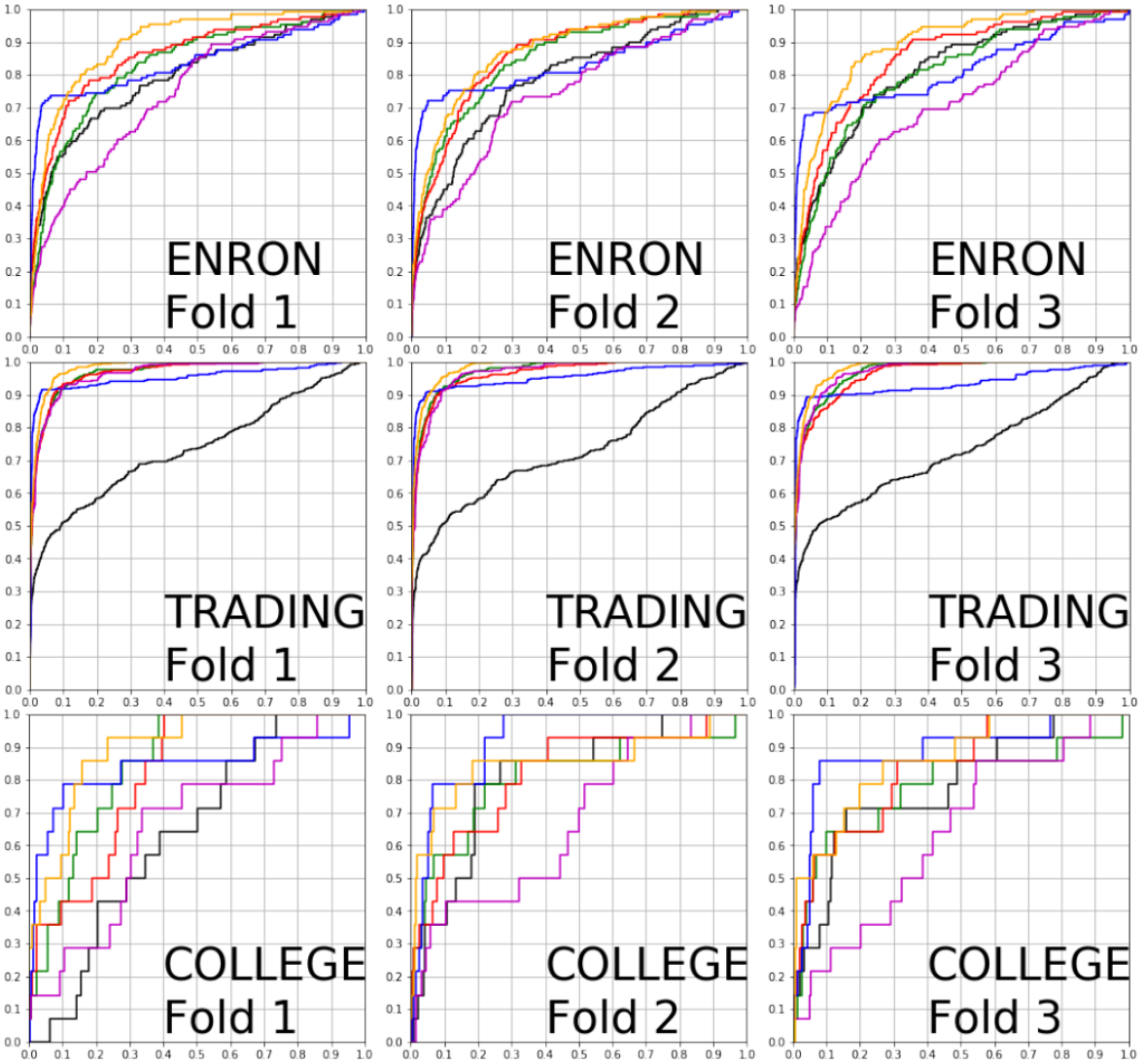
Datasets

- ENRON: email communications among employees of ENRON. 4 months, up to 138 vertices.
- TRADING: four years of international trade. Up to 134 vertices.
- COLLEGE: 7 snapshots of self-assessed friendship networks from a Dutch university. Up to 31 vertices.

Results

		ENRON	TRADING
F1	ATTAS	0.124±0.010	0.171±0.014
	LFP	0.102±0.014	0.160±0.012
AUC	ATTAS	0.806±0.012	0.881±0.011
	LFP	0.806±0.005	0.942±0.002

ATTAS
LFP
RW
DM



Community Detection Experiment

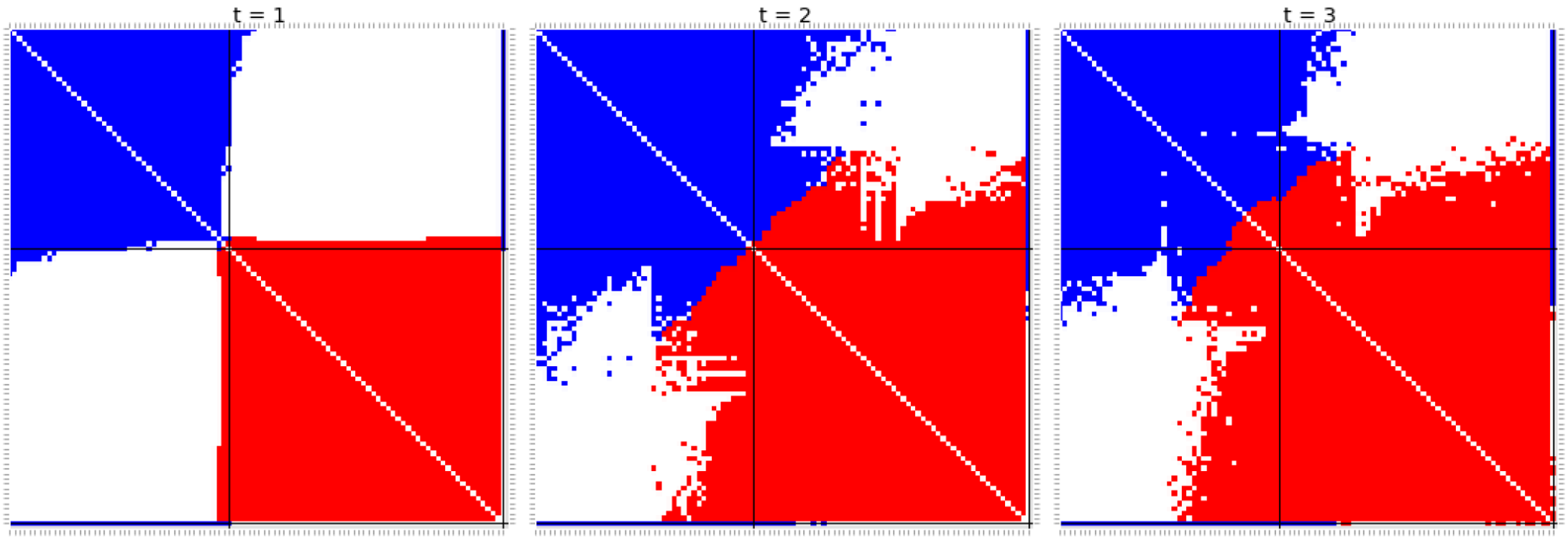


Fig. 4 Adjacency matrices of US Congress with the edges colored according to the inferred community types.

Conclusion

Conclusion

- Even very simple smoothing strategies based on network structure have value
  - Particularly attractive if we want to preserve tools that model uncertainty.
- “Social influence”, while hard to properly quantify in a causal way, is amenable to be exploited on modelling network evolution.
- Even the most bare-bones edge exchangeable models have clear advantages compared to the more common node exchangeable alternatives.
- Future work:
  - Scalability needs to be improved.
  - More exploitation of insights coming from graph convolutional neural networks.
  - A proper causal account of influence may be possible under the right assumptions.

References

- Yin Cheng Ng, Nicolò Colombo and Ricardo Silva. (2018). “Bayesian Semi-supervised Learning with Graph Gaussian Processes”. *Advances in Neural Information Processing Systems* 31 (NeurIPS 2018).

- Yin Cheng Ng and Ricardo Silva (2018). “A Dynamic Edge-exchangeable Model for Sparse Temporal Networks”. arXiv: 1710.04008.

Thank You

ricardo@stats.ucl.ac.uk