

# Practical Algorithm for Topic Modeling

Ke Ye

Viswanath Kammula

# Introduction

Topic Modeling is a popular method that learns thematic structure from large document collections without human supervision.. As part of this model we will be using the maximum likelihood objective to derive the distribution of topics on a given set of vocabulary.

**Input** - documents that are mixtures of topics, which will be modeled as distributions over vocabulary

**Output**- A set of word and their probability to depict a topic distribution



# Dataset

## **NIPS** dataset:

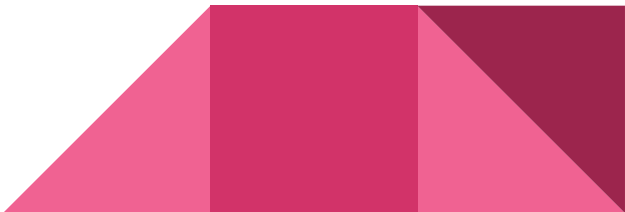
orig source: books.nips.cc

Link - <http://archive.ics.uci.edu/ml/machine-learning-databases/bag-of-words/>

- Number of Documents=1500
- Size of Vocabulary=12419
- Number of Words=1,900,000 (approx)

## **NY Times** dataset:

orig source: ldc.upenn.edu

- Number of Documents=300000
  - Size of Vocabulary=102626
  - Number of Words=100,000,000 (approx)
- 

# Pipeline of Project

1. Splitting data(held-out set), Converting the word-document dataset into a sparse matrix
2. Removing the stopwords and rare words
3. Generating the word2word co-occurrence matrix - Q and normalizing it
4. Using random projection to reduce the dimensionality of Q matrix
5. Finding the Anchor words
6. Recovering the topic distribution using the anchor words
7. Calculated Coherence for evaluation of the model



# Baseline - LDA

As a baseline algorithm we have executed Latent Dirichlet Allocation(LDA).

LDA is an unsupervised learning algorithm which will try to group all the similar contexts under a single topic. It is a generative probabilistic model for collections of discrete data such as text corpora.

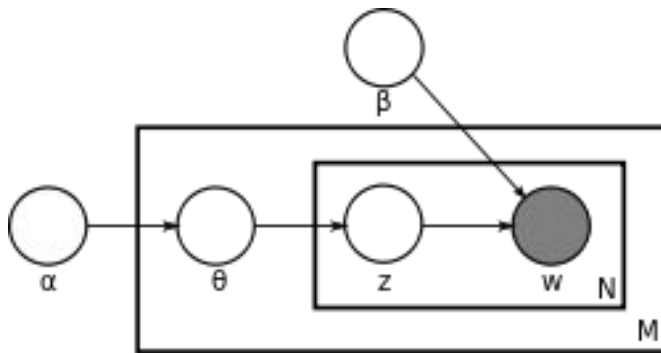
In terms of textual data, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics.



# LDA

LDA assumes the following generative process for each document  $w$  in a corpus  $D$ :

- Choose  $N \sim \text{Poisson}(\xi)$
- Choose  $\theta \sim \text{Dir}(\alpha)$
- For each of the  $N$  words we choose a topic  $T_n \sim \text{Multinomial}(\theta)$ .
- Then choose a word from  $p(\xi_n | T_n, \beta)$ , a multinomial probability conditioned on topic  $T_n$ .



# Anchor Words

We assume every topic is separable and contains at least one anchor word that has non-zero probability only in that topic. If a document contains this anchor word, then it is guaranteed that the corresponding topic is among the set of topics used to generate the document.

## Steps

1. Project the Q-matrix to randomly chosen subspace
2. Find the farthest point from origin as first anchor
3. For  $i=1$  to  $i=k-1$ , find the points that has the largest distance to  $\text{span}(S)$ , which is the largest norm of the projection of  $x$  onto the orthogonal complement of  $\text{span}(S)$



# Recovery methods

- KL Recovery
- L2 Recovery

---

**Algorithm 3.** RecoverKL

---

**Input:** Matrix  $Q$ , Set of anchor words  $\mathbf{S}$ , tolerance parameter  $\epsilon$ .

**Output:** Matrices  $A, R$

Normalize the rows of  $Q$  to form  $\bar{Q}$ .

Store the normalization constants  $\vec{p}_w = Q\vec{1}$ .

$\bar{Q}_{s_k}$  is the row of  $\bar{Q}$  for the  $k^{th}$  anchor word.

**for**  $i = 1, \dots, V$  **do**

    Solve  $C_i = \operatorname{argmin}_{C_i} D_{KL}(\bar{Q}_i || \sum_{k \in \mathbf{S}} C_{i,k} \bar{Q}_{s_k})$

    Subject to:  $\sum_k C_{i,k} = 1$  and  $C_{i,k} \geq 0$

    With tolerance:  $\epsilon$

**end for**

$A' = \operatorname{diag}(\vec{p}_w)C$

Normalize the columns of  $A'$  to form  $A$ .

$R = A^\dagger Q A^{\dagger T}$

**return**  $A, R$

---

## Original Recovery

$$Q = ARA^T = \begin{pmatrix} D \\ U \end{pmatrix} R \begin{pmatrix} D & U^T \end{pmatrix} = \begin{pmatrix} DRD & DRU^T \\ URD & URU^T \end{pmatrix}$$

---

**Algorithm 2.** Original Recover ([Arora et al., 2012b](#))

---

**Input:** Matrix  $Q$ , Set of anchor words  $\mathbf{S}$

**Output:** Matrices  $A, R$

Permute rows and columns of  $Q$

Compute  $\vec{p}_S = Q_S \vec{1}$  (equals  $DR\vec{1}$ )

Solve for  $\vec{z}$ :  $Q_{S,S} \vec{z} = \vec{p}_S$  (Diag( $\vec{z}$ ) equals  $D^{-1}$ )

Solve for  $A^T = (Q_{S,S} \operatorname{Diag}(\vec{z}))^{-1} Q_S^T$

Solve for  $R = \operatorname{Diag}(\vec{z}) Q_{S,S} \operatorname{Diag}(\vec{z})$

**return**  $A, R$



# Recovery methods

- KL and L2 Recovery

We used these two new methods based on Bayes' rule. Considering any two words in a document  $w_1$  and  $w_2$  with corresponding topics  $z_1$  and  $z_2$ , we used  $A_{i,k}$  to index the matrix of word-topic distribution. Given a word2word matrix  $Q$ , its elements can be interpreted as  $Q_{i,j} = p(w_1 = i, w_2 = j)$ .

We denoted the row-normalized  $Q$  matrix as  $Q'$ , which is essential for us to do the recovery step, for it can be interpreted as a conditional probability

$$Q'_{i,j} = p(w_2 = j | w_1 = i)$$



# Recovery methods

- KL and L2 Recovery

We recovered matrix  $A$  by using Bayes' rule:

$$p(w_1 = i | z_1 = k) = \frac{p(z_1 = k | w_1 = i)p(w_1 = i)}{\sum_{i'} p(z_1 = k | w_1 = i')p(w_1 = i')}.$$

We can solve for  $p(w_1 = i)$  since  $\sum_j Q_{i,j} = \sum_j p(w_1 = i, w_2 = j) = p(w_1 = i)$

The algorithm finds the vector of non-negative coefficients  $p(z_1 | w_1 = i)$  which best reconstruct each row of the empirical row normalized matrix  $Q'$  as a convex combination of the rows.

The KL method use KL divergence as the objective function, while L2 method use quadratic loss as the objective to find the best reconstruct.

# Evaluation Metric: Coherence Definition

If we perfectly reconstruct topics, all the high-probability words in a topic should co-occur frequently, otherwise, the model may be mixing unrelated concepts. The Coherence is given by below equation

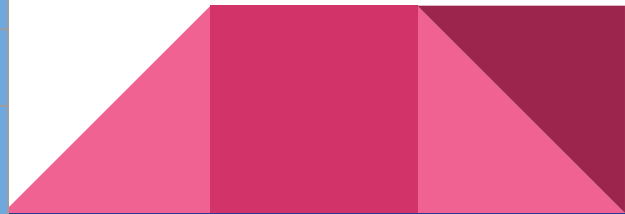
$$Coherence(\mathcal{W}) = \sum_{w_1, w_2 \in \mathcal{W}} \log \frac{D(w_1, w_2) + \epsilon}{D(w_2)},$$

The performance of a topic modeling can be evaluated by calculating the coherence. The larger value of coherence refers to better performance.



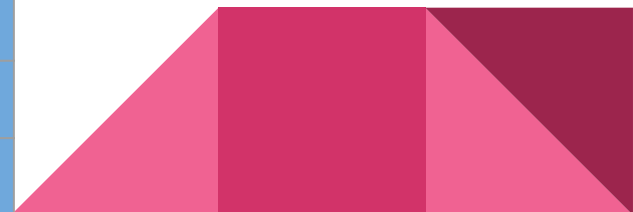
Anchors	Words Recovered
neuroscience :	algorithm network input neural output unit function training system set
<u>part</u>	part neuron input system network direction function unit head response
<u>iii</u>	iii network function neural set error result system problem training
<u>cognitive</u>	cognitive network function unit set input data neural problem system
<u>peter</u>	david michael john richard peter thomas william andrew author index
<u>science</u>	science network unit representation set input weight level data problem
<u>architecture</u>	algorithm network input neural output unit training function system set
<u>theory</u>	theory network data spike set evidence function peak neuron neural
<u>visual</u>	visual network neural system input function data unit signal output
<u>processing</u>	processing neuron input system field cortex object orientation motion direction

# 10 Toples with Original Recovery Method



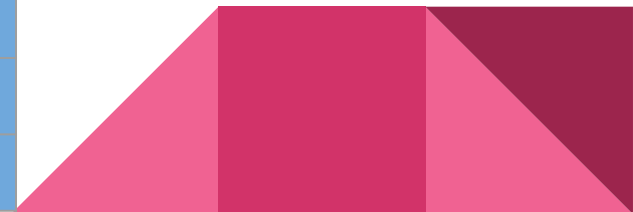
Anchors	Words Recovered
<u>neuroscience</u> :	<i>neuroscience implementation application navigation planning eric christopher speech margin discounted</i>
<u>part</u> :	part control planning navigation robot chip analog implementation action vls
<u>iii</u>	iii interconnection animal vowel connectionist linguistic psychological theories trigger perceived
<u>cognitive</u>	cognitive connectionist language representation activation grammar vowel symbol semantic linguistic
<u>paul</u> :	david michael john author index richard peter william paul thomas
<u>science</u>	science action reinforcement policy environment goal agent robot reward exploration
<u>architecture</u>	network input neural unit algorithm training output set weight system
<u>theory</u>	function network algorithm set data problem result error parameter distribution
<u>visual</u>	neuron visual input field response direction motion system activity object
<u>processing</u>	processing signal network system input neural speech output information data

# 10 Topics with Recover- KL Method



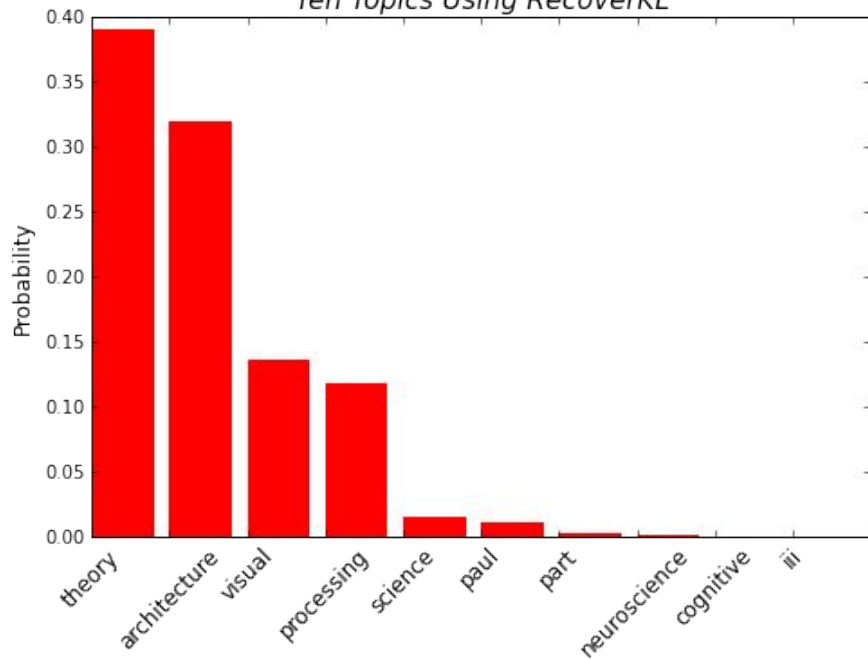
Anchors	Words Recovered
<u>implementation</u>	implementation application navigation planning david richard thomas terrence robert michael
<u>part</u>	part network function input neural algorithm system set data training
<u>iii</u>	iii application david richard thomas robert terrence michael planning andrew
<u>cognitive</u>	cognitive application david planning richard robert michael terrence thomas geoffrey
<u>peter</u>	network function input neural set system data training algorithm unit
<u>science</u>	science application control algorithm planning navigation network language function plan
<u>architecture</u>	network algorithm architecture function input neural set data training problem
<u>theory</u>	network theory function input neural set algorithm data system training
<u>visual</u>	network visual input function neural system set signal data neuron
<u>processing</u>	processing network input neural function system set data unit output

# 10 Topics with Recover- L2 Method

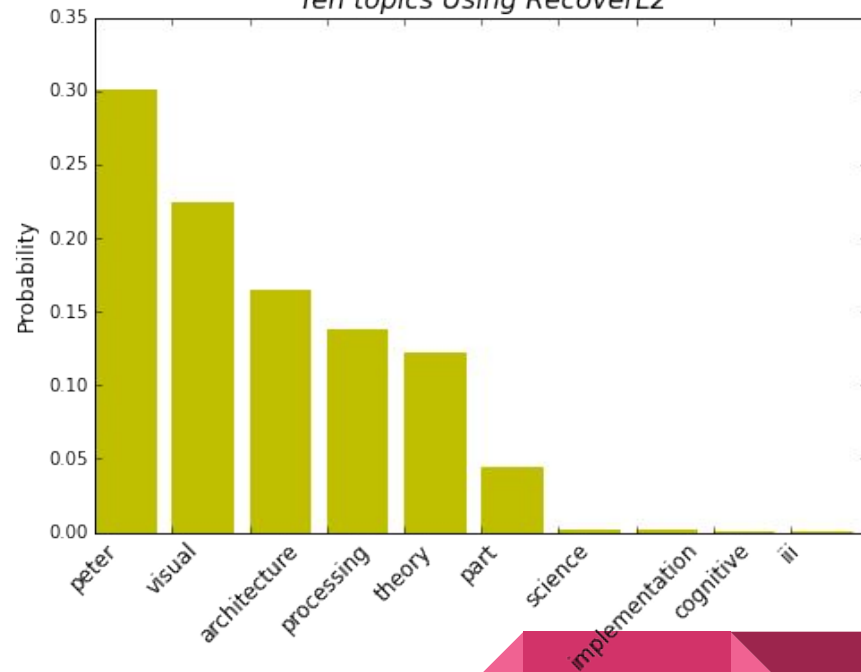


# Probability of Top Topics

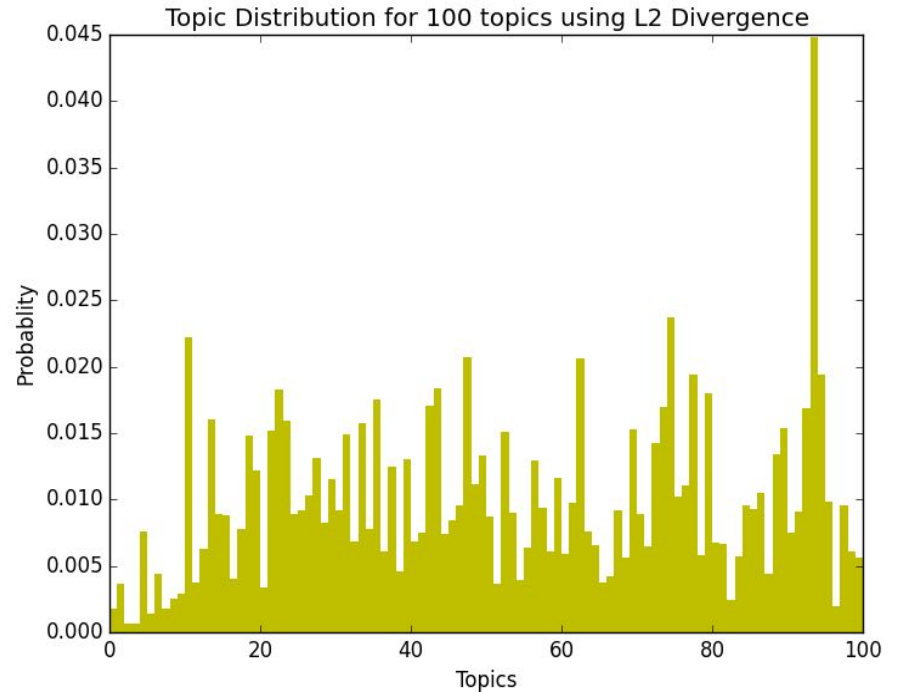
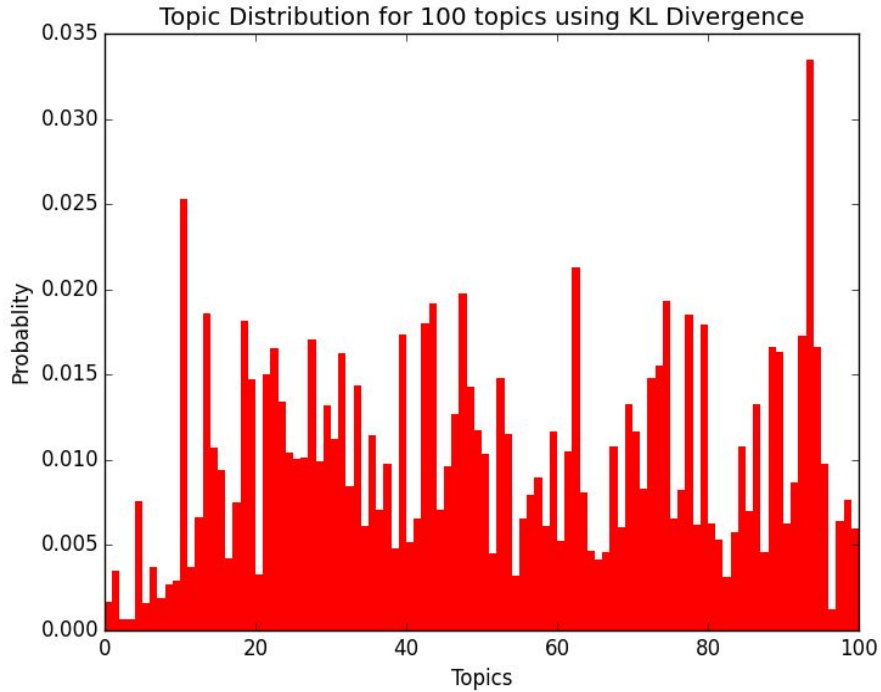
*Ten Topics Using RecoverKL*



*Ten topics Using RecoverL2*

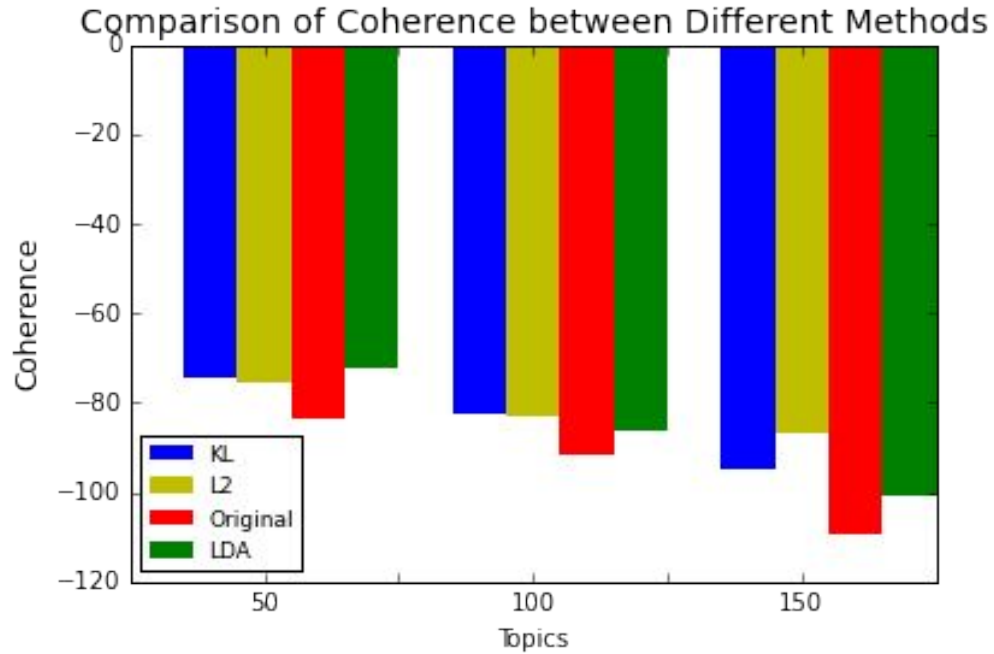


# Topic Distributions using KL and L2 divergence





# Coherence Comparison on Different Models



# Conclusion

This model considers the correlation between the topics that are generated by running the model

The Coherence for this model is better compared to LDA

The KL Recovery method performs better than the other recovery methods.

The running time of these algorithms is effectively independent of the size of the corpus



# References

**[1] Learning word vectors for sentiment analysis.,**

[http://ai.stanford.edu/~amaas/papers/wvSent\\_acl2011.pdf](http://ai.stanford.edu/~amaas/papers/wvSent_acl2011.pdf)

**[2] A practical algorithm for topic modeling with proveable guarentees.,**

[http://cs.nyu.edu/~dsontag/papers/AroraEtAl\\_icml13.pdf](http://cs.nyu.edu/~dsontag/papers/AroraEtAl_icml13.pdf)

**[3] Supplement material for A practical algorithm for topic modeling.,**

[http://cs.nyu.edu/~dsontag/papers/AroraEtAl\\_icml13\\_supp.pdf](http://cs.nyu.edu/~dsontag/papers/AroraEtAl_icml13_supp.pdf)

**[4] Latent Dirichlet Allocation.,**

<https://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf>





**Thank You !**