Latent Dirichlet Allocation

Independent Project Final Report

CIS 798

Instructor: Prof. William H. Hsu

MS Student: Svitlana Volkova

Table of Contents

Latent semantic analysis (LSA)

Latent semantic analysis (LSA) is a technique in natural language processing of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.

LSA can use a term-document matrix which describes the occurrences of terms in documents; it is a sparse matrix whose rows correspond to terms and whose columns correspond to documents.

LSA can intrinsically identify the relationship between words and their stem terms. This matrix is also common to standard semantic models. LSA transforms the occurrence matrix into a relation between the terms and some concepts, and a relation between those concepts and the documents. Thus the terms and documents are now indirectly related through the concepts.

Probabilistic latent semantic analysis (PLSA)

Probabilistic latent semantic analysis (PLSA) is a statistical technique for the analysis of two-mode and co-occurrence data. PLSA evolved from Latent semantic analysis (LSA), adding a sounder probabilistic model. Compared to standard latent semantic analysis which stems from linear algebra and downsizes the occurrence tables (usually via a SVD), probabilistic latent semantic analysis is based on a mixture decomposition derived from a latent class model.

Problem of pLSA:

- ✓ Incomplete: Provide no probabilistic model at the level of documents
- ✓ The number of parameters in the model grows linear with the size of the corpus
- ✓ It is not clear how to assign probability to a document outside of the training data

Latent Dirichlet Allocation (LDA) & pLSA

The pLSA model posits that each word of a *training* document comes from a randomly chosen topic. The topics are themselves drawn from a document-specific distribution over topics, i.e., a point on the topic simplex. There is one such distribution for each document; the set of training documents thus defines an empirical distribution on the topic simplex.

LDA posits that each word of both the observed and unseen documents is generated by a randomly chosen topic which is drawn from a distribution with a randomly chosen parameter. This parameter is sampled once per document from a smooth distribution on the topic simplex.

Transition from LSA/LSI to pLSA/pLSI and to LDA

Latent Dirichlet Allocation was first proposed by David Blei, Andrew Ng, and Michael Jordan in 2002. The commonly cited paper [Ble03] reintroduced the model as a graphical model. As written in their article, this is how the field developed:

- tf-idf Salton and McGill [Sal00]
- Latent Semantic Indexing (LSI) Deerwester et al. [Dee90]
- Probabilistic Latent Semantic Indexing (pLSI) Hofmann [Hof99]

When Deerwester et al. introduced LSA/LSI they were concerned with dealing with *polysemy* (words with multiple meanings) and *synonymy* (multiple words with the same meaning). They proposed a technique based upon matrix decomposition. They called the dimensions "artificial concepts".

When Hofmann introduced pLSI he was concerned with using probability. Although, LSA has been applied with remarkable success in different domains, including automatic indexing. It has a number of deficits, mainly due to its unsatisfactory statistical foundation. Hofmann introduced a probability based model. He called the dimensions in his model "aspects" [Hof].

When Blei et al. introduced LDA they seemed concerned with exchangeability. In Hofmann's model, the order of the documents mattered. Blei et al. removed this dependency from the model. The dimensions are now called "topics".

We can make some inferences as to how optimistic the inventors of a model about how well they captured the semantic meaning of documents were based upon the name they gave to the dimensions that their model found. For LSI they were called "artificial concepts". Then for pLSI they were called "aspects".

This would seem to be a step down in optimism. Then in LDA they are called "topics:. This is a major step up in optimism and ambition. It is not clear how much more optimistic and ambitious one could be.

LDA Application

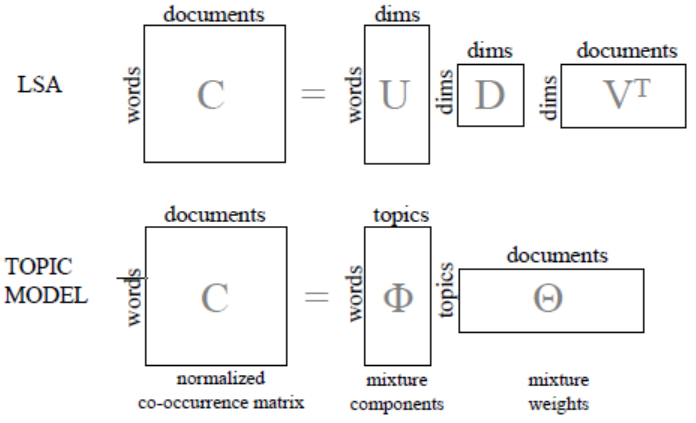
Wei and Croft [Wei07] and Chemudugunta, Smyth, and Steyvers [Che06] have successfully applied the LDA model to information retrieval and shown that it can significantly outperform – in terms of precision-recall – alternative methods such as latent semantic analysis.

LDA models have also been increasingly applied to problems involving very large text corpora: Mimno and McCallum [Mim07] and Newman et al [New07] have all used the LDA model to automatically generate topic models for millions of documents and used these models as the basis for automated indexing and faceted Web browsing.

Topic Models Representation

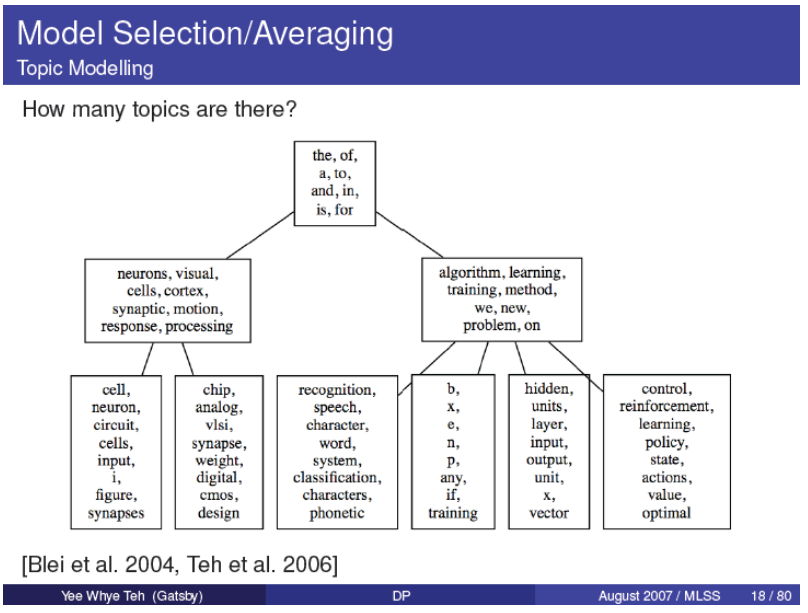
Topic models (e.g., Blei, Ng, & Jordan, 2003; Griffiths & Steyvers, 2002; 2003; 2004; Hofmann, 1999; 2001) are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words.

A topic model is a *generative model* for documents: it specifies a simple probabilistic procedure by which documents can be generated. To make a new document, one chooses a distribution over topics. Then, for each word in that document, one chooses a topic at random according to this distribution, and draws a word from that topic. Standard statistical techniques can be used to invert this process, inferring the set of topics that were responsible for generating a collection of documents.



The matrix factorization of the LSA model compared to topic model

The topics modeling visual representation can be taken from Yee Whye The talk that is based on [Ble06] paper.



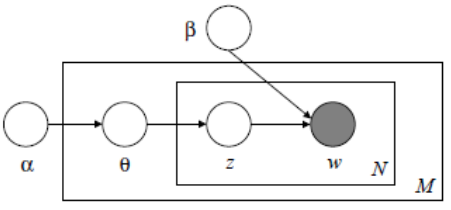
Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) assigns a discrete latent model to words and let each document maintain a random variable, indicating its probabilities of belonging to each topic

LDA has mainly been used to model text corpora, where the notion of exchangeability corresponds to the "bag-of-words" assumption that is commonly employed in such models.

LDA models each document as a mixture over topics, where each vector of mixture proportions is assumed to have been drawn from a Dirichlet distribution. A topic in this model is defined to be a discrete distribution over words from some finite lexicon. For example, if a topic is "astrophysics", then the word "quasar" would presumably be assigned a higher probability than the word "burrito".

LDA Model [Ble03]



The random variables:

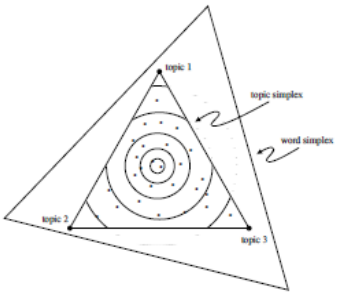
- *word* is represented as a multinomial random variable *w*;
- *topic* is represented as a multinomial random variable z;
- *document* is represented as Dirichlet random variable θ

Prior:

- α - Dirichlet prior over the documents;

Plates:

- repeated sampling of the Dirichlet document variable within the corpus;
- repeated sampling of the multimodal topic variable within documents.



The topic simplex for $k = 3$.

Let's briefly consider the graphical representation of LDA via topics simplex:

- each cornet of the simplex corresponds to a topic – a component of the vector *z*;
- a document is modeled as a point of the simplex - a multimodal distribution over the topics;
- a corpus is modeled as a Dirichlet distribution on the simplex.

LDA model in Matlab

The input is a bag of word representation containing the number of times each words occurs in a document. The outputs are the topic assignments to each word token as well as the counts of the number of times each word is assigned to a topic and the number of times a topic is assigned to a document

INPUTS

- $|WS|$ - $1 * |N|$ vector where $|WS(k)|$ contains the vocabulary index of the k^{th} word token, and $|N|$ is the number of word tokens. The word indices are not zero based, i.e., $min(|WS|) = 1$ and $max(|WS|) = |W| = number\ of\ distinct\ words\ in\ vocabulary$;
- $|DS|$ a $1 * |N|$ vector where $|DS(k)|$ contains the document index of the k^{th} word token. The document indices are not zero based, i.e., $min(|DS|) = 1$ and $max(|DS|) = |D| = number\ of\ documents$;
- |WO| a $1 * |W|$ cell array of strings where $|WO\{k\}|$ contains the k^{th} vocabulary item and $|W|$ is the number of distinct vocabulary items. Not needed for running the Gibbs sampler but becomes necessary when writing the resulting word-topic distributions to a file using the |writetopics| Matlab function.

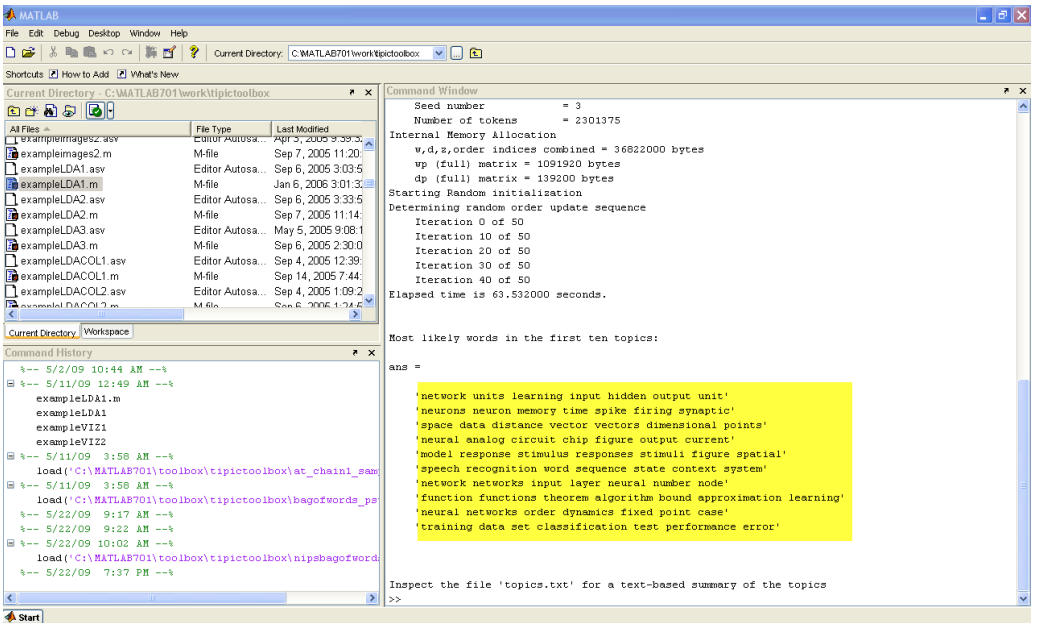
OUTPUT

- $|WP|$ a sparse matrix of size $|W|\ x\ |T|$, where $|W|$ is the number of words in the vocabulary and $|T|$ is the number of topics. $|WP(i,j)|$ contains the number of times word $|i|$ has been assigned to topic $|j|$.
- $|DP|$ a sparse $|D|\ x\ |T|$ matrix, where $|D|$ is the number of documents. $|DP(d,j)|$ contains the number of times a word token in document $|d|$ has been assigned to topic $|j|$.
- $|Z|$ a $1 * |N|$ vector containing the topic assignments where $|N|$ is te number of word tokens. $|Z(k)|$ contains the topic assignment for token $|k|$.

Topic Modeling using LDA

```
This example shows how to run the LDA Gibbs sampler on a small dataset to
% extract a set of topics and shows the most likely words per topic.

% Choose the dataset
dataset = 2; % 1 = psych review abstracts 2 = NIPS papers

if (dataset == 1)
   % load the psych review data in bag of words format
   load 'bagofwords_psychreview';
   % Load the psych review vocabulary
   load 'words_psychreview';
elseif (dataset == 2)
   % load the nips dataset
   load 'bagofwords_nips';
   % load the nips vocabulary
   load 'words_nips';
end

%%
% Set the number of topics
T=50;

%%
% Set the hyperparameters
BETA=0.01;
ALPHA=50/T;

%%
% The number of iterations
N = 100;

%%
% The random seed
SEED = 3;

%%
% What output to show (0=no output; 1=iterations; 2=all output)
OUTPUT = 1;

%%
% This function might need a few minutes to finish
tic
[ WP,DP,Z ] = GibbsSamplerLDA( WS , DS , T , N , ALPHA , BETA , SEED , OUTPUT );
toc

%%
% Just in case, save the resulting information from this sample
if (dataset==1)
```

```
   save 'ldasingle_psychreview' WP DP Z ALPHA BETA SEED N;
end

if (dataset==2)
   save 'ldasingle_nips' WP DP Z ALPHA BETA SEED N;
end
%%
% Put the most 7 likely words per topic in cell structure S
[S] = WriteTopics( WP , BETA , WO , 7 , 0.7 );

fprintf( '\n\nMost likely words in the first ten topics:\n' );

%%
% Show the most likely words in the first ten topics
S( 1:10 )

%%
% Write the topics to a text file
WriteTopics( WP , BETA , WO , 10 , 0.7 , 4 , 'topics.txt' );

fprintf( '\n\nInspect the file ''topics.txt'' for a text-based summary of the topics\n' );
```



Function GibbsSamplerLDA

% Runs the Gibbs sampler for the Latent Dirichlet Allocation (LDA) model

%

% |[WP,DP,Z] = GibbsSamplerLDA(WS,DS,T,N,ALPHA,BETA,SEED,OUTPUT)| will

% run the Gibbs sampler for the LDA model on a bag of words data provided by the vectors |WS| and |DS|. |WS(k)| and |DS(k)| contain the word and document indices for the kth token. The maximum of |WS| is |W|, the vocabulary size. The maximum of |DS| is |D|, the number of documents. |T| is the number of topics. The first output is the sparse matrix |WP|, of size |W| x |T|, where |WP(i,j)| contains the number of times word |i| has been assigned to topic |j|. The second output is |DP|, a sparse |D| x |T| matrix, where |DP(i,j)| contains the number of times a word in document |d| has been assigned to topic |j|. The third output |Z| contains the topic assignments; |Z(k)| contains the topic assignment for token k.

%

% |[WP,DP,Z] = GibbsSamplerLDA(WS,DS,T,N,ALPHA,BETA,SEED,OUTPUT,ZIN)|

will run the sampler from starting state |ZIN|, where |ZIN(k)| contains the topic assignment for token k, saved from a previous sample.

%

% |WS| and |DS| should be in double precision |N| determines the number of iterations to run the Gibbs sampler.

% |ALPHA| and |BETA| are the hyper parameters on the Dirichlet priors for the topic distributions (|theta|) and the topic-word distributions (|phi|)respectively

%

% |SEED| sets the seed for the random number generator

%

% |OUTPUT| determines the screen output by the sampler

% 0 = no output provided

% 1 = show the iteration number only

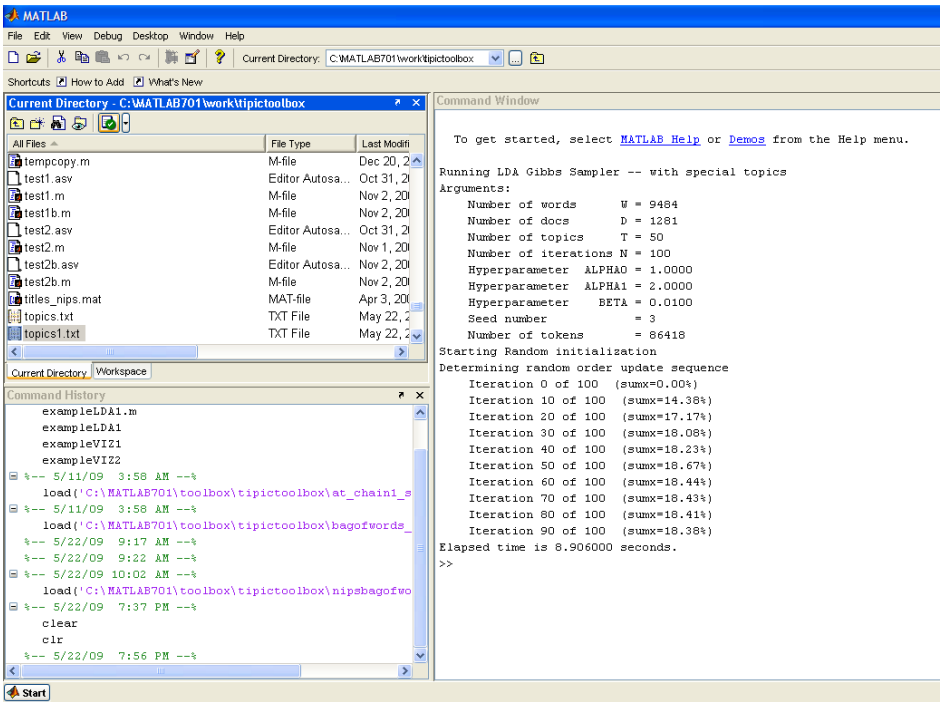
% 2 = show all output

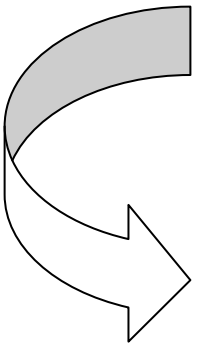
% A good setting for the number of iterations will depend on the number of topics and the complexity of problem. For most problems, 500 to 2000 iterations will suffice.

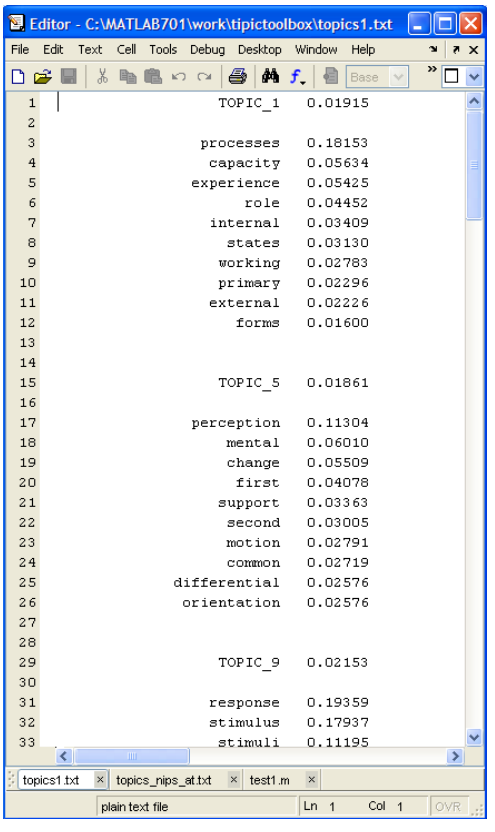
%

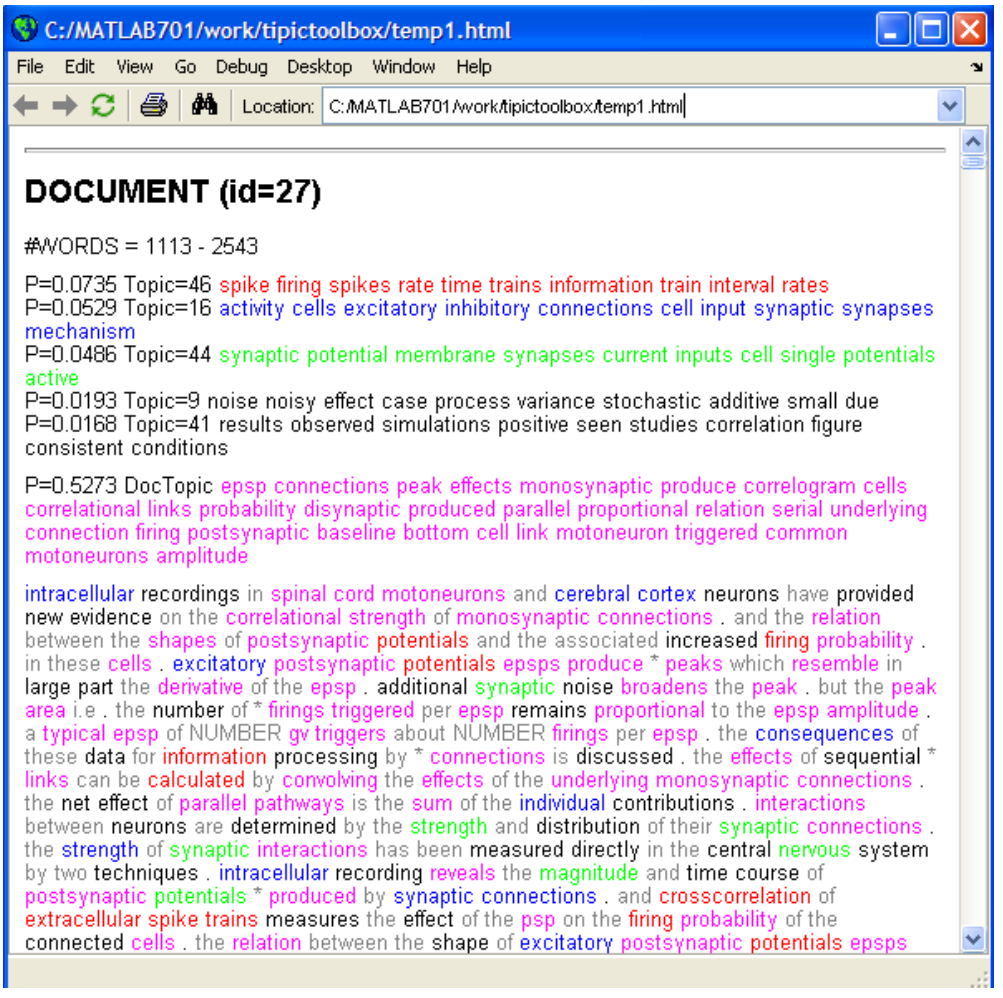
% Appropriate values for |ALPHA| and |BETA| depend on the number of topics and the number of words in vocabulary. For most applications, good results can be obtained by setting |ALPHA = 50 / T| and |BETA = 200 / W|

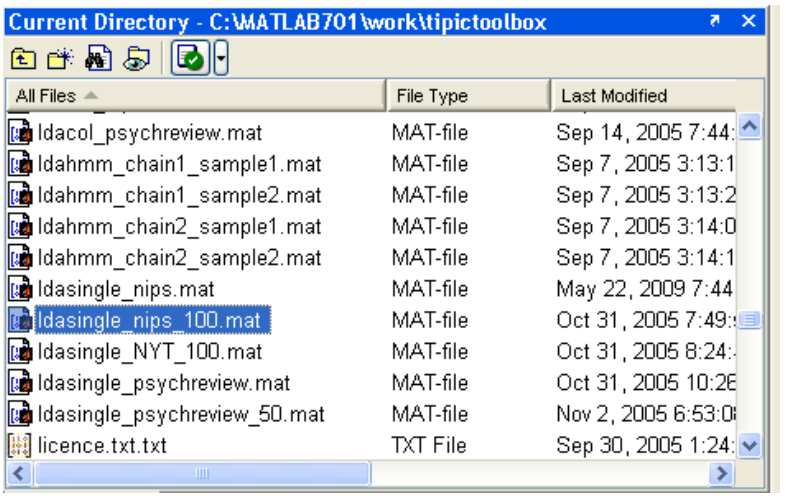
Running test on LDA & topics generation








C:/MATLAB701/work/tipictoolbox/temp1.html
File Edit View Go Debug Desktop Window Help
Location: C:/MATLAB701/work/tipictoolbox/temp1.html
DOCUMENT (id=27)
#WORDS = 1113 - 2543
P=0.0735 Topic=46 spike firing spikes rate time trains information train interval rates
P=0.0529 Topic=16 activity cells excitatory inhibitory connections cell input synaptic synapses mechanism
P=0.0486 Topic=44 synaptic potential membrane synapses current inputs cell single potentials active
P=0.0193 Topic=9 noise noisy effect case process variance stochastic additive small due
P=0.0168 Topic=41 results observed simulations positive seen studies correlation figure consistent conditions
P=0.5273 DocTopic epsp connections peak effects monosynaptic produce correlogram cells correlational links probability disynaptic produced parallel proportional relation serial underlying connection firing postsynaptic baseline bottom cell link motoneuron triggered common motoneurons amplitude
intracellular recordings in spinal cord motoneurons and cerebral cortex neurons have provided new evidence on the correlational strength of monosynaptic connections . and the relation between the shapes of postsynaptic potentials and the associated increased firing probability . in these cells . excitatory postsynaptic potentials epsps produce * peaks which resemble in large part the derivative of the epsp . additional synaptic noise broadens the peak . but the peak area i.e . the number of * firings triggered per epsp remains proportional to the epsp amplitude . a typical epsp of NUMBER gv triggers about NUMBER firings per epsp . the consequences of these data for information processing by * connections is discussed . the effects of sequential * links can be calculated by convolving the effects of the underlying monosynaptic connections . the net effect of parallel pathways is the sum of the individual contributions . interactions between neurons are determined by the strength and distribution of their synaptic connections . the strength of synaptic interactions has been measured directly in the central nervous system by two techniques . intracellular recording reveals the magnitude and time course of postsynaptic potentials * produced by synaptic connections . and crosscorrelation of extracellular spike trains measures the effect of the psp on the firing probability of the connected cells . the relation between the shape of excitatory postsynaptic potentials epsps


Current Directory - C:\MATLAB701\work\tipictoolbox
All Files
File Type
Last Modified
ldacol_psychreview.mat MAT-file Sep 14, 2005 7:44:
ldahmm_chain1_sample1.mat MAT-file Sep 7, 2005 3:13:1
ldahmm_chain1_sample2.mat MAT-file Sep 7, 2005 3:13:2
ldahmm_chain2_sample1.mat MAT-file Sep 7, 2005 3:14:0
ldahmm_chain2_sample2.mat MAT-file Sep 7, 2005 3:14:1
ldasingle_nips.mat MAT-file May 22, 2009 7:44
ldasingle_nips_100.mat MAT-file Oct 31, 2005 7:49:
ldasingle_NYT_100.mat MAT-file Oct 31, 2005 8:24:
ldasingle_psychreview.mat MAT-file Oct 31, 2005 10:2
ldasingle_psychreview_50.mat MAT-file Nov 2, 2005 6:53:0
licence.txt.txt TXT File Sep 30, 2005 1:24:

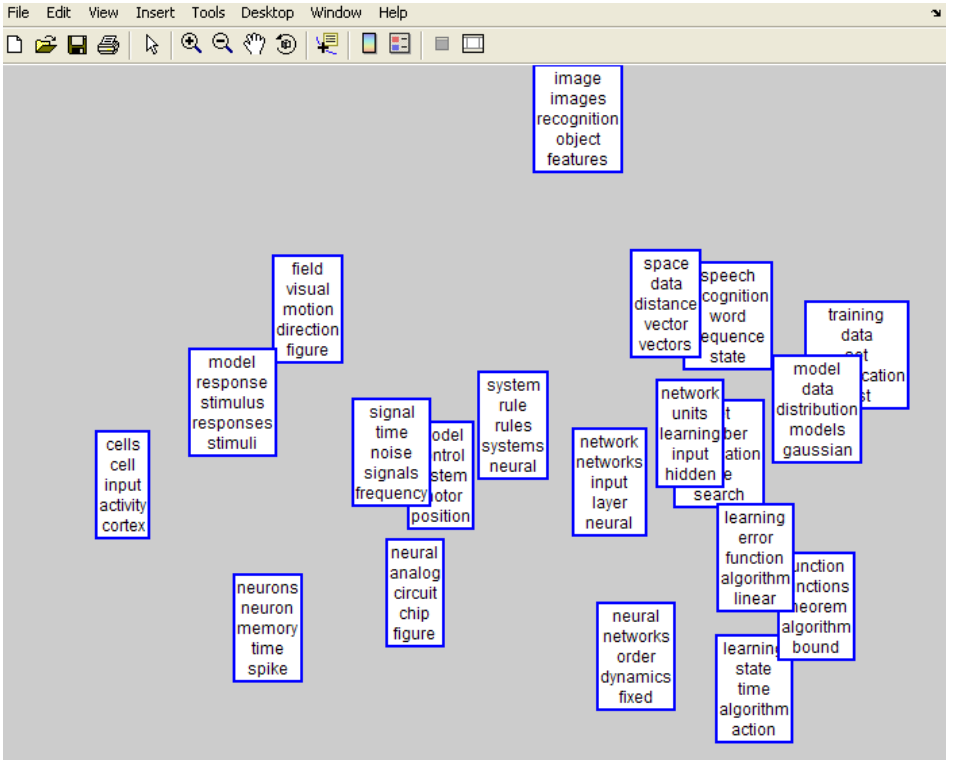
Visualization script

Editor - C:\MATLAB701\work\lda\exampleVIZ1.m

File Edit Text Desktop Window Help

```
1  %% Example 1 to visualize topic model results
2  % This example shows how to visualize topics in a 2D map.
3  %
4  % load a document-topic count matrix saved for the nips dataset
5  load 'ldasingle_nips';
6  load 'words_nips';
7
8  %%
9  % extract the topics in a cell array of strings
10 [S]=WriteTopics( WP,BETA,WO,5,0.6 );
11
12 fprintf( 'Please wait while calculating visualization...\n' );
13 drawnow;
14
15 %%
16 % visualize these topics in a 2D map. Have each topic by displayed
17 % vertically
18 VisualizeTopics( DP,ALPHA,S,'vertical');
```

Document's visualization



References

[Ble03] Blei, D.M., Ng, A.Y., Jordan, M.I. (2003). Latent Dirichlet allocation, Journal of Machine Learning Research, 3, pp.993-1022

[Che06] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In Neural Information Processing Systems 19. MIT Press, 2006.

[Dee90] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. Journal of the American Society of Information Science, 41:993{1022, 1990}.

[Hof99] T. Hofmann. Probabilistic latent semantic indexing. Proceedings of the Twenty-Second Annual International SIGIR Conference, 1999.

[Ish01] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. Journal of the American Statistical Association, 96(453):161{173, 2001}.

[Mim07] D. Mimno and A. McCallum. Organizing the OCA: learning faceted subjects from a library of digital books. In JCDL '07: Proceedings of the 2007 conference on Digital libraries, pages 376–385, New York, NY, USA, 2007. ACM

[Nea00] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics, 9:249{265, 2000}.

[New07] D. Newman, K. Hagedorn, C. Chemudugunta, and P. Smyth. Subject metadata enrichment using statistical topic models. In JCDL '07: Proceedings of the 2007 conference on Digital libraries, pages 366–375, New York, NY, USA, 2007. ACM.

[Ram07] R. V. Ramamoorthi and K. R. Srikanth. Dirichlet processes. In Encyclopedia of Statistical Sciences. John Wiley and Sons, New York, 2007.

[Sal83] Gerard Salton and Michael J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA, 1983.

[Set94] J. Sethuraman. A constructive definition of Dirichlet priors. Statistica Sinica, 4:639{650, 1994}.

[Yu05] Yu, K., Yu, S., Volker Tresp, V. (2005). Dirichlet enhanced latent semantic analysis. In Workshop on Artificial Intelligence and Statistics AISTAT 2005.

[Wei06] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In SIGIR, pages 178–185, 2006.

LDA implementations:

- http://www.ics.uci.edu/iporteou/fastlda
- http://www.cs.princeton.edu/~blei/lda-c/
- http://chasen.org/~daiti-m/dist/lda/
- http://gibbslda.sourceforge.net/
- http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm#Matlab_Datasets

Data sets at the UCI Machine Learning Repository

- http://archive.ics.uci.edu/ml/machine-learningdatabases/bag-of-words/

Video lectures on: video.google.com.

- http://video.google.com/videoplay?docid=3077213787166426672
- http://video.google.com/videoplay?docid=-8568727794989317846

Tutorial on Gibbs Sampling

- http://web.mit.edu/wingated/www/introductions/mcmc-gibbsintro.pdf