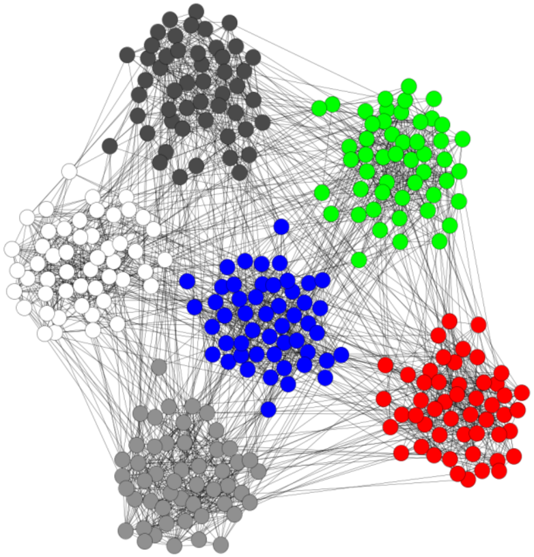
Community Discovery in Dynamic Networks via non-negative matrix factorization

Niloufar Afsariardchi
Supervisor: Mark Coates
McGill University
March 15,2011



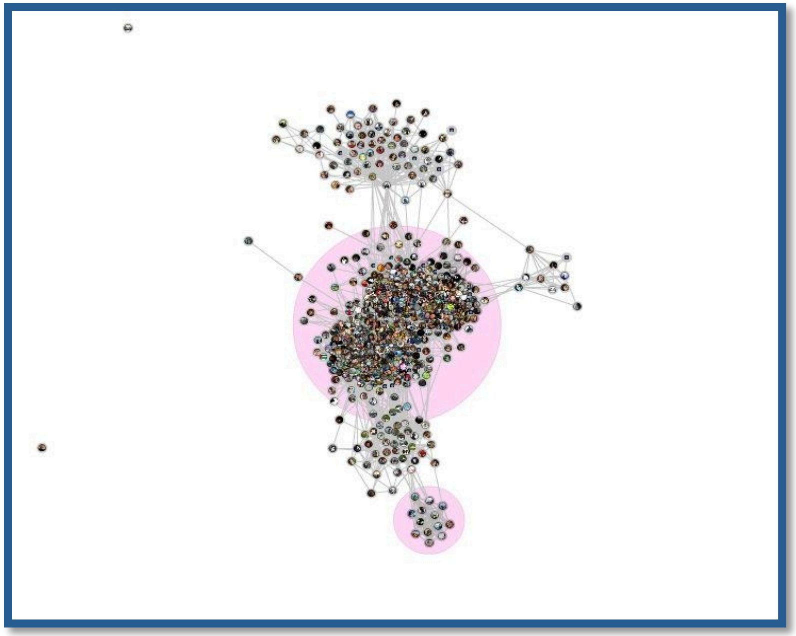
Outline

- Introduction
- Related Work
- Goals
- Non-negative matrix factorization
- Automatic relevance determination
- Dynamic model
- Experimental results
- Summary and future work

Communities Everywhere…

Why Dynamic Clustering?

- Recommendation Systems
- Detecting anomalies
- Studying dynamic features of social and biological networks



Social Networks: Facebook

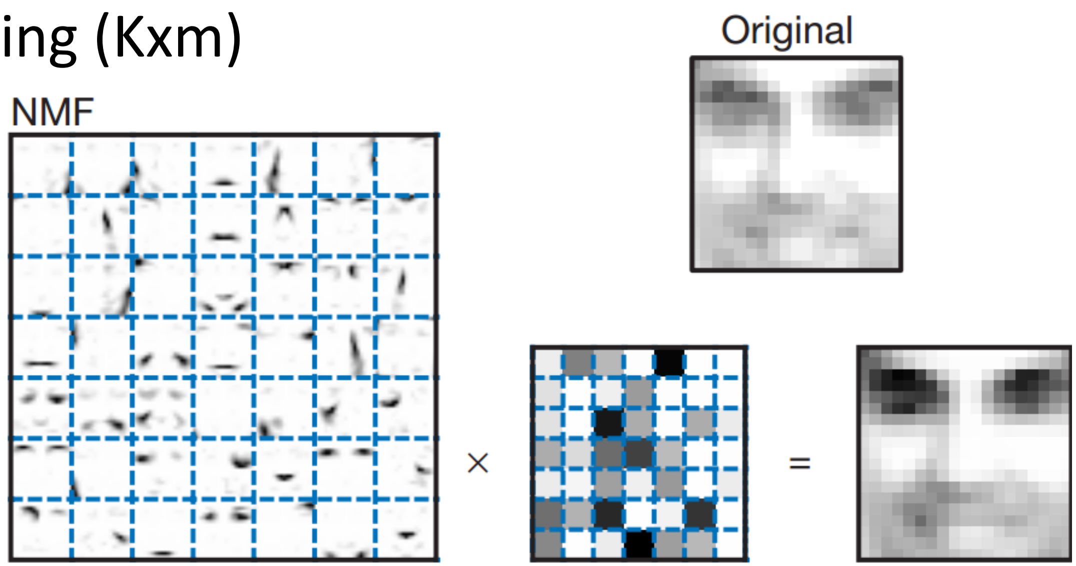
Goals

- Soft clustering
- Addressing the problem of unknown number of clusters over time
- Handling changing number of clusters over time
- Temporal smoothness

Learning the parts of objects by non-negative matrix factorization [Lee1999]

$$V_{ij} \approx (WH)_{ij} = \sum_{k=1}^{K} W_{ik} H_{kj}$$

- V: image database (nxm, m facial image each has n pixel)
- W: Basis images (nxK)
- H: Encoding (Kxm)



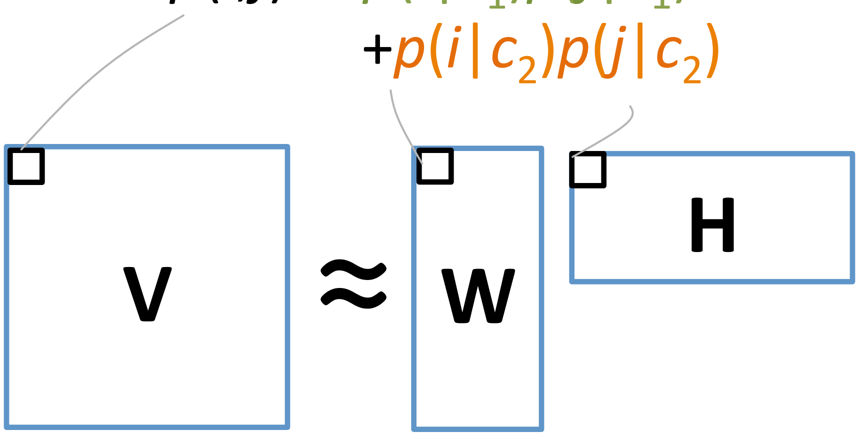
* Illustration from Daniel D. Lee and H. Sebastian Seung (2001). "Algorithms for Non-negative Matrix Factorization". Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press. pp. 556–562

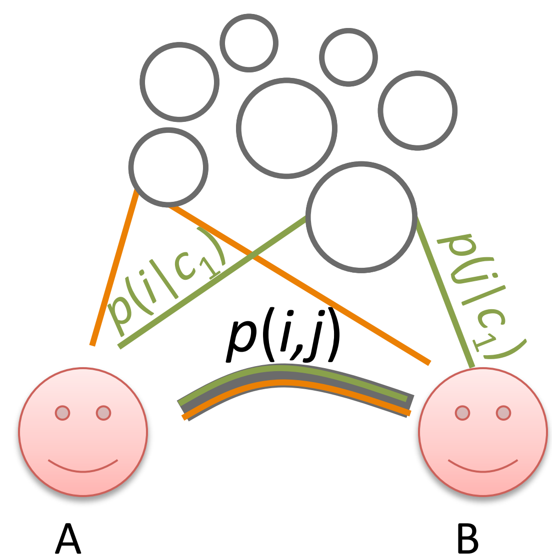
NMF for Clustering in Network

$$V_{ij} \approx (WH)_{ij} = \sum_{k=1}^{K} W_{ik} H_{kj}$$

- V : Matrix of observation
- w_{ik} : Probability of participation of node i in k-th community
- h_{ki} : Probability of individual i to be attracted by k-th community

$$p(i,j) \approx p(i|c_1)p(j|c_1) + p(i|c_2)p(j|c_2)$$





NMF inference

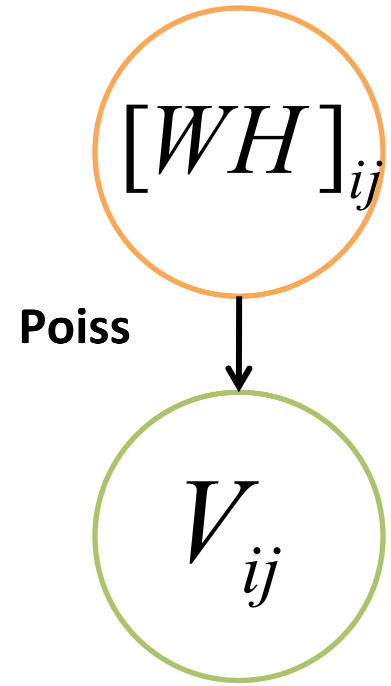
- MAP estimation
- Maximizing likelihood function with Poisson distribution = minimizing KL-divergence

$$W^*, H^* = \max p(W, H \mid V)$$

⇩ coordinate decent

$$H \leftarrow \left(\frac{H}{W^T}\right) \cdot \left[W^T \left(\frac{V}{WH} \right) \right]$$

$$W \leftarrow \left(\frac{W}{H^T}\right) \cdot \left[\left(\frac{V}{WH} \right) H^T \right]$$

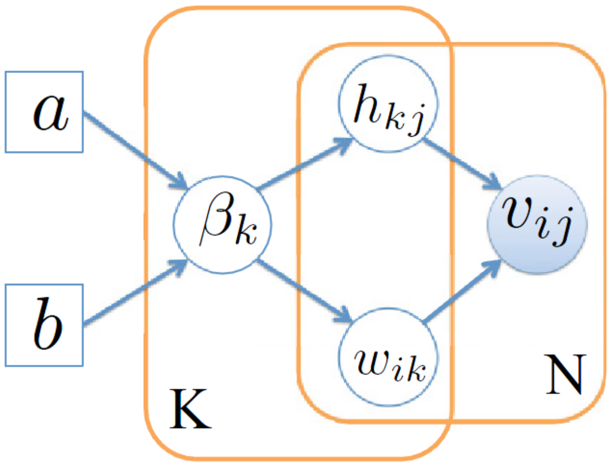


Goals

- ✓ **Soft clustering**
- Unknown number of clusters over time
- Changing number of clusters over time
- Temporal smoothness

Automatic relevance determination[Tan2009]

- Consider β_k s as prior of each community and a, b as hyperparameters
- Bayesian NMF Model



$$p(\beta_k|a_k, b_k) = \frac{b_k^{a_k}}{\Gamma(a_k)} \beta_k^{a_k - 1} \exp(-\beta_k b_k)$$

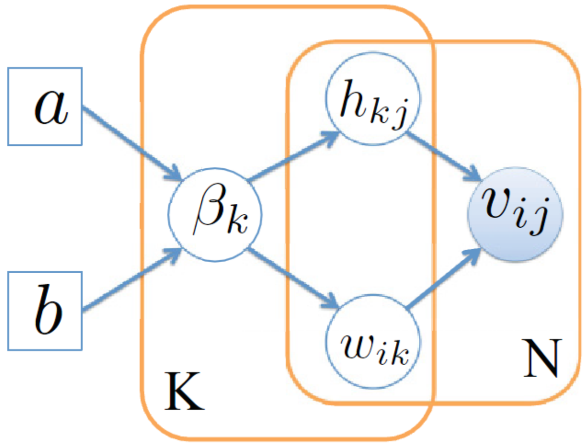
$$p(w_{fk}|\beta_k) = \mathcal{HN}(w_{fk}|0, \beta_k^{-1})$$
$$p(h_{kn}|\beta_k) = \mathcal{HN}(h_{kn}|0, \beta_k^{-1}),$$

- Log priors can be written as:

$$-\log p(\mathbf{W}|\boldsymbol{\beta}) \stackrel{c}{=} \sum_k \sum_f \frac{1}{2}\beta_k w_{fk}^2 - \frac{F}{2}\log \beta_k,$$

$$-\log p(\mathbf{H}|\boldsymbol{\beta}) \stackrel{c}{=} \sum \sum \frac{1}{2}\beta_k h_{kn}^2 - \frac{N}{2}\log \beta_k.$$

Automatic relevance determination[Tan2009]



- We can rewrite the likelihood function,

$$-\log p(\mathbf{W},\mathbf{H},\boldsymbol{\beta}|\mathbf{V}) \stackrel{c}{=} -\log p(\mathbf{V}|\mathbf{W},\mathbf{H}) - \log p(\mathbf{W}|\boldsymbol{\beta}) - \log p(\mathbf{H}|\boldsymbol{\beta}) - \log p(\boldsymbol{\beta})$$

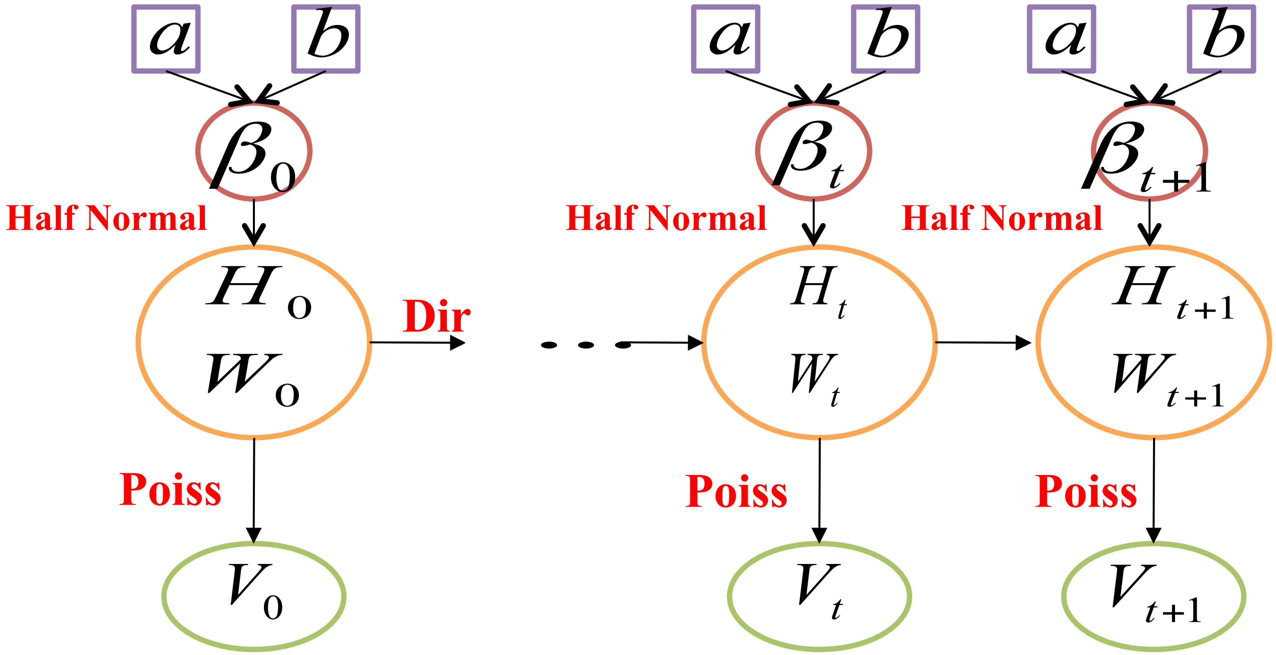
- The objective function we want to minimize,

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\beta}} C_{\mathrm{MAP}}(\mathbf{W},\mathbf{H},\boldsymbol{\beta}) \stackrel{\Delta}{=} -\log p(\mathbf{W},\mathbf{H},\boldsymbol{\beta}|\mathbf{V})$$

Goals

- ✓ **Soft clustering**
- ✓ **Unknown number of clusters over time**
- ✓ **Changing number of clusters over time**
- Temporal smoothness

Dynamic Model



$$\log p(\mathbf{W}_t|\mathbf{W}_{t-1}) = \log \frac{1}{B(\psi_t)} \prod w_{ik,t}^{\mu w_{ik,t-1}}$$

$$= \sum_{ik} \mu w_{ik,t-1} \log w_{ik,t} + c$$

$$\log p(\mathbf{H}_t|\mathbf{H}_{t-1}) = \log \frac{1}{B(\psi'_t)} \prod h_{ki,t}^{\mu h_{ki,t-1}}$$

$$= \sum_{ki} \mu h_{ki,t-1} \log h_{ki,t} + c$$

Parameter inference

- Parameter inference using multiplicative update rules
- Point estimation to find the maximum of likelihood function

$$\log \mathrm{L}(\mathrm{W}_t, \mathrm{H}_t) = \log \mathrm{P}(\mathrm{W}_t, \mathrm{H}_t, \beta_t \mid \mathrm{V}_t)$$
$$+ \log \mathrm{P}(\mathrm{W}_t \mid \mathrm{W}_{t-1}) + \log \mathrm{P}(\mathrm{W}_t \mid \mathrm{W}_{t-1})$$

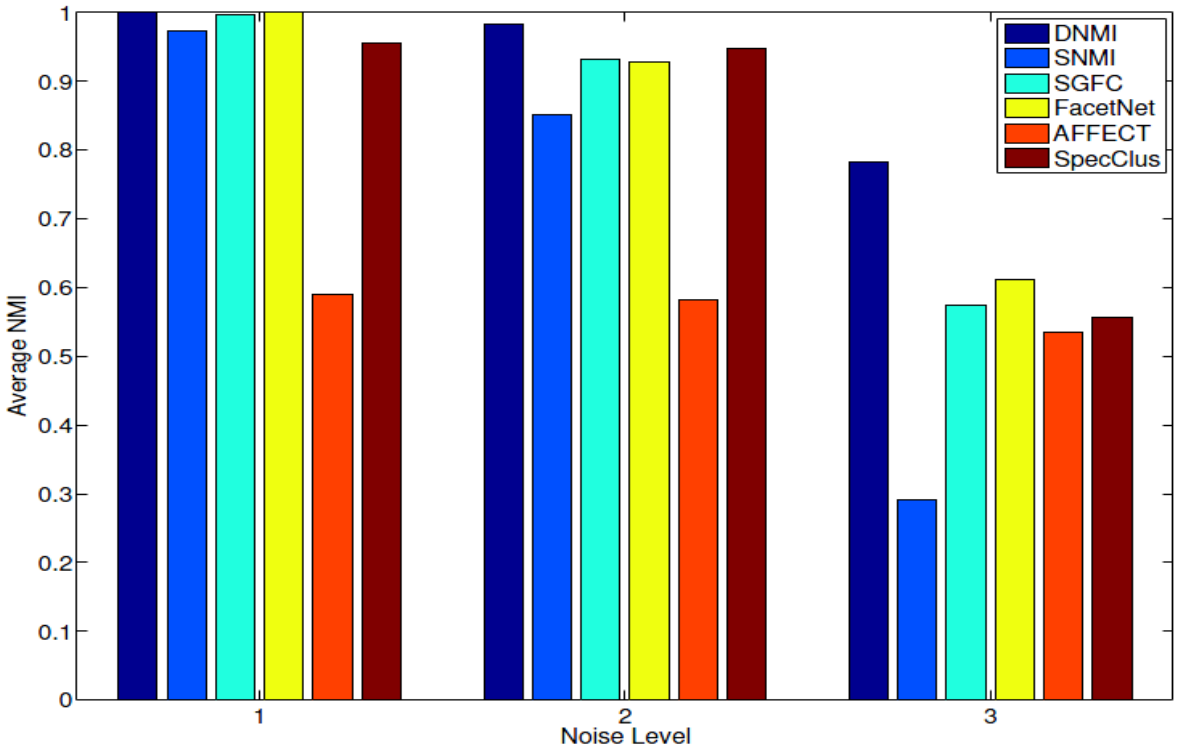
- Replacing $\mu = \dfrac{1-\alpha}{\alpha}$. We have,

$$\mathbf{H} \leftarrow \frac{\alpha \mathbf{H}}{\mathbf{W}^T \mathbf{1}_{F\times N} + \mathrm{diag}(\boldsymbol{\beta})\mathbf{H}} \cdot \left[\mathbf{W}^T \left(\frac{\mathbf{V}}{\mathbf{WH}}\right)\right] + (1-\alpha)\mathbf{W}_{t-1}$$
$$\mathbf{W} \leftarrow \frac{\alpha\ \mathbf{W}}{\mathbf{1}_{F\times N}\mathbf{H}^T + \mathbf{W}\mathrm{diag}(\boldsymbol{\beta})} \cdot \left[\left(\frac{\mathbf{V}}{\mathbf{WH}}\right)\mathbf{H}^T\right] + (1-\alpha)\mathbf{H}_{t-1}$$

$$\boldsymbol{\beta} \leftarrow \frac{F+N+2(\mathbf{a}-1)}{\mathbf{1}_{1\times F}(\mathbf{W}\cdot\mathbf{W}) + (\mathbf{H}\cdot\mathbf{H})\mathbf{1}_{N\times 1} + 2\mathbf{b}}$$

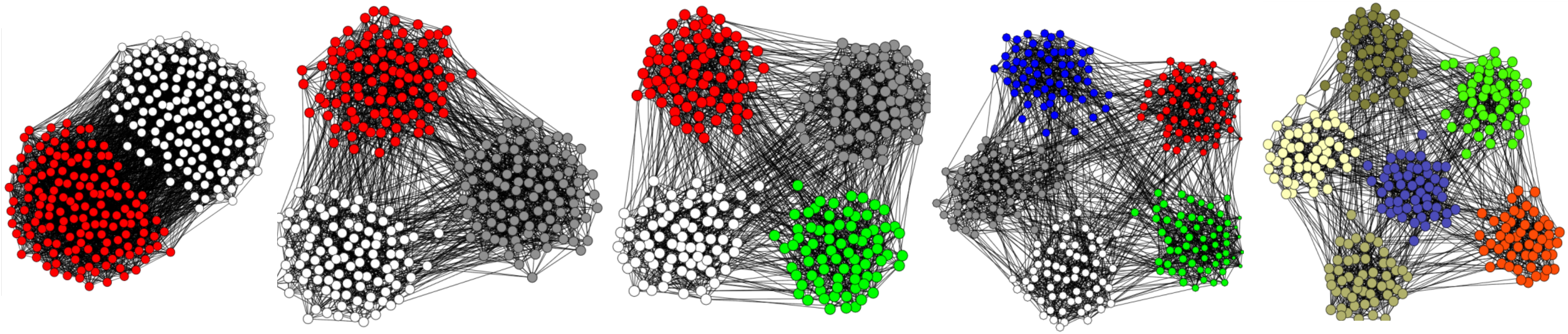
Experiment 1

- 128 nodes
- 4 clusters of 32 nodes
- Average degree =16
- Zout= 2,3,4
- Alpha=0.9

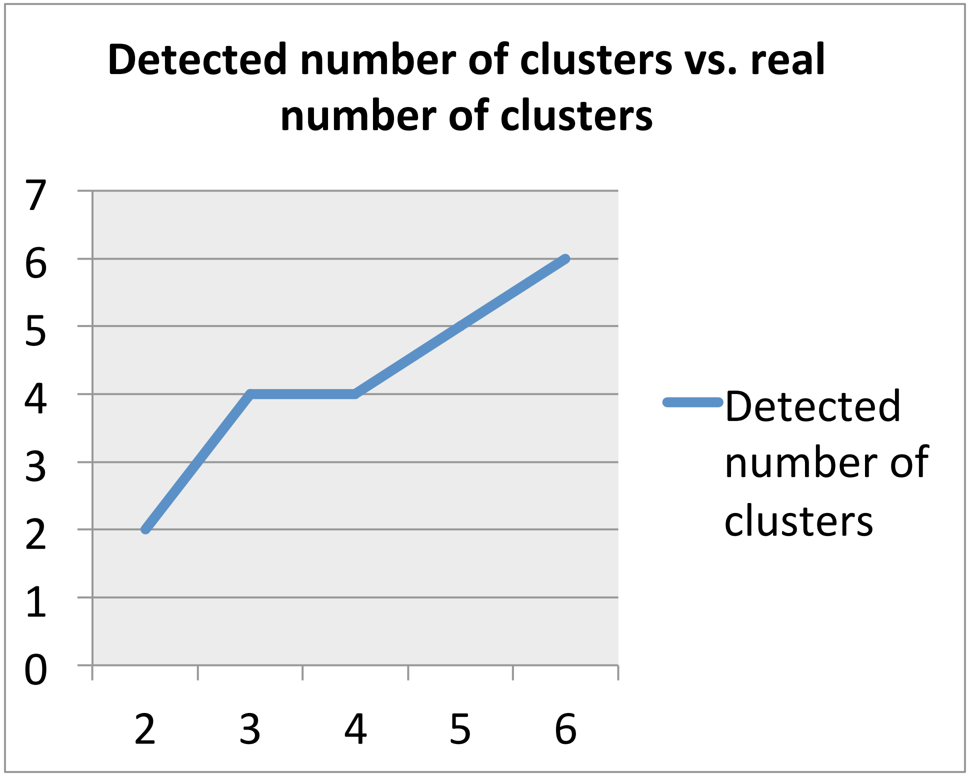


- Dynamic part: 3 nodes from each cluster leave their original cluster and join other cluster

Experiment 2



- 300 nodes
- Average degree=16
- Alpha=0.9
- Changing number of clusters



Summary and Future work

- Used non-negative matrix factorization for soft clustering
- Introduced automatic relevance determination
- Proposed dynamic model
- Parameter inference
- Showed that DMNF gives competitive results and is capable of detecting changing number of clusters
- How to set alpha? How to avoid local optimums?

Questions?

Thanks!