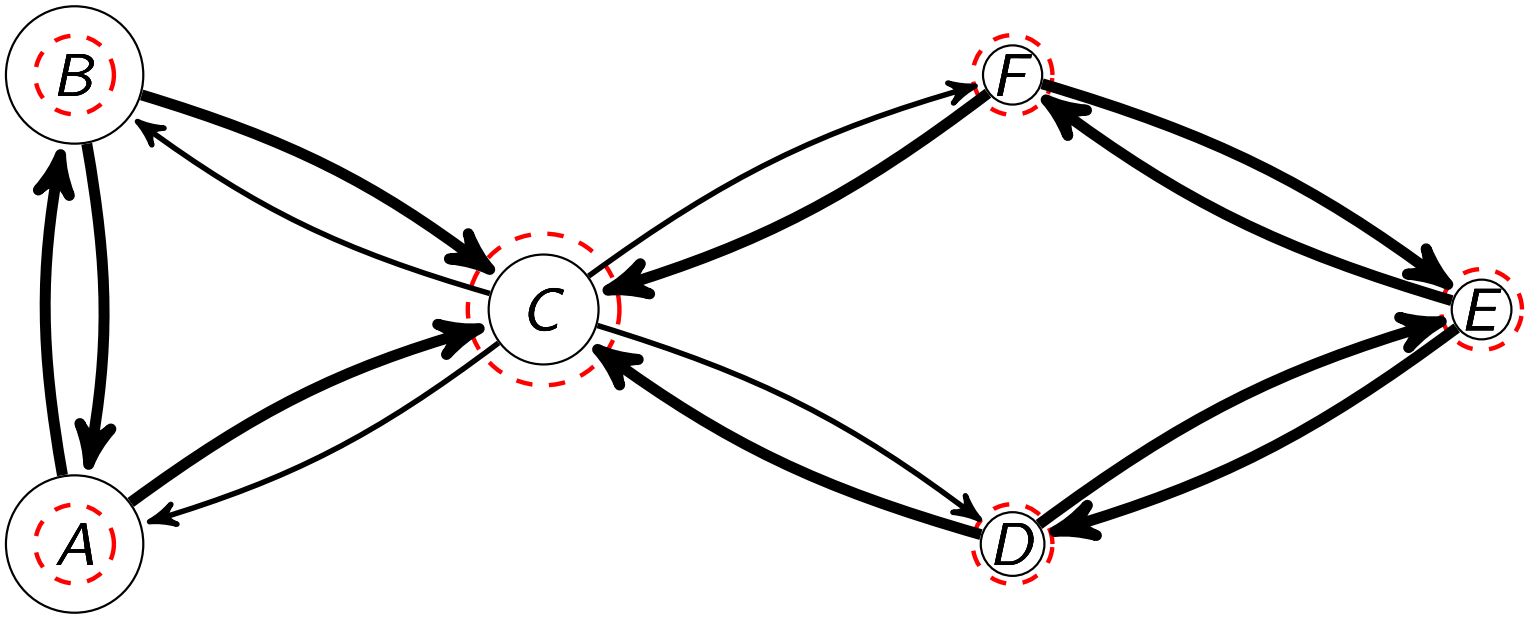
Learning the transition matrix for the weighted inverted PageRank problem

The PageRank algorithm

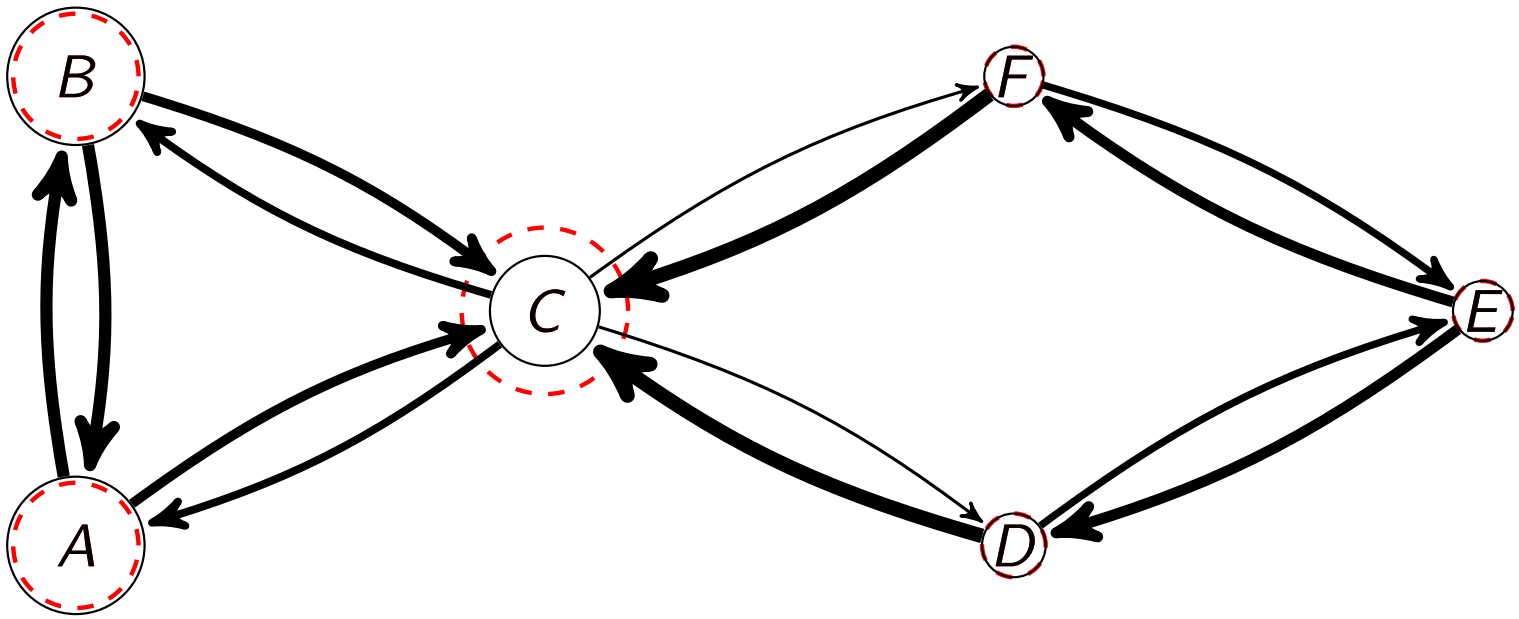


Why called inverted and weighted?

- Let's assume we know the relative importance of each node (p) in advance according to some oracle $\rightarrow$ inverted
- Find a transition matrix such that a random walker's stationary distribution (q) converges to a previously defined oracle distribution $\rightarrow$ weighted

A simple solution

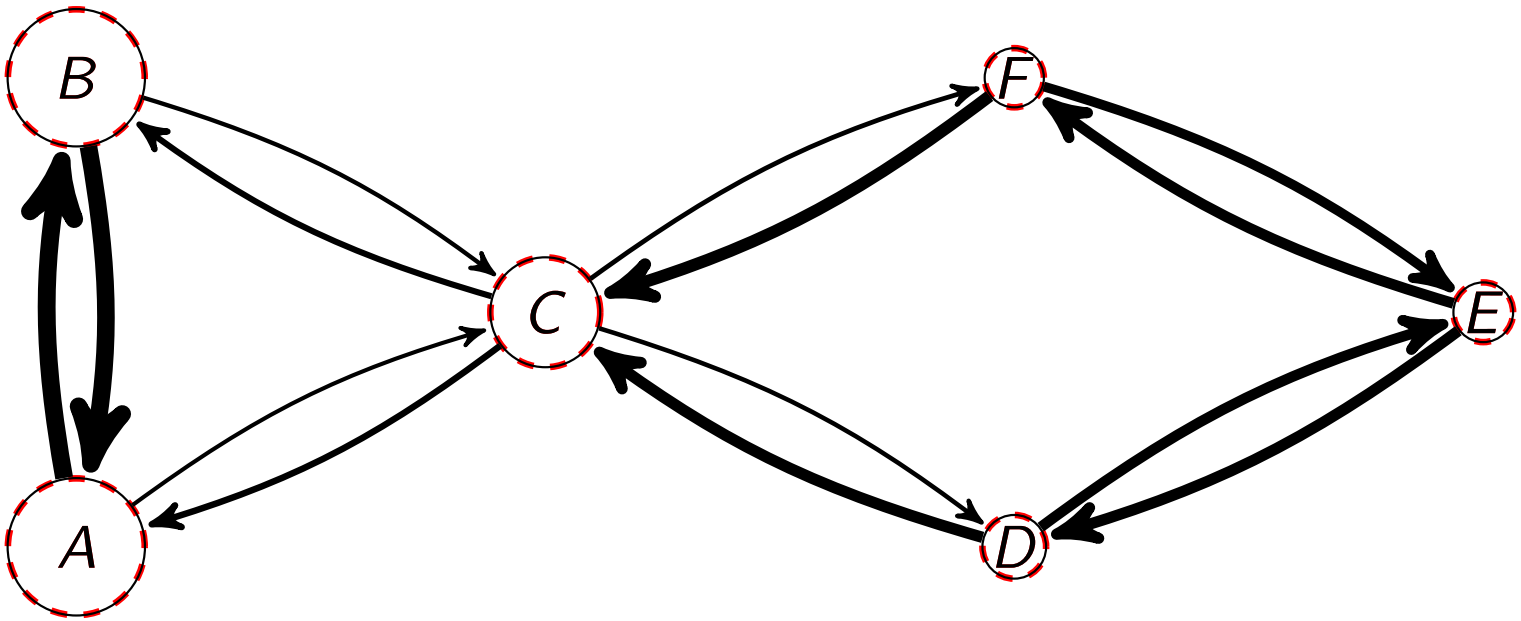
- Why not set $w_{ij} \propto p_j$? E.g. $p = [0.25, 0.25, 0.2, 0.1, 0.1, 0.1]$



$$KL(p \parallel q) = 4.4e-02 < 1.5e-01$$

A better solution

- Reminder: $p = [0.25, 0.25, 0.2, 0.1, 0.1, 0.1]$



$$KL(p \parallel q) = 2.2e-08 \ll 1.5e-01$$

What do we optimize for?

$$\min_M KL(p \parallel q) = \min_M \sum_{j=1}^{|V|} p_j \cdot \log \frac{p_j}{q_j} \tag{1}$$

We can apply the recursive definition of a node's (weighted) PageRank value, i.e.

$$q_j = \sum_{i \in In(j)} q_i \cdot Prob(i \to j) \tag{2}$$

$P(i \to j)$ can be determined applying the softmax function over the unconstrained edge weights, i.e.

$$P(i \to j) = \frac{\exp m_{ij}}{\sum_{k \in Out(i)} \exp m_{ik}} \tag{3}$$

Putting parts together

- The final unconstrained objective:

$$\min_M J(M) = \min_M \sum_{j=1}^{|V|} p_j \log \frac{p_j}{\sum_{i \in In(j)} q_i \frac{\exp m_{ij}}{\sum_{k \in Out(i)} \exp m_{ik}}}$$

- For a graph $G = (V, E)$, we have $|E|$ variables
  - $|E|$ is often around $10^7 - 10^9 \rightarrow$ quasi-Newton optimization

$$\frac{\partial J}{\partial m_{ij}} = \frac{q_i \cdot \exp m_{ij}}{\left(\sum_{k \in Out(i)} \exp m_{ik} \right)^2} \cdot \sum_{k \in Out(i)} \left(\frac{p_j}{q_j} - \frac{p_k}{q_k} \right) \cdot \exp m_{ik}$$

  - $O(|E| \cdot |V|)$ computation (luckily $|E| \ll |V|^2$ tends to hold)

A constrained setting

- An alternative constrained objective could have been:

$$\min_W J(W) = \min_W \sum_{j=1}^{|V|} p_j \log \frac{p_j}{\sum_{i \in In(j)} q_i w_{ij}}$$

$$\text{subject to} \quad w_{ij} \geq 0, \ i = 1, \ldots, |V|$$

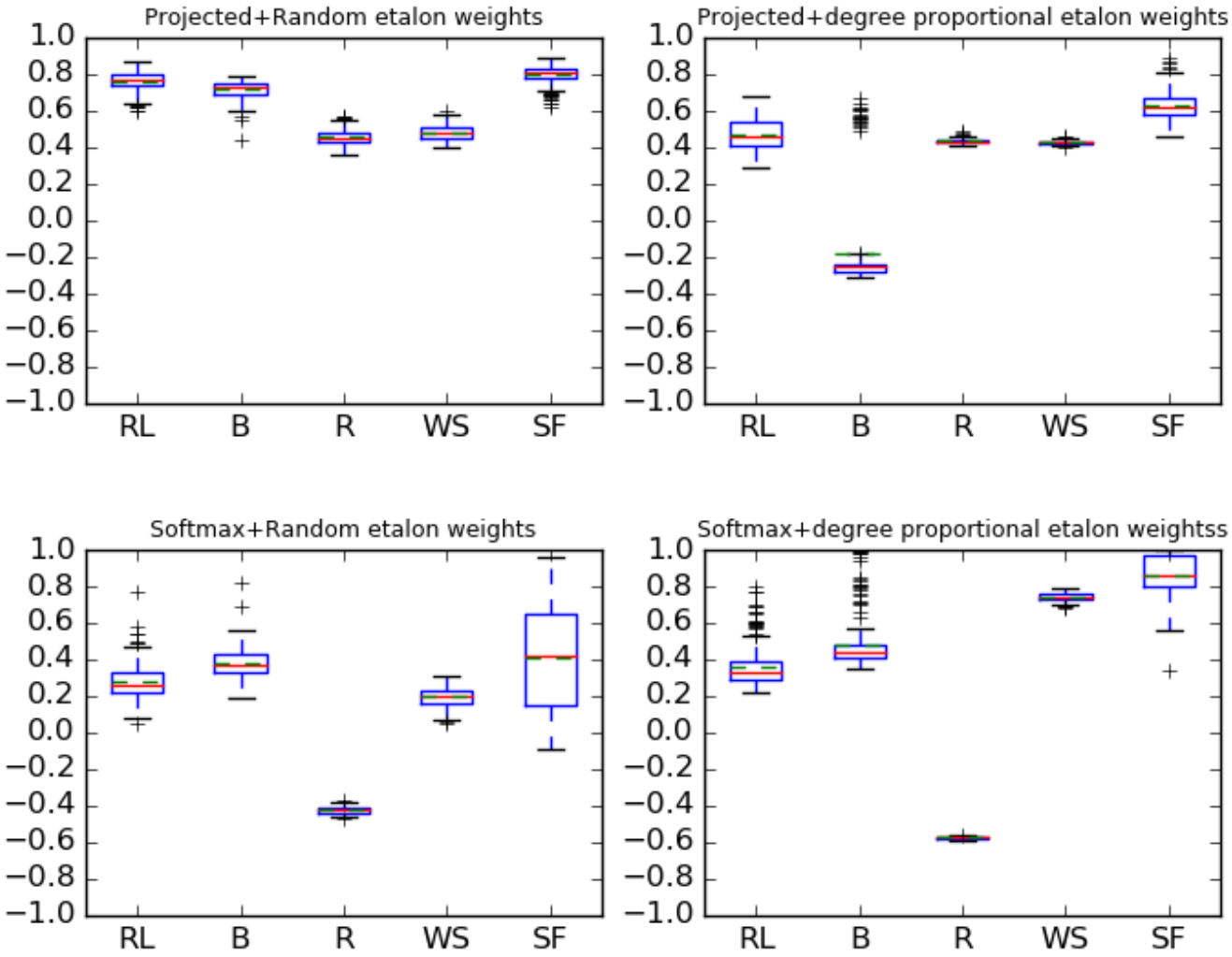
$$\sum_{i=1}^{m} w_{ij} = 1, \ i = 1, \ldots, |V|.$$

- $O(|E|)$ computation for the unconstrained gradient
- **Projected** quasi-Newton method
  - Instead of $W_{i+1} \leftarrow W_i - \alpha B(W_i)^{-1} \nabla_{W_i}$ use $W_{i+1} \leftarrow P[W_i - \alpha B(W_i)^{-1} \nabla_{W_i}]$ for updating
  - Downside
    - Projection takes $O(|V|^2 \log(|V|))$ (worst case) computation
    - We lose some of the nice properties of the optimizer

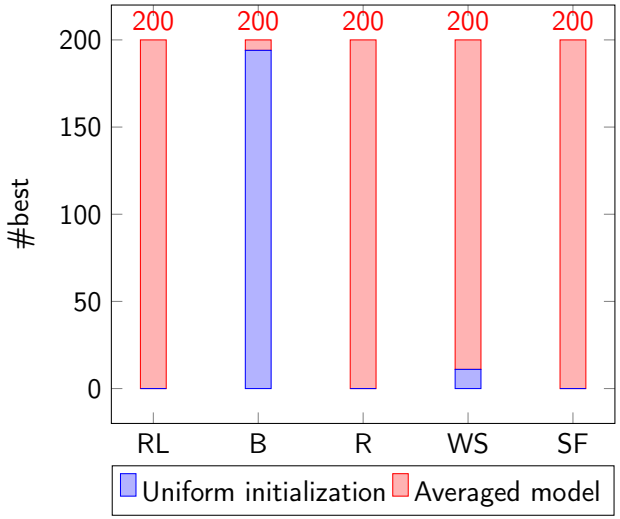
Synthetic experiments

- Are solutions with low KL-divergence really what we want? No.
- Graphs were generated from 5 graph types
  - Ring lattice (RL)
  - Bipartite (B)
  - Erdős-Rényi (R)
  - Watts-Strogatz (WS)
  - Scale-free (SF)
- Gold standard edge weights were either
  - Randomly chosen
  - Drawn from a Dirichlet prior according to the degrees of neighbor nodes
- 200 graphs of each 10 (5*2) types were generated
- Optimization was performed 200 times starting with different initial guesses for edge weights
  - One run initialized with uninformed weights, i.e. $w_{ij} = \frac{1}{OutDeg(i)}$
  - In the remaining cases the initial w_{ij}s were selected randomly
  - An averaged meta-model of the 200 runs was formed then

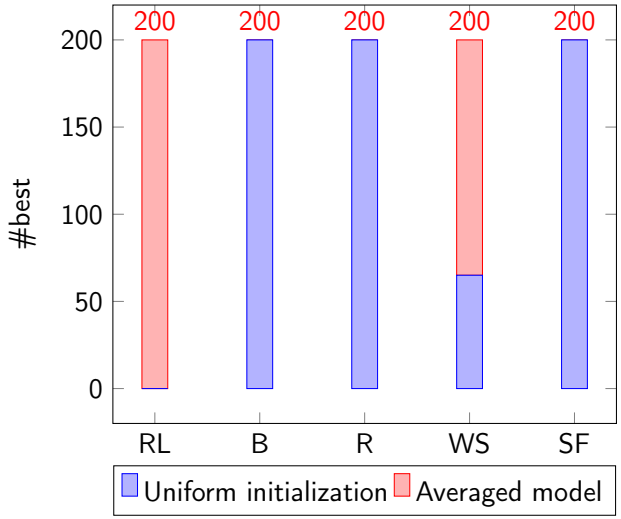
Synthetic experiments – correlation



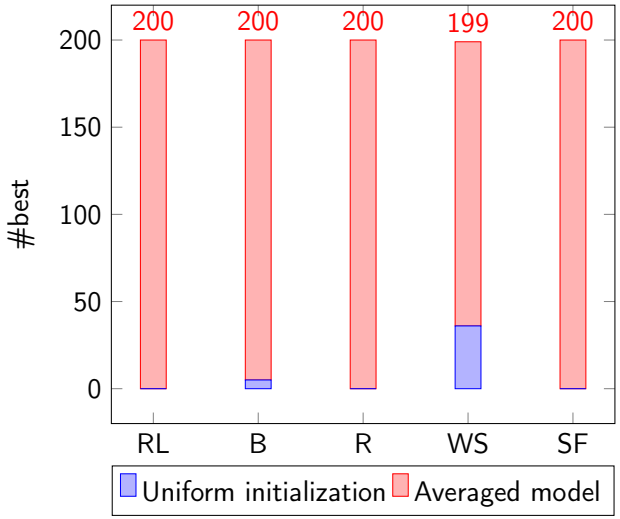
Synthetic experiments – The effect of averaging



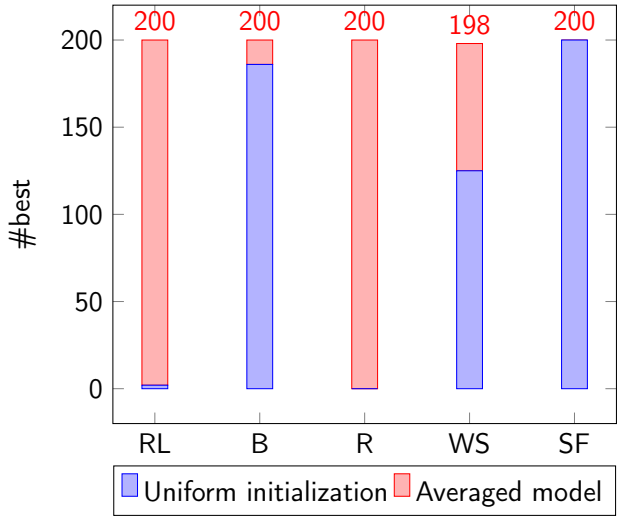
(a) Projected+random



(b) Projected+degree

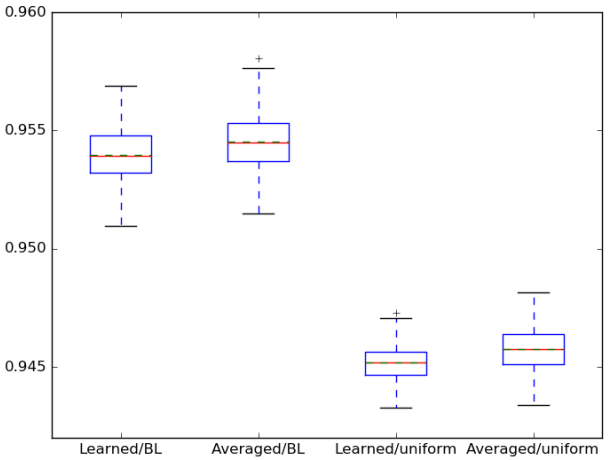


(c) Softmax+random

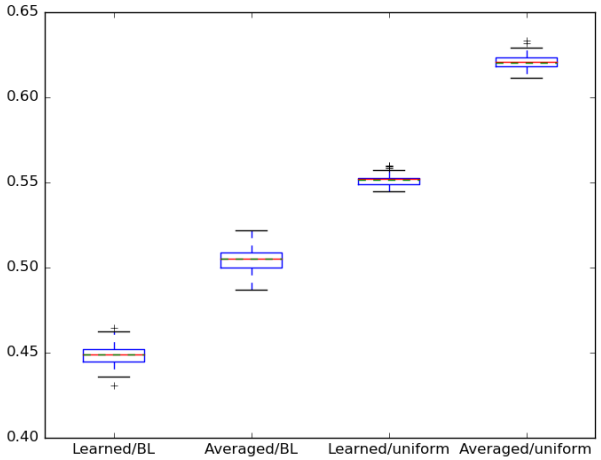


(d) Softmax+degree

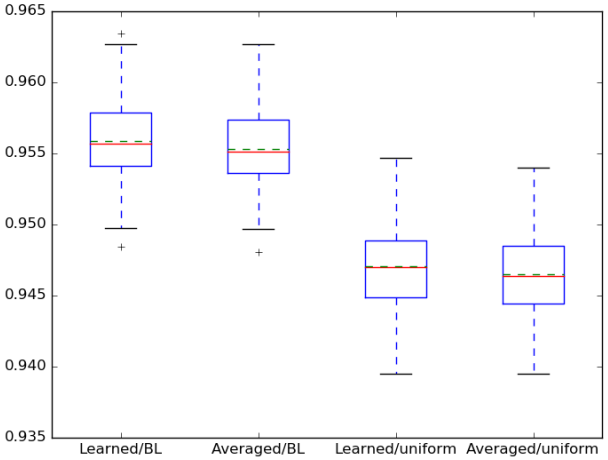
Synthetic experiments – relative MSE on random networks



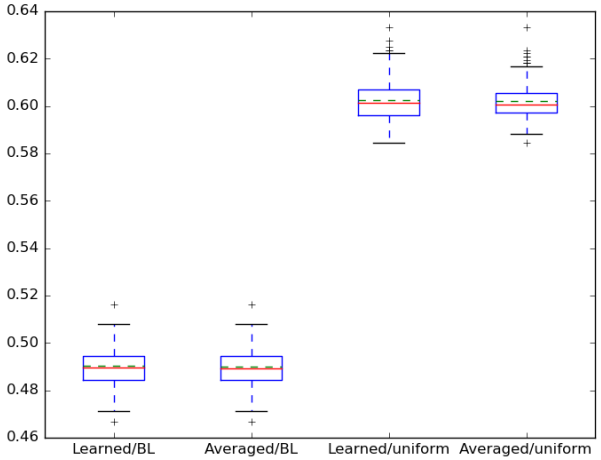
(a) Projected+random



(b) Projected+degree

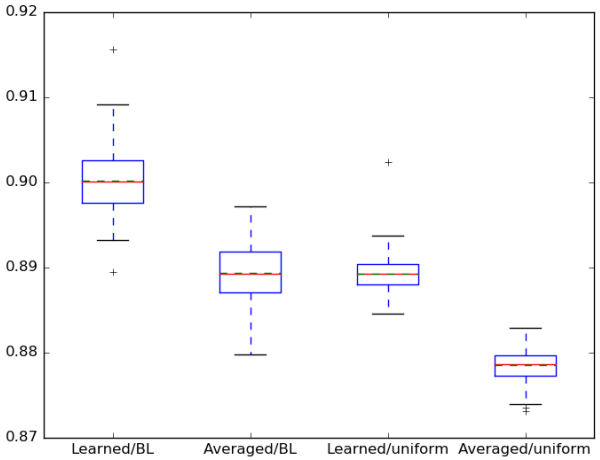


(c) Softmax+random

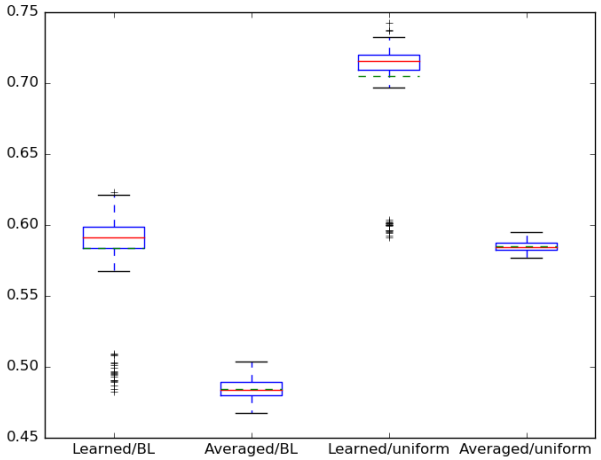


(d) Softmax+degree

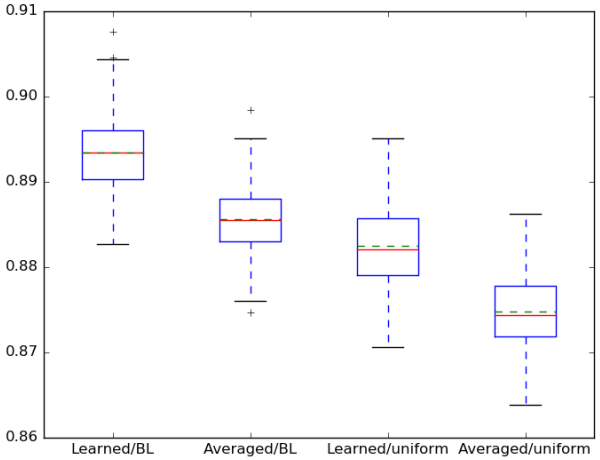
Synthetic experiments – relative MSE on bipartite networks



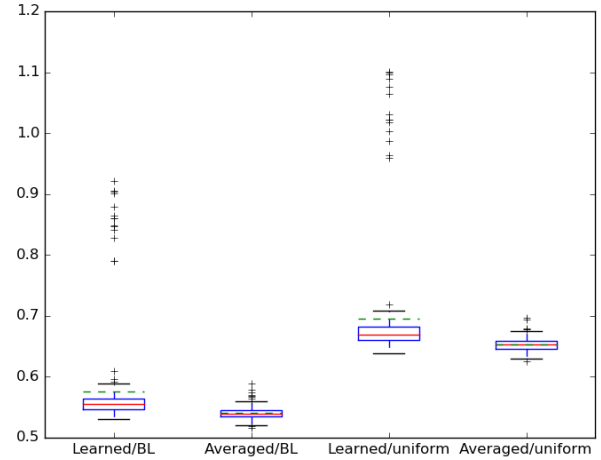
(a) Projected+random



(b) Projected+degree

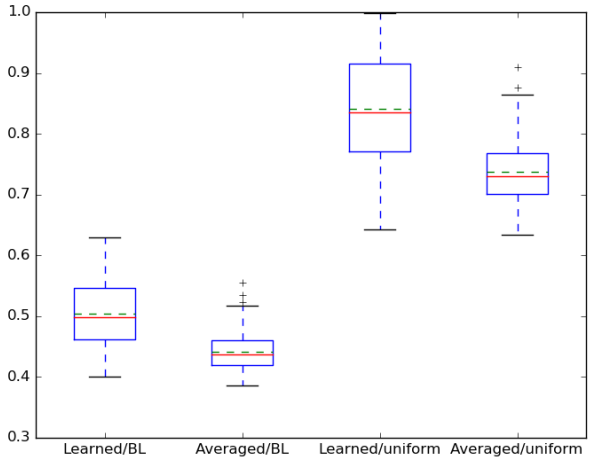


(c) Softmax+random

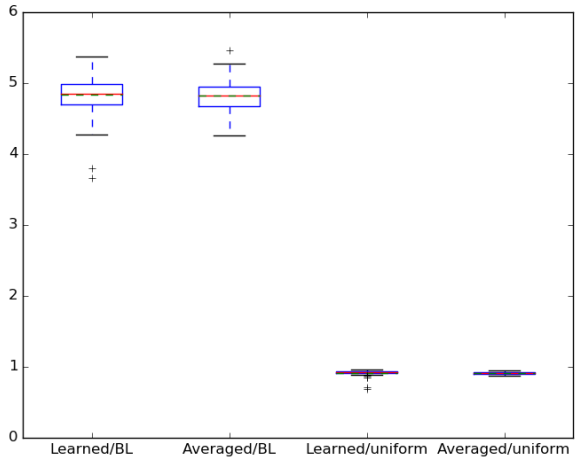


(d) Softmax+degree

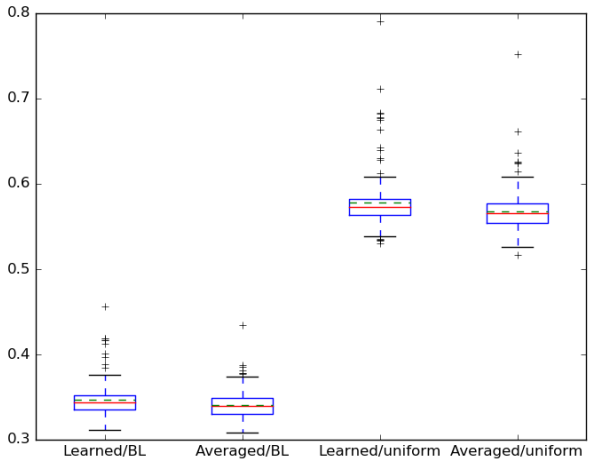
Synthetic experiments – relative MSE on scale-free networks



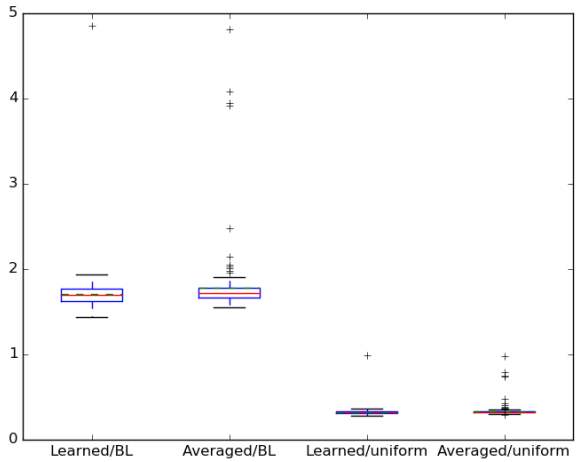
(a) Projected+random



(b) Projected+degree



(c) Softmax+random



(d) Softmax+degree

Experiments on collaboration networks

- Let $G = (V, E)$ be a graph in which
  - $i \in V$ are authors, $|V| \approx 16.6K$
  - $(i, j) \in E \Leftrightarrow i$ and j are co-authors, $|E| \approx 114K$

Avgd. projected		PMI baseline	
Berend, Gabor	w		PMI
->Farkas, Richard	0.189	->Berend, Gabor	-2.89
->Vincze, Veronika	0.186	->Hangya, Viktor	-3.58
->Berend, Gabor	0.158	->Nagy, Istvan	-5.01
->Zarriess, Sina	0.157	->Vincze, Veronika	-5.72
->Nagy, Istvan	0.156	->Farkas, Richard	-6.2
->Hangya, Viktor	0.153	->Zarreiss, Sina	-6.3

- 226 authors with >=5 collaborators such that the nodes with biggest weight are themselves
  - Typically big players including e.g. Oren Etzoni, Dan Klein, Jun'ichi Tsujii, Janyce Wiebe, György Szarvas

Language usage graphs and language modeling

- Let $G = (V, E)$ be a graph in which
  - Vertices represent n consecutive tokens (n-grams)
  - $(i, j) \in E \Leftrightarrow$ n-gram j succeeds n-gram i
    - i and j has to be such that $j[k-1] = i[k], \forall 2 \leq k \leq n$
- w_{ij} can then be interpreted as the probability of seeing n-gram j after n-gram i is observed, i.e. $P(j[n]|i)$
  - $P(j[n]|i)$ can be calculated in a ML fashion
    - Doing so, however, requires the maintenance of co-occurrence counts for any possible pairs of n-grams

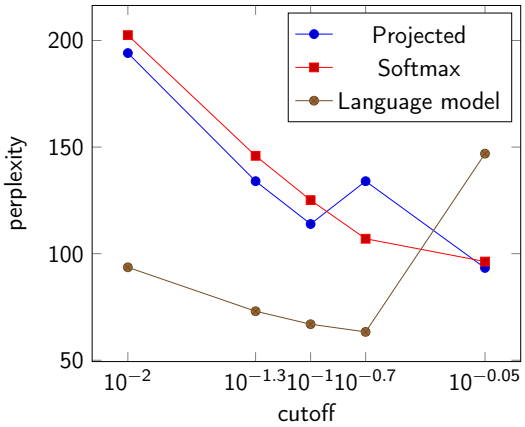
Language modeling

- $40K/2.5K$ English train/test sentences ($950K/57K$ words)
- Projected and softmax models with 25 initializations each

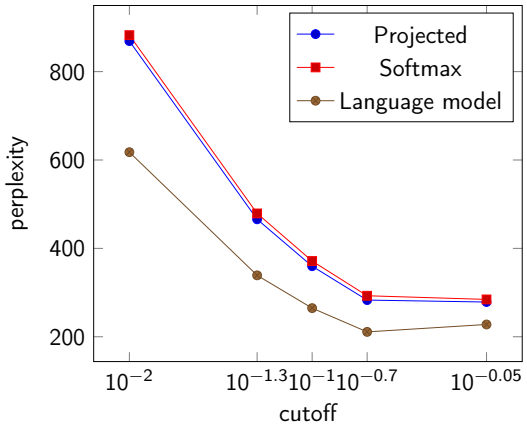
n	|V|	|E|	$\text{MSE}_{projected}$	$\text{MSE}_{softmax}$
1	39,312	325,607	8.4e-2	2.1e-2
2	325,609	643,958	7.5e-3	5.1e-3
3	643,960	802,160	2.5e-3	2.0e-3

Language model quality

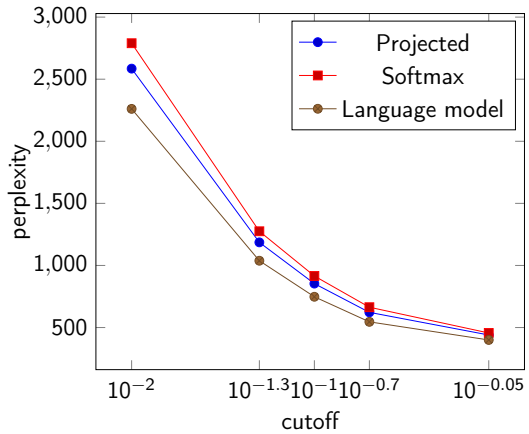
- Perplexity $= 2^{-LL(\text{test set})}$
- Cut-off (smoothing) has to be applied



(a) 1st order models



(b) 2nd order models



(c) 3rd order models

Exemplar generated sentences

- The country abandoned its former devotion to a glitzy life that drilling for gas supply . '' said William Seidman , who had the power of government employees ' favored charities .
- The bank made emergency loans to less-developed countries .
- The markets are returning to normalcy .

Experiments on Wikipedia

- Hungarian (20/10/2015) and English (02/10/2015) Wikipedia
  - $|V_{hu}| \approx 500K$, $|E_{hu}| \approx 40M$
  - $|V_{en}| \approx 12M$, $|E_{en}| \approx 375M$
- Node importances were the relative query hits each page received during 20/10/2015
- 5 iterations run for both the projected and softmax models

Avg. runtime/iter.	hu	en
Projected	<3h	<8h
Softmax	<0.5h	<3h

  - Almost no variance in the runtime of the softmax models
    - However, runtimes for the projected models were $\in [1.25, 5.25]$ hours for Hungarian and $\in [4, 13]$ hours for English

Quantitative results on Wikipedia

- Baseline KLs of 1.05 and 1.74 for Hungarian and English

Model	Projected	Softmax
1	$1.07 \rightarrow 0.44$	$1.07 \rightarrow 1.07$
2	$1.09 \rightarrow 0.51$	$1.07 \rightarrow 1.07$
3	$1.10 \rightarrow 0.32$	$1.07 \rightarrow 1.07$
4	$1.09 \rightarrow 0.29$	$1.07 \rightarrow 1.07$
5	$1.10 \rightarrow 0.38$	$1.07 \rightarrow 1.06$

Table : Results on the Hungarian Wikipedia

Model	Projected	Softmax
1	$1.82 \rightarrow 0.70$	$1.82 \rightarrow 1.79$
2	$1.86 \rightarrow 0.72$	$1.82 \rightarrow 1.79$
3	$1.80 \rightarrow 0.75$	$1.82 \rightarrow 1.79$
4	$1.78 \rightarrow 0.72$	$1.82 \rightarrow 1.79$
5	$1.84 \rightarrow 0.79$	$1.82 \rightarrow 1.79$

Table : Results on the English Wikipedia

Qualitative results on Wikipedia – strong pairs

- For each node $i \in V$ of graph $G = (V, E)$, its strongest neighbor is $s_i = \arg\max_{j \in V, j \neq i} w_{ij}$
- (i, j) is a strong pair if $s_i = j$ and $s_j = i$ holds
- There were approx. 16K such pairs on the English Wikipedia

Article_i	Article_j
Johnelle Hunt	Johnnie Bryan Hunt
Wendy Mesley	Peter Mansbridge
Kaley Cuoco	Ryan Sweeting
Labyrinth	Daedalus
Element 115	Ununpentium
MITI	Ministry of International Trade and Industry

Table : Sample strong pairs of the English Wikipedia

Qualitative results on Wikipedia – ‘Association chains‘

- Given a starting article always proceed along edges with the highest weight
  - Machine learning→Data science→Genomics→Illumina dye sequencing→Cytosine→APOBEC
  - Hungary→Hungarian notation→Varchar→Data→Standard deviation→68–95–99.7 rule
  - Sagrada Família→Catalan Modernism→Modernisme→Turn of the century→Information Age→Mark Zuckerberg

Conclusion

- Edge weights reflecting the strength of connections seems to be a task that can be learned
- The network topology and the relative importances of the nodes are needed for that
- Determining the relative importance scores of the nodes is a crucial part
  - Most likely requires some domain knowledge and/or luck
- The softmax model is faster but the projected model produced nicer numbers (not always)

Future directions

- Find shortest path in a graph based on the edge weights as kind of a ‘heuristic‘ (6 degrees of separation)
- Link prediction and frequent itemset mining?
- Possible relation to the previous talks of the semester?
- Analyzing how regularization affect these models