Automated Gene Classification using Nonnegative Matrix Factorization on Biomedical Literature

Kevin Heinrich

PhD Dissertation Defense
Department of Computer Science, UTK

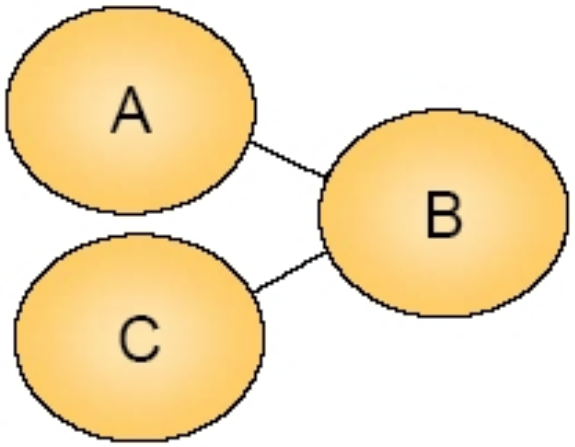
March 19, 2007

What Is The Problem?

- Understanding functional gene relationships requires expert knowledge.
- Gene sequence analysis does not necessarily imply function.
- Gene structure analysis is difficult.
- Issue of scale.
  - Biologists know a small subset of genes.
  - Thousands of genes.
- Time & Money.

Defining Functional Gene Relationships

- Direct Relationships.
  - Known gene relationships (e.g. A-B).
  - Based on term co-occurrence.[1]
- Indirect Relationships.
  - Unknown gene relationships (e.g. A-C).
  - Based on *semantic structure.*



[1] Jenssen et al., *Nature Genetics*, 28:21, 2001.

Semantic Gene Organizer

- Gene information is compiled in human-curated databases.
  - Medical Literature, Analysis, and Retrieval System Online (MEDLINE)
  - EntrezGene (LocusLink)
  - Medical Subject Heading (MeSH)
  - Gene Ontology (GO)
- *Gene documents* are formed by taking titles and abstracts from MEDLINE citations cross-referenced in the Mouse, Rat, and Human EntrezGene entries for that gene.
- Examines literature (phenotype) instead of genotype.
- Can be used as a guide for future gene exploration.

Vector Space Model

- Gene documents are parsed into *tokens*.
- Tokens are assigned a weight, w_{ij}, of ith token in j^{th} document.
- An $m \times n$ term-by-document matrix, A, is created.

$$A = [w_{ij}]$$

  - Genes are m-dimensional vectors.
  - Tokens are n-dimensional vectors.

Term-by-Document Matrix

	d_1	d_2	d_3	$\dots$	d_n
t_1	w_{11}	w_{12}	w_{13}		w_{1n}
t_2	w_{21}	w_{22}	w_{23}		w_{2n}
t_3	w_{31}	w_{32}	w_{33}		w_{3n}
t_4	w_{41}	w_{42}	w_{43}		w_{4n}
$\vdots$				$\ddots$	
t_m	w_{m1}	w_{m2}	w_{m3}		w_{mn}

Typically, a term-document matrix is sparse and unstructured.

Weighting Schemes

- Term weights are the product of a local, global component, and document normalization factor.

$$w_{ij} = l_{ij} g_i d_j$$

- The log-entropy weighting scheme is used where

$$\begin{aligned} l_{ij} &= \log_2 (1 + f_{ij}) \\ g_i &= 1 + \left(\frac{\sum_j (p_{ij} \log_2 p_{ij})}{\log_2 n} \right), p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}} \end{aligned}$$

Latent Semantic Indexing (LSI)

LSI performs a truncated singular value decomposition (SVD) on M into three factor matrices

$$A = U\Sigma V^T$$

- U is the $m \times r$ matrix of eigenvectors of AA^T
- V^T is the $r \times n$ matrix of eigenvectors of $A^T A$
- Σ is the $r \times r$ diagonal matrix of the r nonnegative singular values of A
- r is the rank of A

SVD Properties

- A rank-k approximation is generated by truncating the first k column of each matrix, i.e., $A_k = U_k \Sigma_k V_k^T$
- A_k is the closest of all rank-k approximations, i.e., $\|A - A_k\|_F \leq \|A - B\|$ for any rank-k matrix B

SVD Querying

- Document-to-Document Similarity
$$A_k^T A_k = (V_k \Sigma_k)(V_k \Sigma_k)^T$$
- Term-to-Term Similarity
$$A_k A_k^T = (U_k \Sigma_k)(U_k \Sigma_k)^T$$
- Document-to-Term Similarity
$$A_k = U_k \Sigma_k V_k^T$$

Advantages of LSI

- A is sparse, factor matrices are dense. This causes improved recall for concept-based matching.
- Scaled document vectors can be computed once and stored for quick retrieval.
- Components of factor matrices represent concepts.
- Decreasing number of dimensions compares documents in a broader sense and achieves better compression.
- Similar word usage patterns get mapped to same geometric space.
- Genes are compared at a concept level rather than a simple term co-occurrence level resulting in vocabulary independent comparisons.

Presentation of Results

Problem:

- Biologists are familiar with interpreting trees.
- LSI produces ranked lists of related terms/documents.

Solution:

- Generate pairwise distance data, i.e., $1 - \cos\theta_{ij}$
- Apply distance-based tree-building algorithm
  - Fitch - $O(n^4)$
  - NJ - $O(n^3)$
  - FastME - $O(n^2)$

Defining Functional Gene Relationships on Test Data

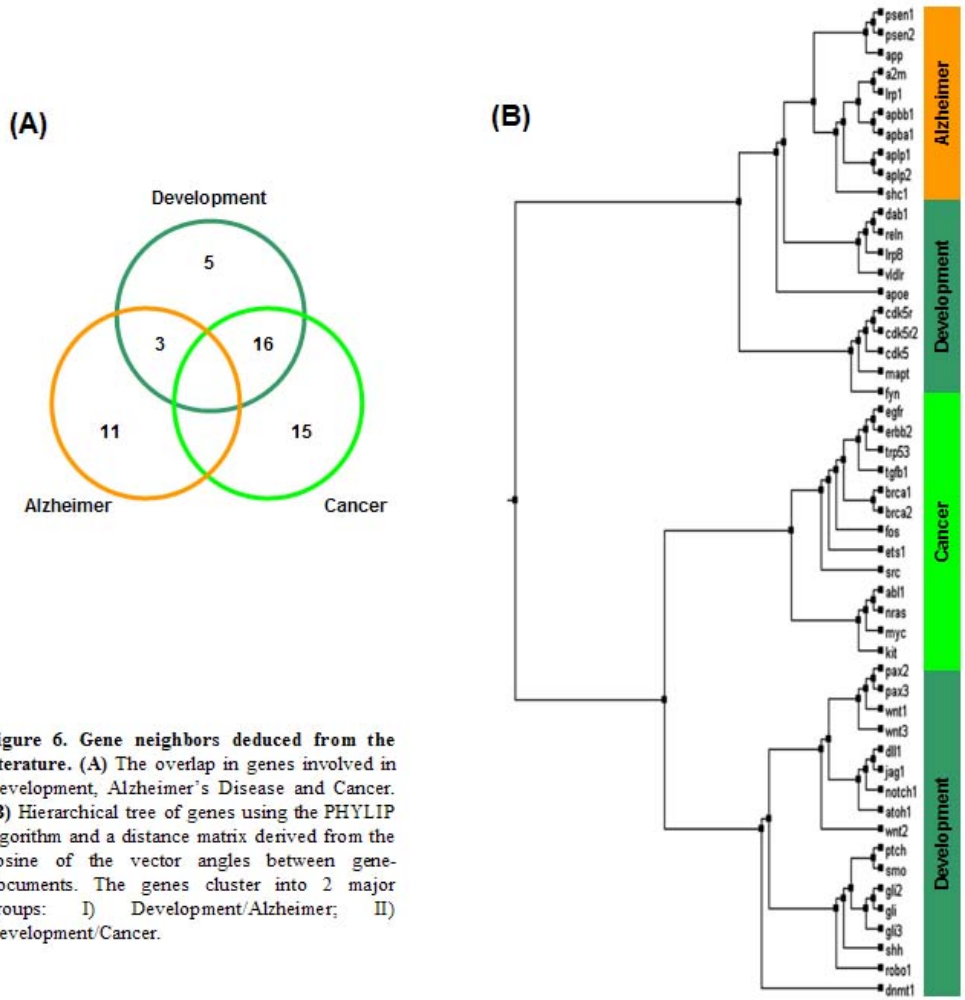


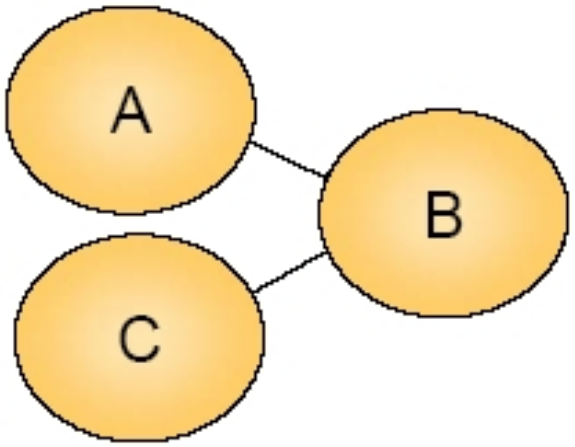
Figure 6. Gene neighbors deduced from the literature. (A) The overlap in genes involved in Development, Alzheimer's Disease and Cancer. **(B)** Hierarchical tree of genes using the PHYLIP algorithm and a distance matrix derived from the cosine of the vector angles between gene-documents. The genes cluster into 2 major groups: I) Development/Alzheimer; II) Development/Cancer.

"Problems" with LSI

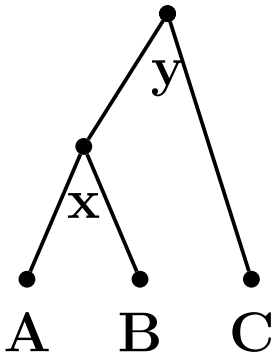
- Initial term weights are nonnegative; SVD introduces negative components.
- Dimensions of factored space do not have an immediate interpretation.
- Want advantages of factored/reduced dimension space, but want to interpret dimensions for clustering/labeling trees.
- Issue of scale—understand small collections better rather than huge collections.

Defining Functional Gene Relationships

- Direct Relationships.
  - Known gene relationships (e.g. A-B).
  - Based on term co-occurrence.[2]
- Indirect Relationships.
  - Unknown gene relationships (e.g. A-C).
  - Based on *semantic structure.*



- Label Relationships (e.g. x & y).



[2] Jenssen et al., *Nature Genetics*, 28:21, 2001.

NMF Problem Definition

Given nonnegative V, find W and H such that

$$V \approx WH$$

- $W, H \geq 0$
- W has size $m \times k$
- H has size $k \times n$

NMF Problem Definition

Given nonnegative V, find W and H such that

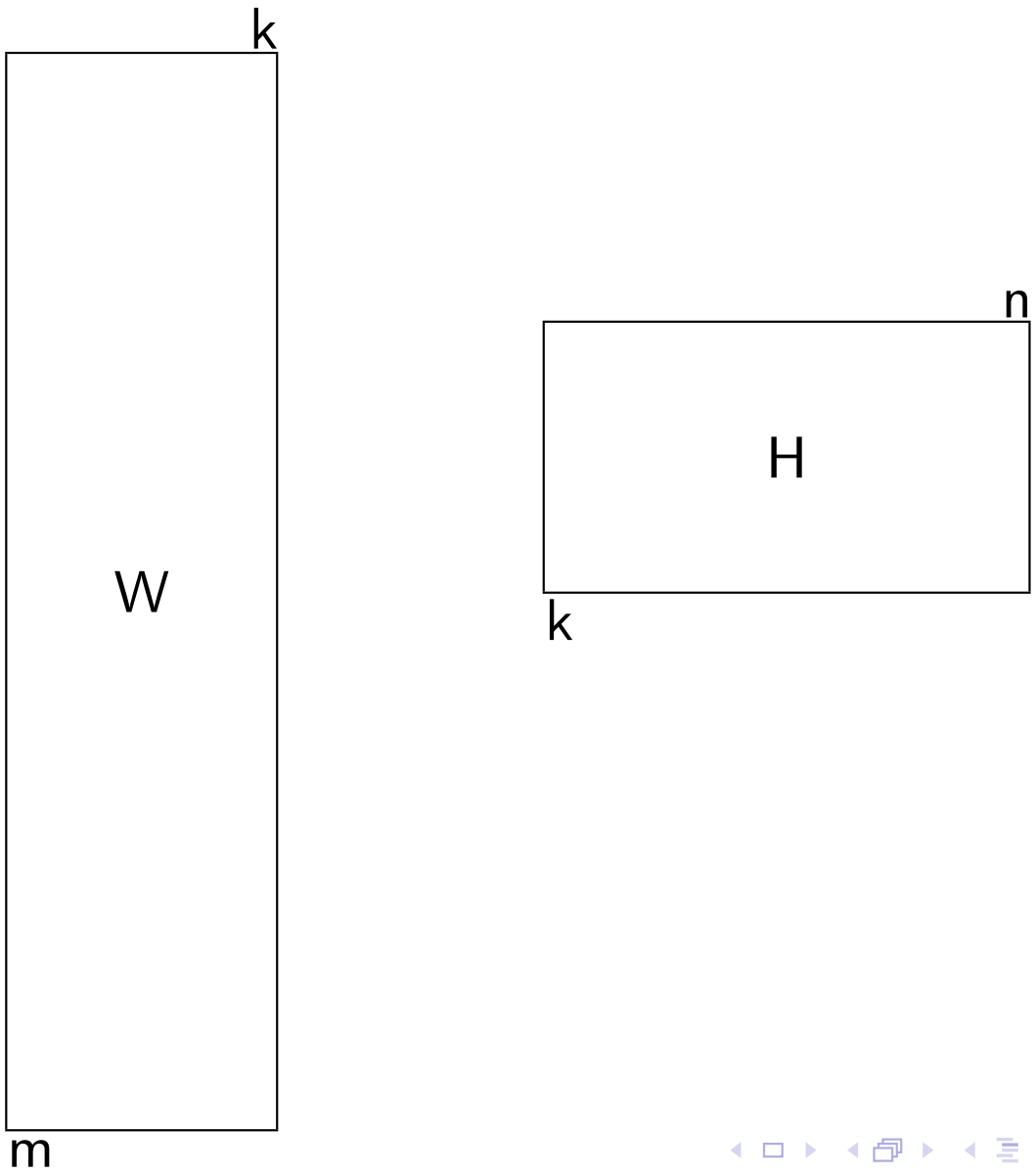
$$V \approx WH$$

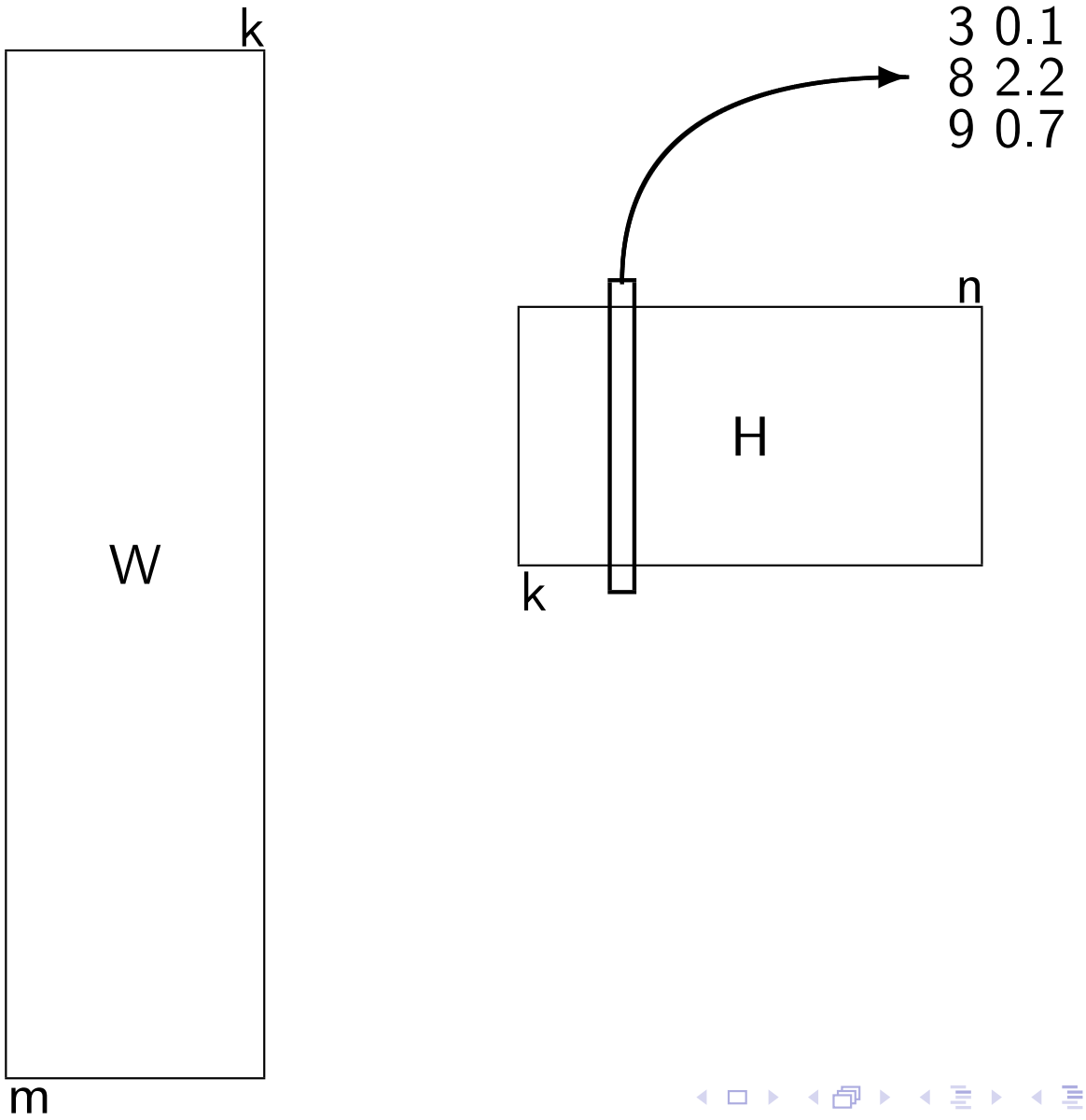
- $W, H \geq 0$
- W has size $m \times k$
- H has size $k \times n$
- W and H are not unique.
  i.e., $WDD^{-}1H$ for any invertible nonnegative D

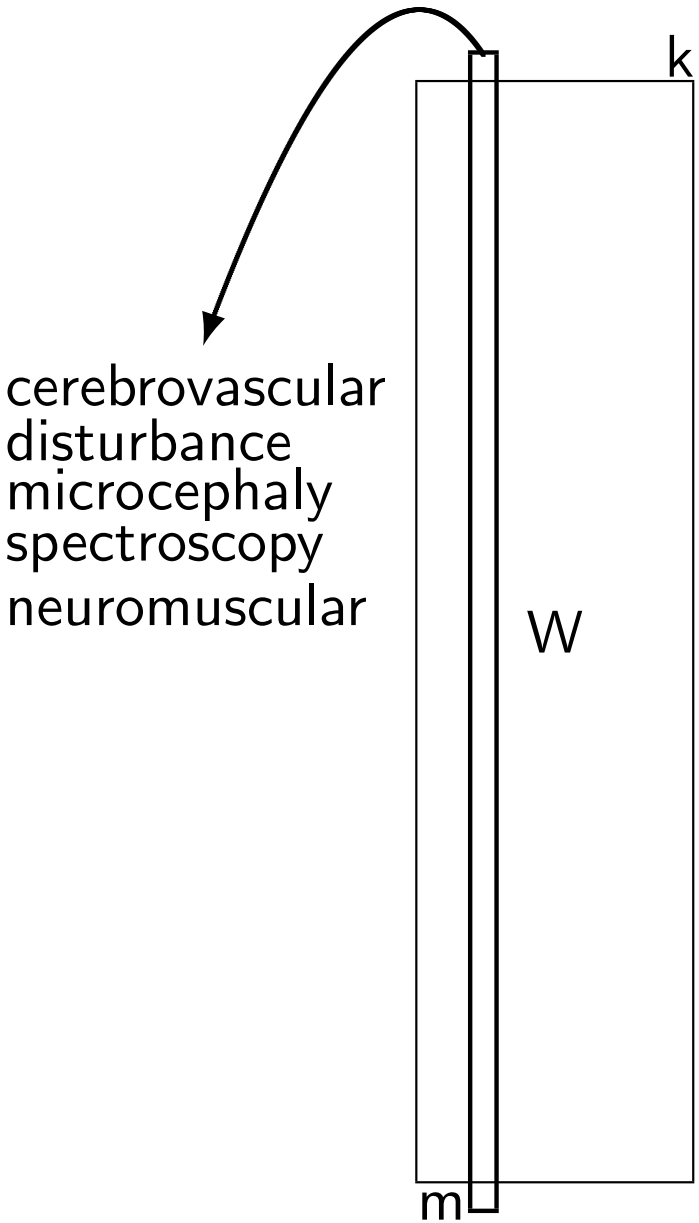
NMF Interpretation

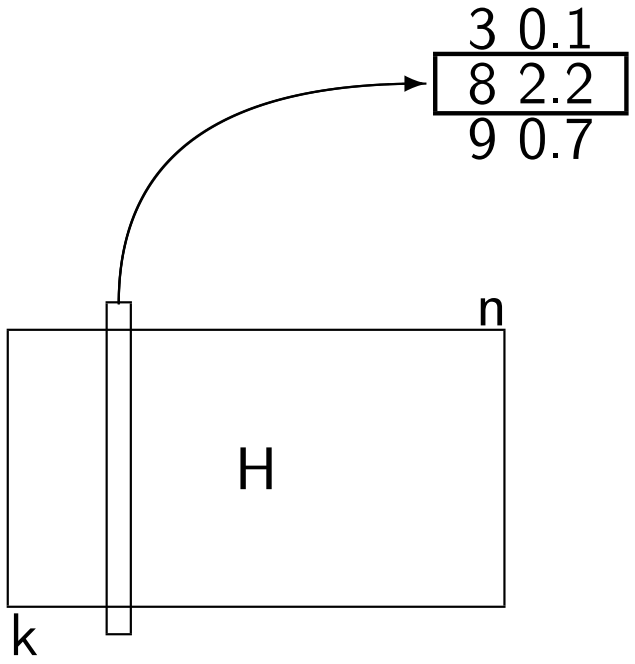
$$V \approx WH$$

- Columns of W are k "feature" or "basis" vectors; represent semantic concepts.
- Columns of H are linear combinations of feature vectors to approximate corresponding column in V.
- Choice of k determines accuracy and quality of basis vectors.
- Ultimately produces a "parts-based" representation of the original space.


k
W
m
n
H
k


k
W
m
3 0.1
8 2.2
9 0.7
n
H
k


k
cerebrovascular
disturbance
microcephaly
spectroscopy
neuromuscular
W
m


3 0.1
8 2.2
9 0.7
n
H
k

Euclidean Distance (Cost Function)

$$E\left(W,H\right) = \|V - WH\|_F^2 = \sum_{i,j} \left(V_{ij} - \left(WH\right)_{ij}\right)^2$$

- Minimize $E(W,H)$ subject to $W,H \geq 0$.
- $E(W,H) \geq 0$.
- $E(W,H) = 0$ if and only if $V = WH$.
- $\|V - WH\|$ convex in W or H separately, not both simultaneously.
- No guarantee to find global minima.

Initialization Methods

Since NMF is an iterative algorithm, W and H must be initialized.

- Random positive entries.
- Structured initialization typically speeds convergence.
  - Run k-means on V.
  - Choose representative vector from each cluster to form W and H.

Most methods do not provide static starting point.

Non-Negative Double SVD

- NNDSVD is one way to provide a static starting point.[3]
- Observe $A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T$, i.e. sum of rank-1 matrices
- Foreach j
  - Compute $C = u_j v_j^T$
  - Set to 0 all negative elements of C
  - Compute maximum singular triplet of C, i.e., $[\hat{u}, \hat{s}, \hat{v}]$
  - Set jth column of W to $\hat{u}$ and jth row of H to $\sigma_j \hat{s} \hat{v}$
- Resulting W and H are influenced by SVD.

[3] Boutsidis & Gallopoulos, Tech Report, 2005

NNDSVD Variations

Zero elements remain "locked" during MM update.

- NNDSVDz keeps zero elements.
- NNDSVDe assigns $\epsilon = 10^{-9}$ to zero elements.
- NNDSVDa assigns average value of A to zero elements.

Update Rules

Update rules should

- decrease the approximation.
- maintain nonnegativity constraints.
- maintain other constraints imposed by the application (smoothness/sparsity).

Multiplicative Method (MM)

$$H_{cj} \leftarrow H_{cj} \frac{\left(W^T V\right)_{cj}}{\left(W^T W H\right)_{cj} + \epsilon}$$

$$W_{ic} \leftarrow W_{ic} \frac{\left(V H^T\right)_{ic}}{\left(W H H^T\right)_{ic} + \epsilon}$$

- ϵ ensures numerical stability.
- Lee and Seung proved MM non-increasing under Euclidean cost function.
- Most implementations update H and W "simultaneously."

Other Objective Functions

$$\|V - WH\|_F^2 + \alpha J_1(W) + \beta J_2(H)$$

α and β are parameters to control level of additional constraints.

Smoothing Update Rules

For example, set $J_2(H) = \|H\|_F^2$ to enforce smoothness on H to try to force uniqueness on W.[4]

$$H_{cj} \leftarrow H_{cj} \frac{\left(W^T V\right)_{cj} - \beta H_{cj}}{\left(W^T W H\right)_{cj} + \epsilon}$$

$$W_{ic} \leftarrow W_{ic} \frac{\left(V H^T\right)_{ic} - \alpha W_{ic}}{\left(W H H^T\right)_{ic} + \epsilon}$$

[4] Piper et. al., AMOS, 2004

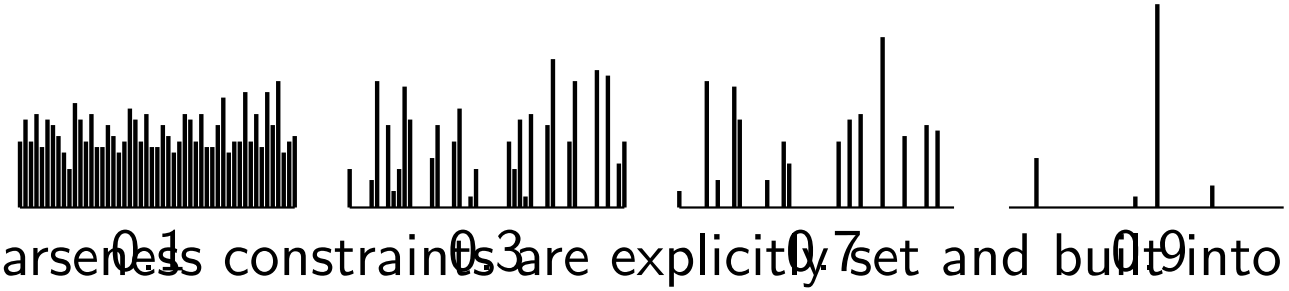
Sparsity

Hoyer defined sparsity as

$$sparseness\left(\vec{x}\right) = \frac{\sqrt{n} - \frac{\sum |x_i|}{\sqrt{\sum x_i^2}}}{\sqrt{n} - 1}$$

- Zero if and only if all components have same magnitude.
- One if and only if x contains one nonzero component.

Visual Interpretation



- Sparseness constraints are explicitly set and built into update algorithm.
- Can be applied to W, H, or both.
- Dominant features are (hopefully) preserved.

Sparsity Update Rules with MM

Pauca et. al. implemented sparsity within MM as

$$
\begin{aligned}
H_{cj} &\leftarrow H_{cj} \frac{\left(W^T V\right)_{cj} - \beta\left(c_1 H_{cj} + c_2 E_{cj}\right)}{\left(W^T W H\right)_{cj} + \epsilon} \\
c_1 &= \omega^2 - \omega \frac{\|\bar{H}\|_1}{2\|\bar{H}\|_2} \\
c_2 &= \|\bar{H}\| - \omega \|\bar{H}\|_2 \\
\omega &= \sqrt{kn} - \left(\sqrt{kn} - 1\right) \text{sparseness}(H)
\end{aligned}
$$

Benefits of NMF

- Automated cluster labeling (& clustering)
- Synonym generation (features)
- Possible automated ontology creation
- Labeling can be applied to *any* hierarchy

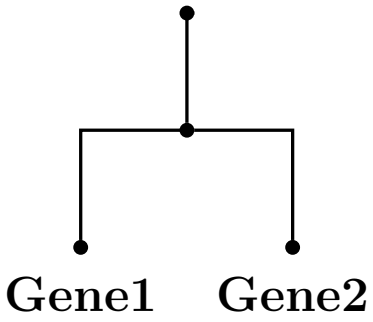
Comparison of SVD vs. NMF

	SVD	NMF
Solution Accuracy	A	B
Uniqueness	A	C
Convergence	A	C-
Querying	A	C+
Interpretability of Parameters	A	C
Interpretability of Elements	D	A
Sparseness	D-	B+
Storage	B-	A

Labeling Algorithm

Given a hierarchical tree and a weighted list of terms associated with each gene, assign labels to each internal node

- Mark each gene as labeled.
- For each pair of labeled sibling nodes
  - Add all terms to parent's list
  - Keep top t terms



inherited 1.05
genetic 1
disorder 0.95
tumor 0.95
Alzheimer 0.9
memory 0.8
cancer 0.8

Alzheimer 0.9	tumor 0.95
memory 0.8	cancer 0.8
disorder 0.6	inherited 0.6
genetic 0.5	genetic 0.5
inherited 0.45	disorder 0.35

Calculating Initial Term Weights

Three different methods to calculate initial term weights:

- Assign global weight associated with each term to each document.
- Calculate document-to-term similarity (LSI).
- For NMF, for each document j:
  - Determine top feature i (by examining H).
  - Assign dominant terms from feature vector i to document j, scaled by coefficient from H.
  - (Can be extended/thresholded to assign more features)

MeSH Labeling

- Many studies validate via inspection.
- For automated validation, need to generate a "correct" tree labeling.
  - Take advantage of expert opinions, i.e., indexers from MeSH.
  - Create MeSH meta-document for each gene, i.e., list of MeSH headings.
  - Label hierarchy using global weights of meta-collection.

Recall

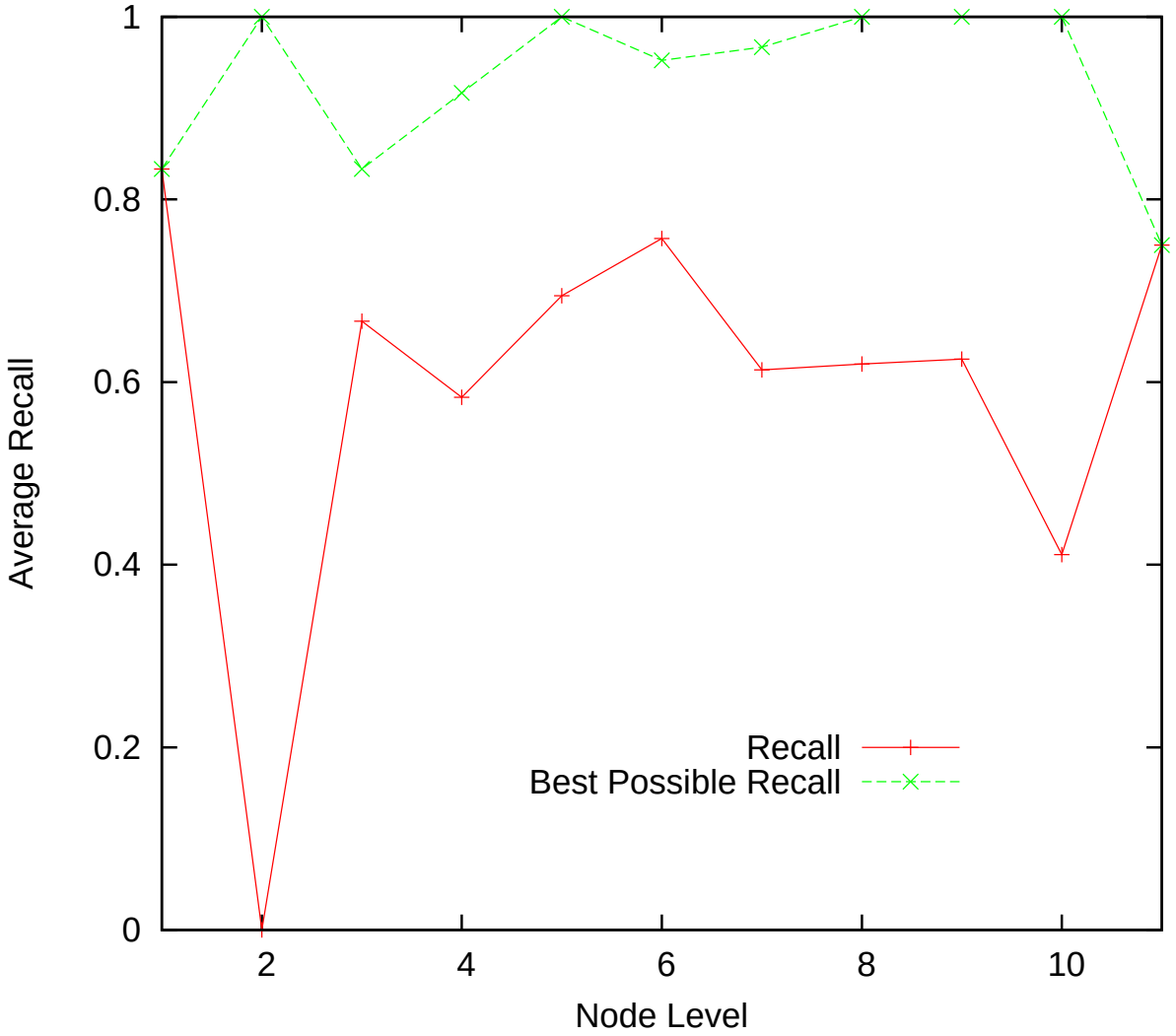
- From traditional IR, recall is ratio of relevant returned documents to all relevant documents.
- Extending this to trees, recall can be found at each node.
- Averaging across each tree depth level produces *average recall*.
- Averaging average recalls across all levels produces *mean average recall*.

Feature Vector Replacement

Unfortunately, MeSH vocabulary is too restrictive, so nearly all runs produced near 0% recall.

- Map NMF vocabulary to MeSH terminology.
- Foreach document i:
  - Determine j, the highest coefficient from ith column of H.
  - Choose top r MeSH headings from corresponding MeSH meta-document.
  - Split each MeSH heading into tokens.
  - Add each token to jth column of W', where weight is global MeSH header weight $\times$ coefficient from H.

Result: feature vector comprised solely of MeSH terms (MeSH feature vector).


1
0.8
0.6
0.4
0.2
0
Average Recall
2
4
6
8
10
Node Level
Recall
Best Possible Recall

Data Sets

Five collections were available:

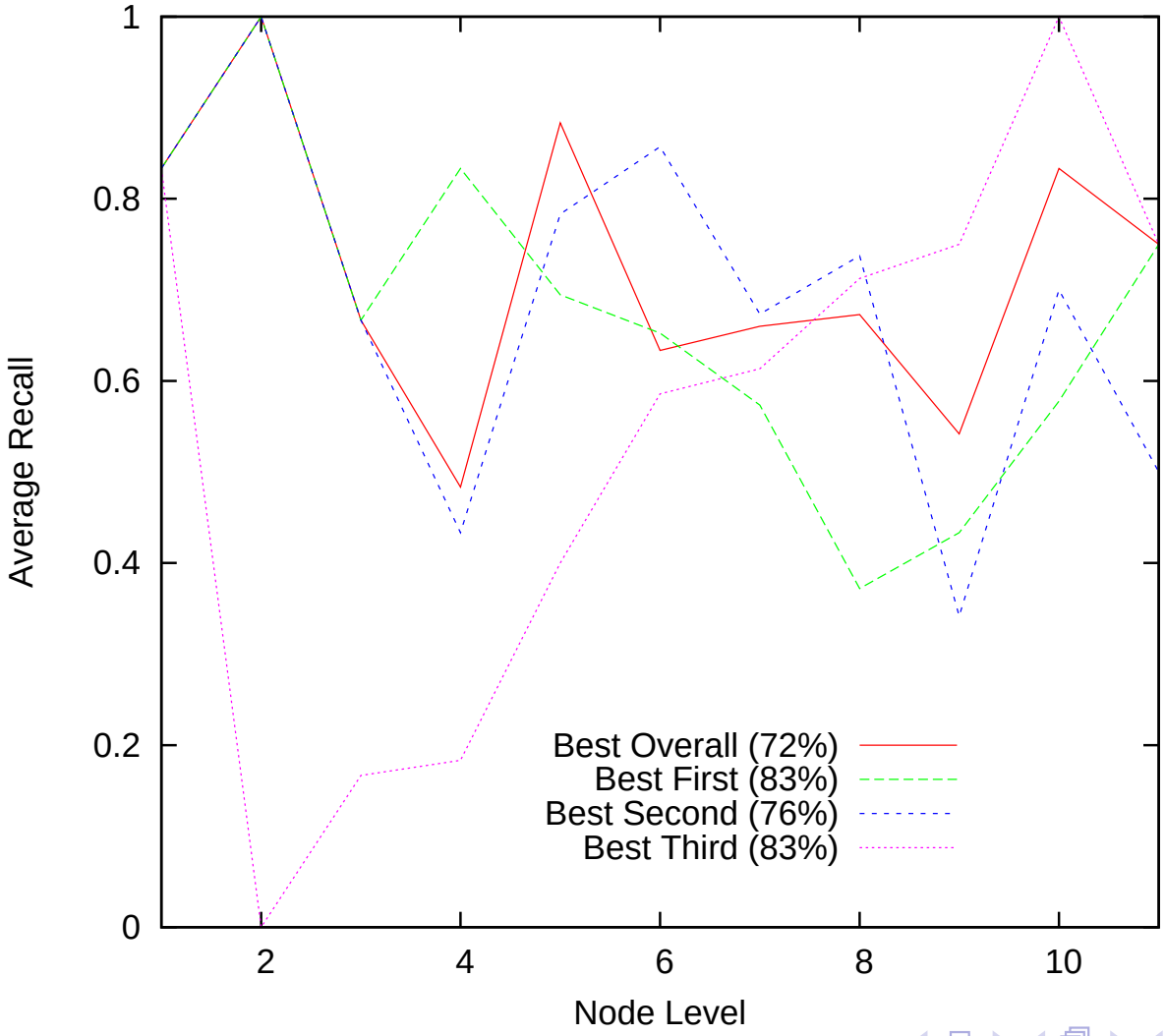
- 50TG, test set of 50 genes
- 115IFN, set of 115 interferon genes
- 3 cerebellar datasets, 40 genes of unknown relationship

Constraints

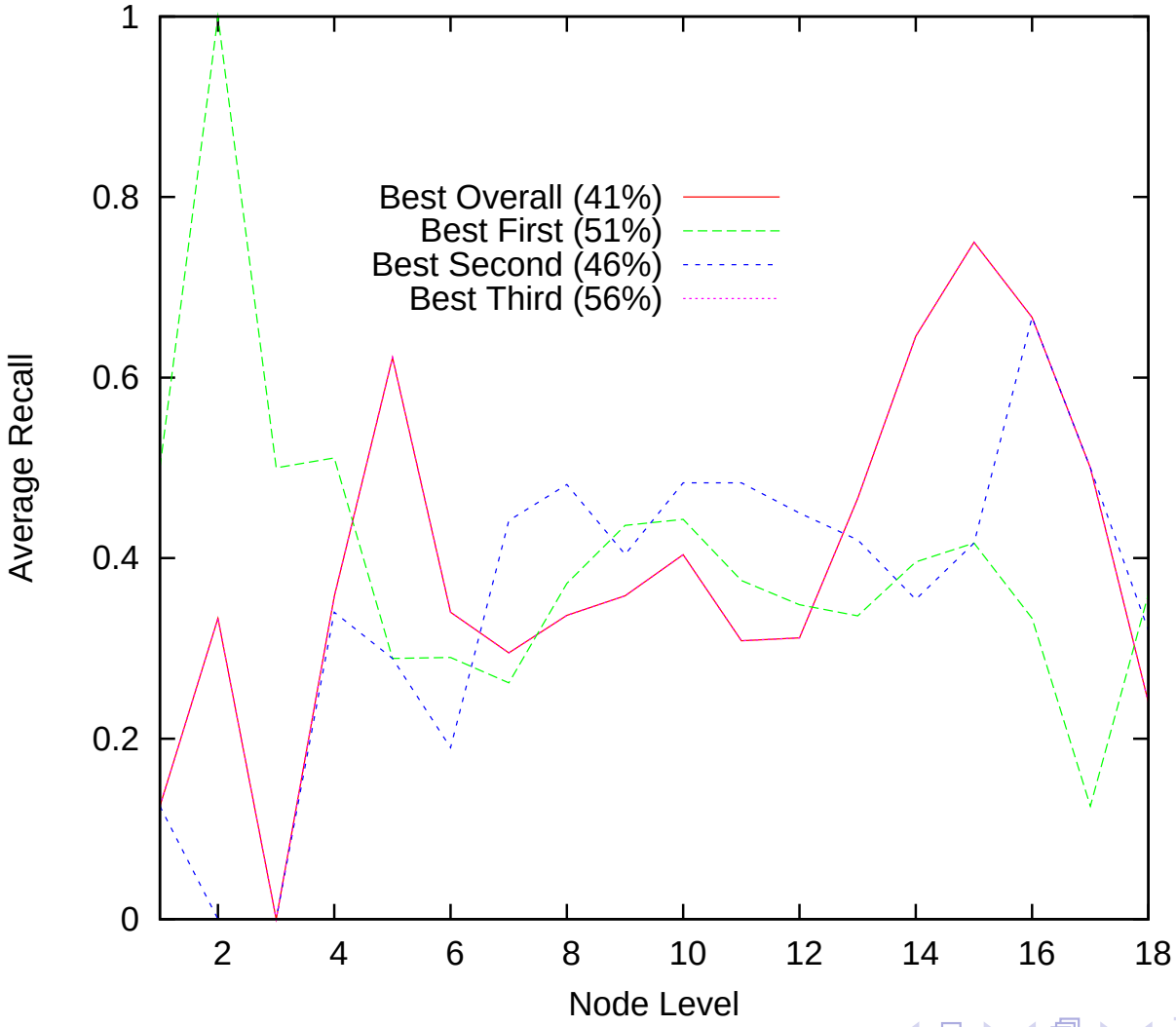
Each set was run under the given constraints for $k = 2, 4, 6, 8, 10, 15, 20, 25, 30$:

- no constraints
- smoothing W with $\alpha = 0.1, 0.01, 0.001$
- smoothing H with $\beta = 0.1, 0.01, 0.001$
- sparsifying W with $\alpha = 0.1, 0.01, 0.001$ and sparseness $= 0.1$, 0.25, 0.5, 0.75, 0.9
- sparsifying H with $\beta = 0.1, 0.01, 0.001$ and sparseness $= 0.1$, 0.25, 0.5, 0.75, 0.9

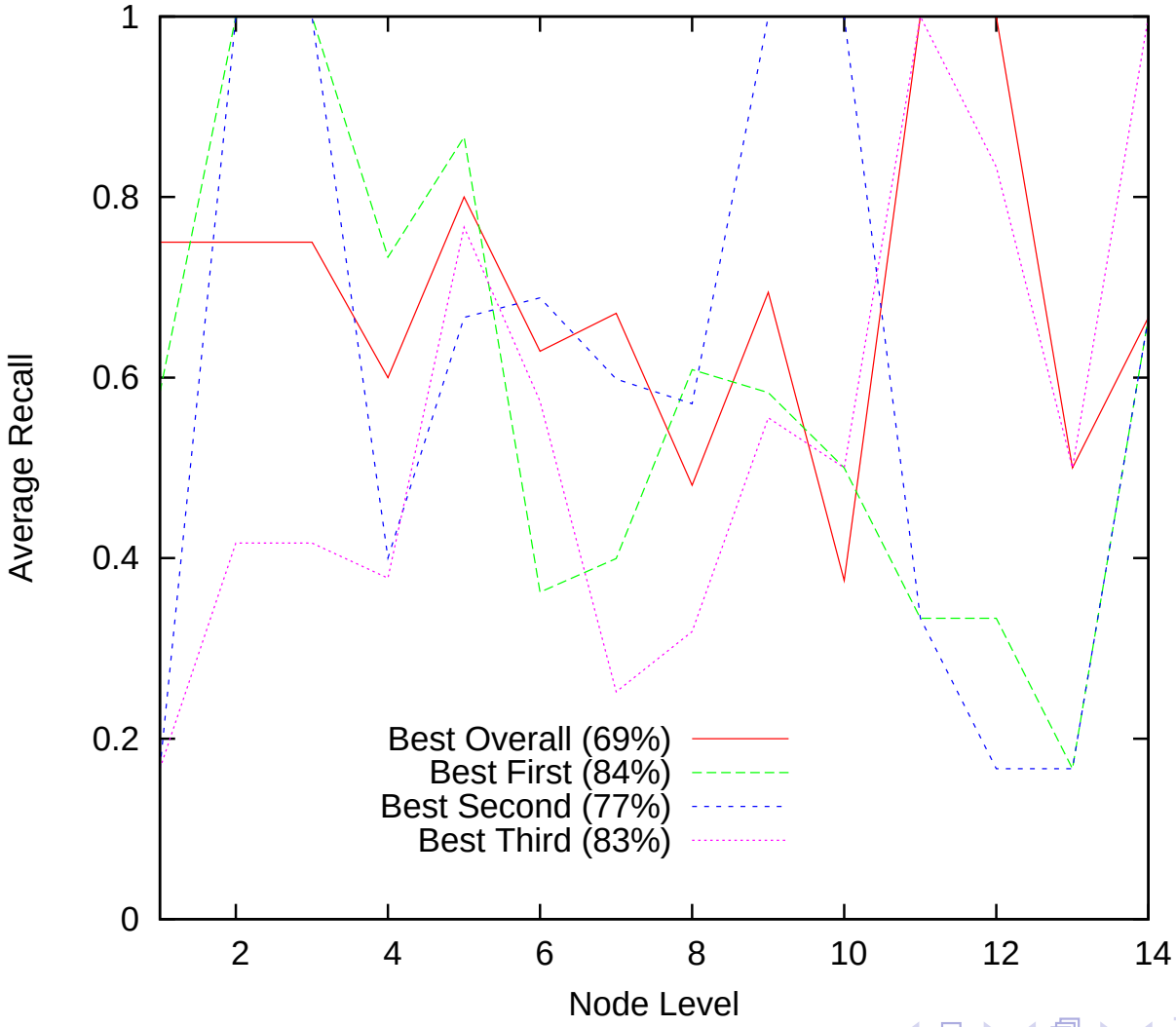
50TG Recall


1
0.8
0.6
0.4
0.2
0
Average Recall
2
4
6
8
10
Node Level
Best Overall (72%)
Best First (83%)
Best Second (76%)
Best Third (83%)

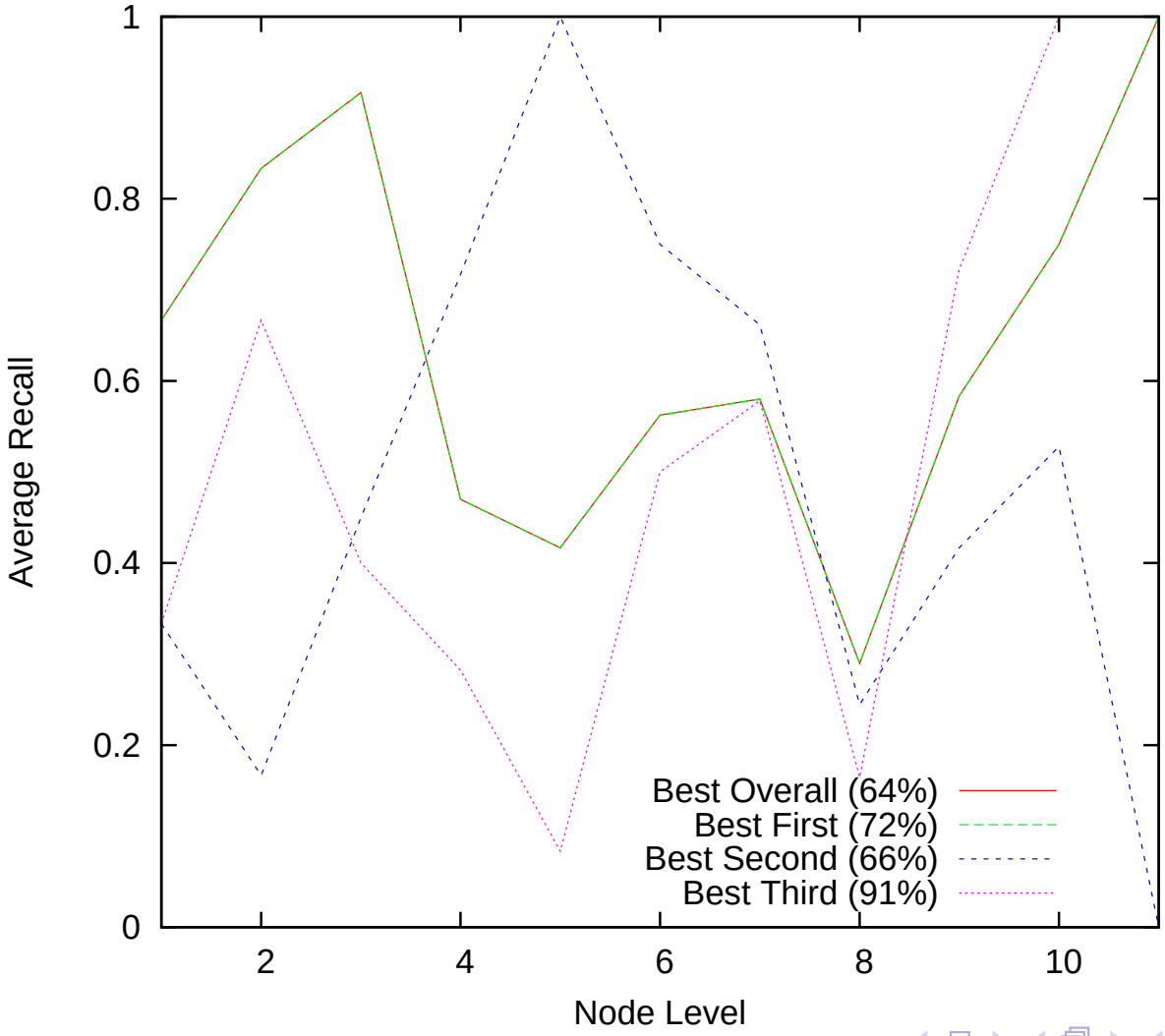
115IFN Recall



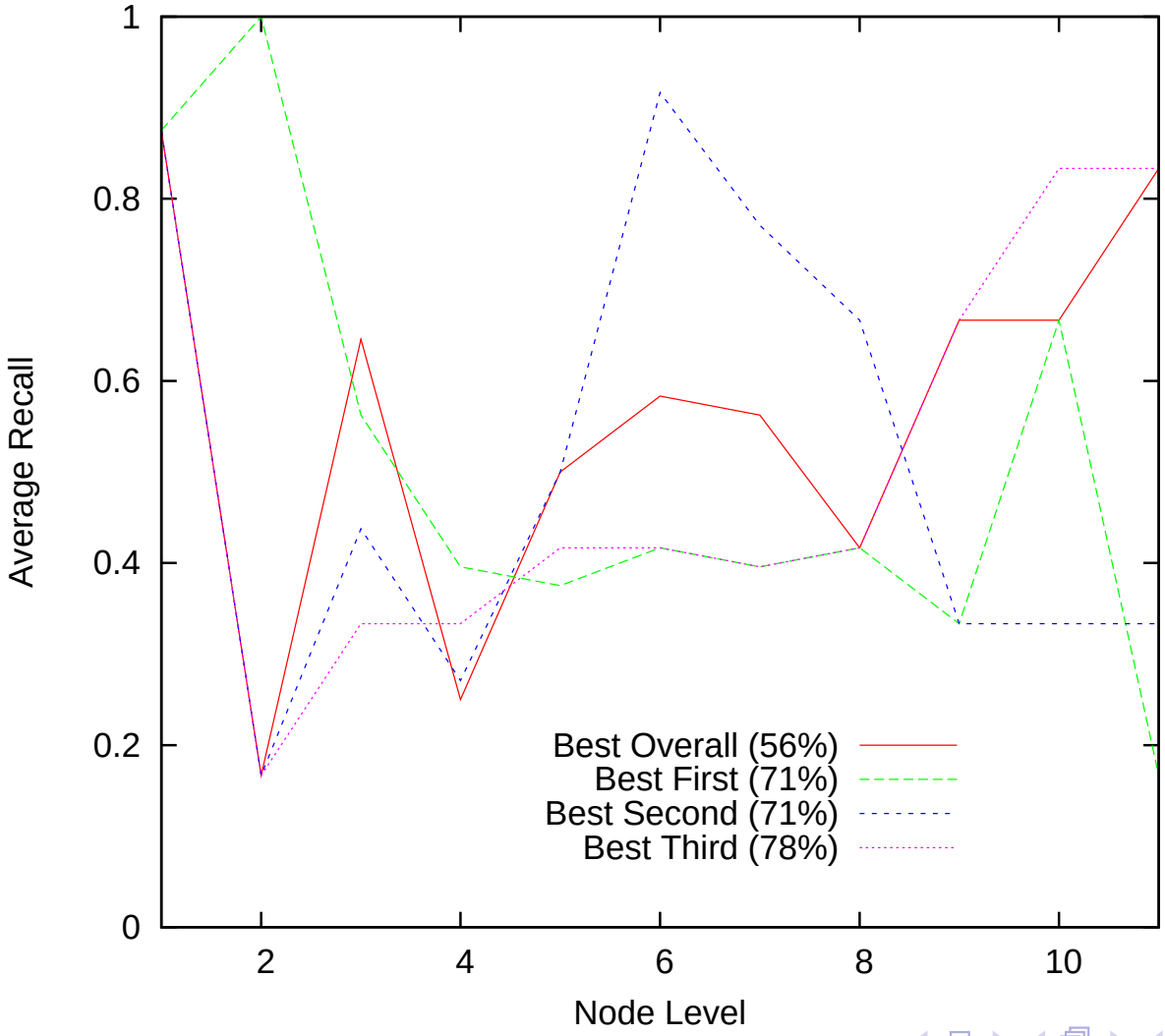
Math1 Recall



Mea Recall


1
0.8
0.6
0.4
0.2
0
Average Recall
2
4
6
8
10
Node Level
Best Overall (64%)
Best First (72%)
Best Second (66%)
Best Third (91%)

Sey Recall


1
0.8
0.6
0.4
0.2
0
Average Recall
2
4
6
8
10
Node Level
Best Overall (56%)
Best First (71%)
Best Second (71%)
Best Third (78%)

Observations

- Recall was more a function of initialization and choice of k than constraint.
- Often, the NMF run with the smallest approximation error did not produce the optimal labeling.
- Overall, sparsity did not perform well.
- Smoothing W achieved the best MAR in general.
- Small parameters choices and larger values of k performed better.

Future Work

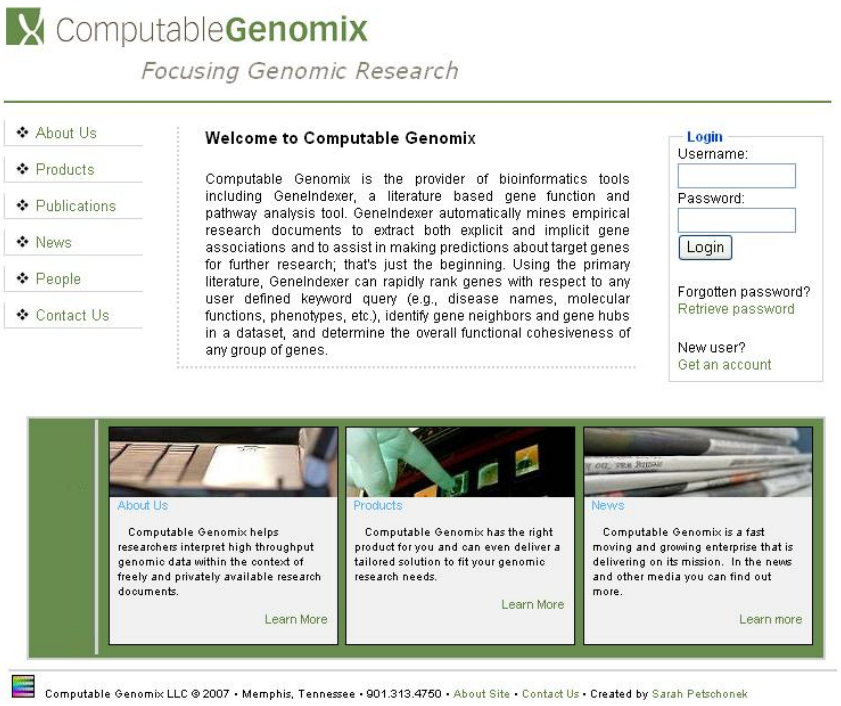
- Lots of NMF algorithms, each with different strengths/weaknesses.
- More complex labeling scheme.
- Different weighting schemes.
- Investigate hierarchical graphs.
- Visualization?
- Gold Standard?

SGO & CGx

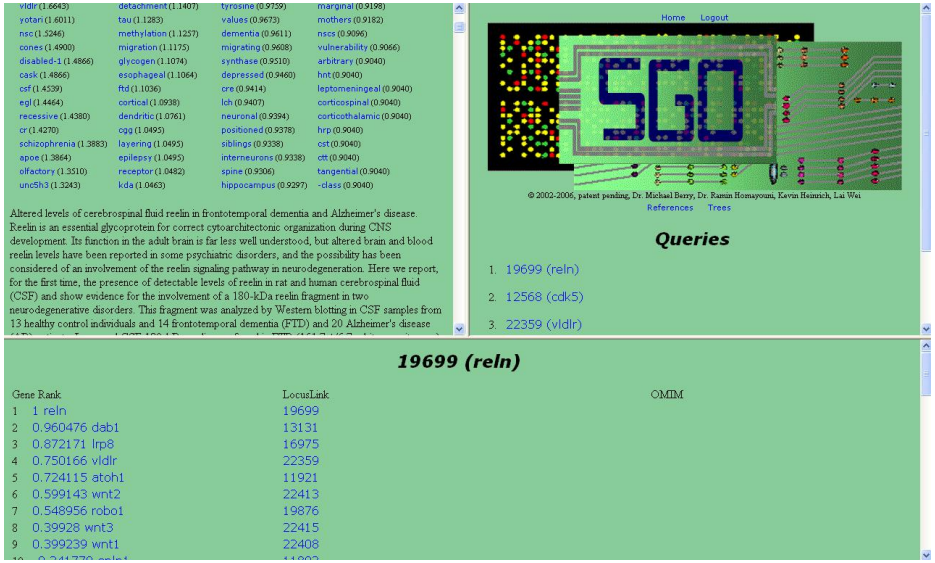
This tool as implemented is called the Semantic Gene Organizer (SGO).

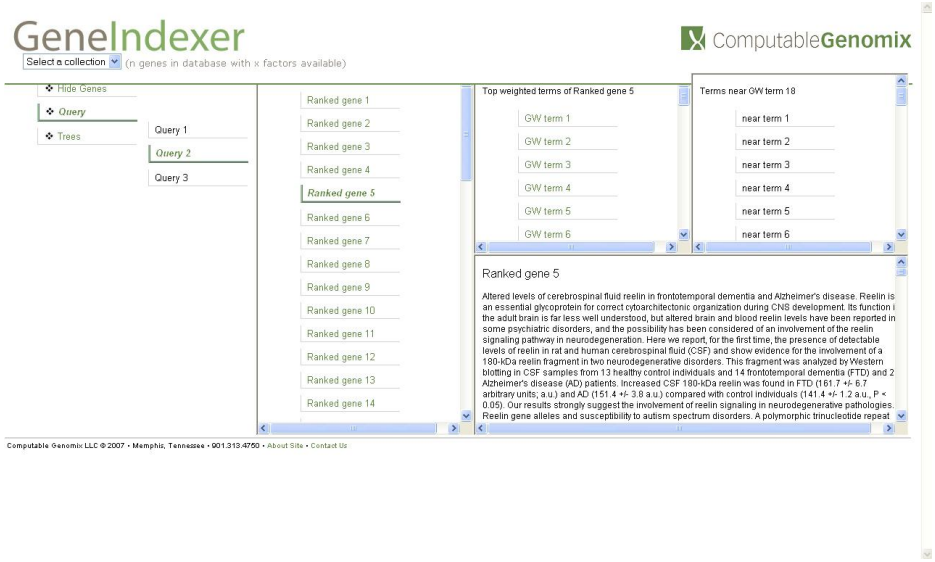
Much of the technology used in SGO are the basis for Computable Genomix, LLC.

SGO & CGx Screenshots



SGO & CGx Screenshots





Acknowledgments

SGO: shad.cs.utk.edu/sgo
CGx: computablegenomix.com

Committee:

- Michael W. Berry, chair
- Ramin Homayouni (Memphis)
- Jens Gregor
- Mike Thomason

Paúl Pauca, WFU