Topic Modeling with LSA, pLSA, LDA and Word Embedding

Hsuan-Yu Yeh (Amelie) Follow
Mar 12, 2019 · 6 min read

Natural language processing very useful even in music production. Alex Da Kid worked with IBM and created the "Not Easy". To create such a popular song, the team first need to figure out what's the topic that most people care about nowadays. They parsed the titles of New York Time, movie summaries, internet search to find out the culture trends and found out the answer was "heartbreak". This can be completed through machine learning, topic modeling. So, how does the topic modeling work?

LISTEN TO ARTICLE

REMAINING 7:37 SPEED 1X

Topic modeling and word embedding are two import fields of natural language processing. Topic model is a type of statistical model that extracts abstract topics from the corpus. Through topic modeling, we can discover hidden semantic structures in the text body. LSA use the matrix decomposition to grab the topics. pLSA and LDA are based on probabilistic model. Word embedding is used to grab features from text data. This techniques map words, sentence to vectors and numbers.

Topic Modeling

Latent Semantic Analysis (LSA)

One of foundational techniques in topic modeling. The core idea is that we construct a m*n matrix A with m documents and n words in our vocabulary. For A, each row represents a document and each column represents a word. In simple version LSA, each

entry can simply be a raw count of the number of times. Consequen[illegible] replace the row counts in the document-term matrix with tf-idf sco[illegible] account for the significance of each word in the document.

In most condition, A is very sparse, noisy and redundant across its many dimensions. As a result, to find the topics which can capture relationship between words and documents, we perform dimension reduction on A. We perform the truncated SVD. A = Uk*Sk*Vk^T. In this case, Uk ∈ m*k, Vk ∈ n*k. In both Uk and Vk, column corresponds to one of our topic. In U rows represent document vectors expressed in terms of topics. In V rows represent word vectors expressed in terms of topics.

With these document vectors and word vectors, we can apply measures such as cosine similarity to evaluate the similarity of different documents, the similarity of different words and the similarity of word, documents.

. . .

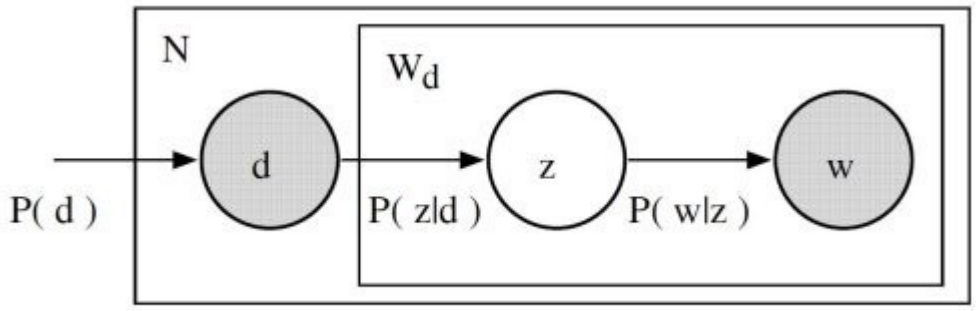
. . .

Probability Latent Semantic Analysis (pLSA)

pLSA uses probability instead of SVD to tackle the problem. The core idea is to find a probabilistic model that can generate the data we observe in our document-term matrix. We want a model P(D, W) such that P(d, w) corresponds to that entry in the matrix.

We can express the model as $P(D, W) = \sum_z P(Z)P(D|Z)P(W|Z) = P(D)\sum_z P(Z|D)P(W|Z)$. This is very similar to LSA where the probability of the topic P(Z) corresponds to the diagonal matrix. The probability of document given topic P(D|Z) corresponds to the

document-topic matrix U and the probability of word given topic P(W|Z) corresponds to the term-topic matrix V.



Latent Dirichlet Allocation (LDA)

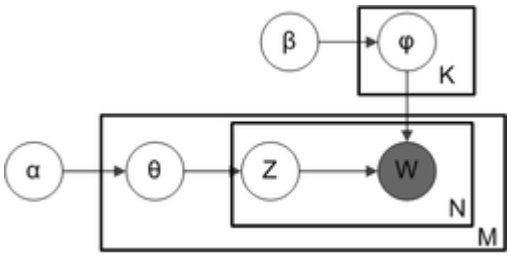
LDA is a Bayesian version of pLSA. It uses Dirichlet priors for the document-topic and word-topic distributions, lending itself to better generalization. The Dirichlet distribution provides a way of sampling probability distribution of a specific type.

From a Dirichlet disrtibution Dir(α), we draw a sample representing the topic distribution of a particular document. This topic distribution is θ. From θ, we select a particular topic Z based on the distribution.

From Dirichlet distribution Dir(β), we draw a sample representing the word distribution of the topic Z. This word distribution is φ. From φ, we choose the word w.

The process of generating each word from a document becomes

1. Choose θi ~ Dir(α), θi is the topic distribution of document i
2. Choose ϕi ~ Dir(β), ϕi is the word distribution of topic i
3. For each of the word positions i, j, where i ∈ {1, …, M}, j ∈ {1, …, N}, choose a topic zij ~ Multinomial(θi), wij ~ Multinomial(ϕzij)



In LDA, the data set serve as training data for the dirichlet distribution of dicument-topic distributions. If we haven't seen a document, we can easily sample from the dirichlet distribution and move forward from there.

Word Embedding

Word embeddings are the texts converted into numbers and there may be different numerical representations of the same texts.

Frequent Based Embedding (Bag of Words)

It is a way of extracting features from text data for use in modeling. It use the technique of vectorization, which means to convert sentence or corpus into vectors. It's like a dictionary with corpus as rows and words as columns.

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Example for bag of words

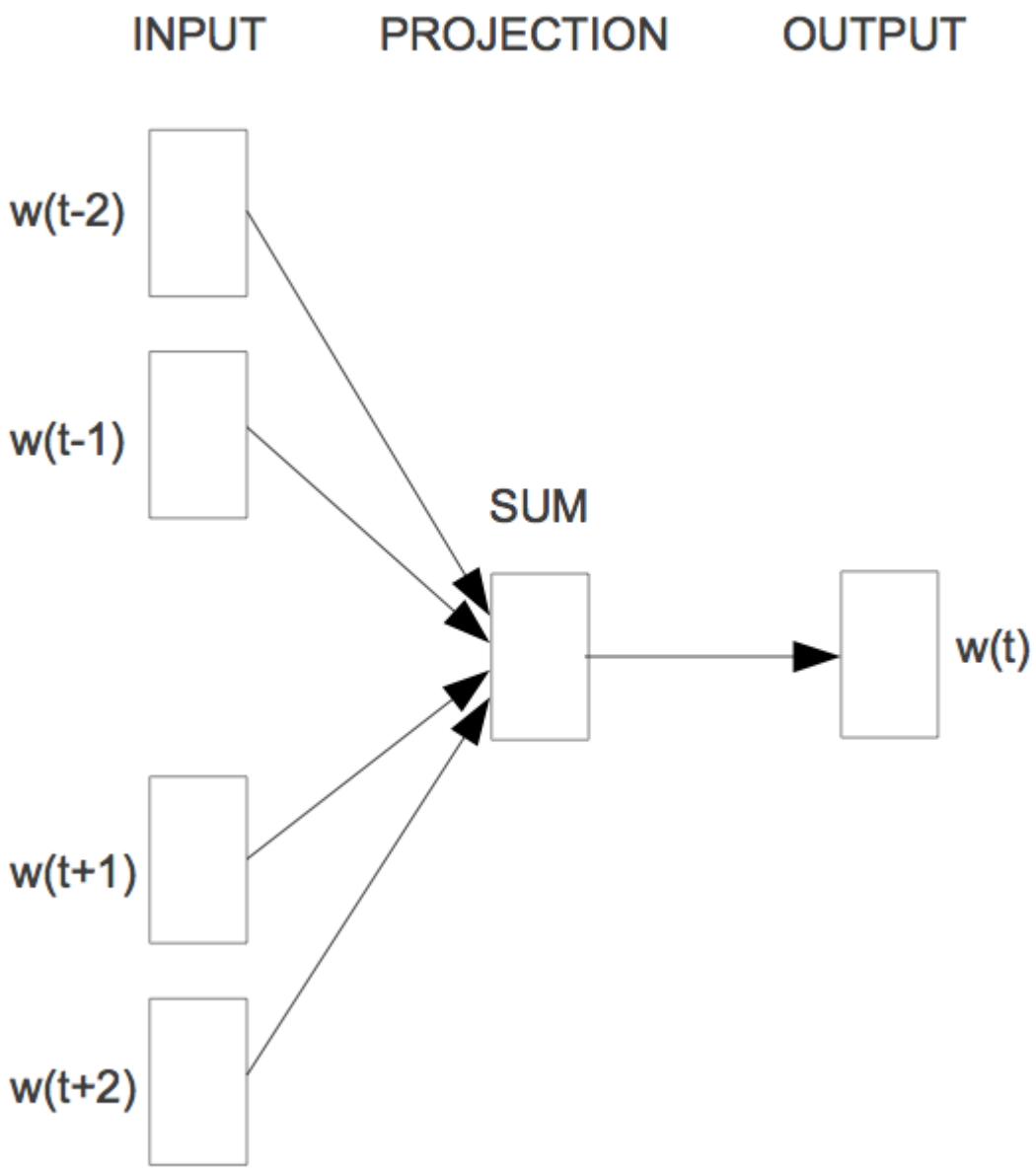
1. Count the word frequency
2. Tf-idf transformation

Prediction Based Embedding

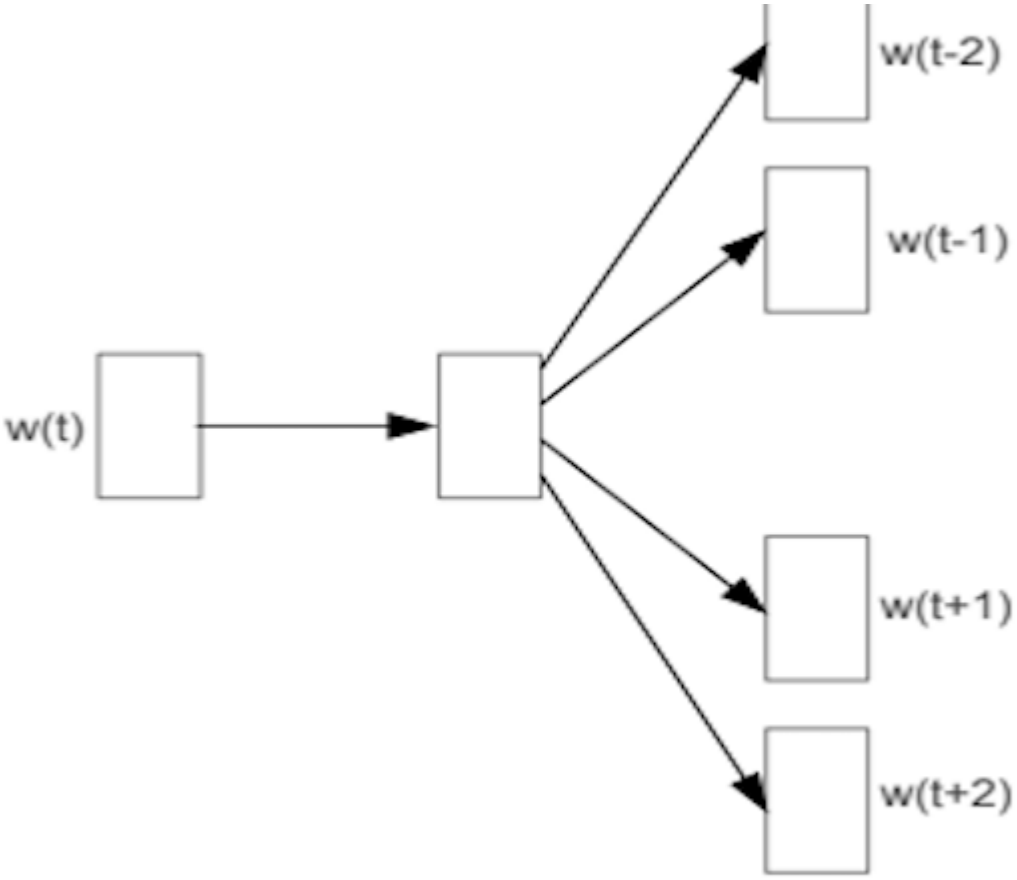
How to construct a word embedding? Word2Vec is a technique to construct embedding. There are two methods: Skip Gram and Common bag of words(CBOW).

CBOW: The model take the one hot encoding of words as input and output of Neural Network. In the process of predicting the target word, we learn the vector representation of the target word. Our goal is not to predict but to find the weight matrix and take it as our word vector. We can take use multiple context words to predict the target word.

Then take the average over all the context words input. So how word representations are generated using the context words.



Skip-Gram: In this model, we flip the process of CBOW. The input of the Neural Network is the target word and the output is the probability of each context word. For each context positions, we get the probability distributions with V dimensions. We also take the weight matrix as our word embedding.



Skip-gram

Both models got their advantage and disadvantage. CBOW is faster and has better representation for frequent words. Skip-Gram works well in small amount of data and represent rare words well.

Relationship between Topic Modeling and Word Embedding

Given word distributions produced by a topic model, we can present each word by a distributed vector embedding where each dimension representing a topic. The value means the probability of the word given the topic. Thus, we can also view topic modeling as a word embedding model.

Word embedding and topic modeling come from two different research communities. Word embeddings come from the neural net research tradition, while topic modelings come from Bayesian model research tradition. Word embedding can be used to improve topic models like Lda2Vec.

Naturallanguageprocessing Signal Processing Google Assistant Google Home

Conversational UI