Probabilistic Latent Semantic Analysis

Dan Oneaţă

1 Introduction

Probabilistic Latent Semantic Analysis (pLSA) is a technique from the category of topic models. Its main goal is to model co-occurrence information under a probabilistic framework in order to discover the underlying semantic structure of the data.

It was developed in 1999 by Th. Hofmann [7] and it was initially used for text-based applications (such as indexing, retrieval, clustering); however its use shortly spread in other fields: such as computer vision [14, 16, 10] or audio processing [5].

PLSA can be regarded in two seemingly different ways:

- Latent variable model. The probabilistic structure of pLSA is based on a statistical model, called the *aspect model*. The latent/hidden variables (represented by topics/concepts) are associated with the observed variables (represented by documents and words, for the text domain).

- Matrix factorization. Similarly to Latent Semantic Indexing (LSI) [3], pLSA aims to factorize the sparse co-occurrence matrix in order to reduce its dimensionality. However, pLSA is usually viewed as a more sound method as it provides a probabilistic interpretation, whereas LSI achieves the factorization by using only mathematical foundations (more precisely, LSI uses the singular value decomposition method).

2 Theoretical presentation

In order to make the theoretical presentation more explicit and easy to understand we will refer, without loss of generality, to the text domain[1]. For this particular application, our training data is a corpus—a large set of documents—that is usually represented in the form of a document-term matrix (this indicates the number of times each word appears in each document). The goal of pLSA is to use this co-occurrence matrix to extract the so-called "topics" and explain the documents as a mixture of them.

2.1 Latent variable model

PLSA considers that our data can be expressed in terms of 3 sets of variables:

- Documents: $d \in \mathcal{D} = \{d_1, \cdots, d_N\}$—observed variables. Let N be their number, defined by the size of our given corpus.

- Words: $w \in \mathcal{W} = \{w_1, \cdots, w_M\}$—observed variables. Let M be the number of distinct words from the corpus.

- Topics: $z \in \mathcal{Z} = \{z_1, \cdots, z_K\}$—latent (or hidden) variables. Their number, K, has to be specified a priori.

[1] Also the terminology used will be relevant to the natural language processing domain (i.e., documents, words, topics). However, it is not hard to make the correspondences for any other domain or application that uses co-occurrence data.

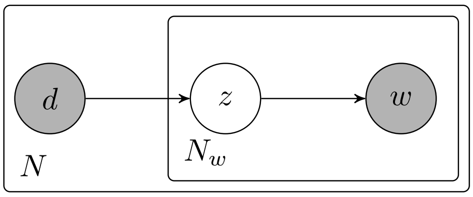


Figure 1: The graphical model using plate representation. It describes the generative process for each of the N documents in the collection. N_w denotes the number of words in document d. Each word w has associated a latent concept z from which is generated. The shaded circles indicate observed variables, while the unshaded one represents the latent variables.

These are linked in a graphical model (based on the aspect model) that associates the topics z with the observed pairs (d, w) (see Figure 1). This also describes a generative process for the documents [4]:

- First we select a document d_n with probability $P(d)$.
- For each word $w_i, i \in \{1, \cdots, N_w\}$ in the document d_n:
  - Select a topic z_i from a multinomial conditioned on the given document with probability $P(z|d_n)$.
  - Select a word w_i from a multinomial conditioned on the previous chosen topic with probability $P(w|z_i)$.

There are some important assumptions made by the presented model:

- Bag-of-words. Intuitively, each document is regarded as an unordered collection of words[2]. More precisely, this means that the joint variable (d, w) is independently sampled and, consequently, the joint distribution of the observed data will factorize as a product:

$$P(\mathcal{D}, \mathcal{W}) = \prod_{(d,w)} P(d, w).$$

- Conditional independence. This means that words and documents are conditionally independent given the topic: $P(w, d|z) = P(w|z)P(d|z)$ or $P(w|d, z) = P(w|z)$. (This can be easily proved by using d-separation into our graphical model: the path from d to w is blocked by z.)

The model can be completely defined by specifying the joint distribution. We can obtain $P(d, w)$ by using the product rule:

$$\begin{aligned} P(d, w) &= P(d)P(w|d) \\ P(w|d) &= \sum_{z \in \mathcal{Z}} P(w, z|d) \\ &= \sum_{z \in \mathcal{Z}} P(w|d, z)P(z|d). \end{aligned}$$

Using the conditional independence assumption, we obtain:

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d) \quad (1)$$

$$P(w, d) = \sum_{z \in \mathcal{Z}} P(z)P(d|z)P(w|z). \quad (2)$$

Equation 1 is the mathematical representation of the mixture model (see Figure 2). The parameters of the model are $P(w|z)$ and $P(z|d)$; their number is $(M-1)K$, respectively $N(K-1)$, which means that the total number of parameters grows linearly with the size of the corpus[3] and the model becomes prone to overfitting (as stated in [1]). The parameters

[2] http://en.wikipedia.org/wiki/Bag_of_words_model Date last accessed: 06/04/2011

[3] We have $(M - 1)K$ parameters for $P(w|z)$, instead of MK, because of the normalization constraint $\sum_{w \in \mathcal{W}} P(w|z) = 1, \forall z \in \mathcal{Z}$. The same reasoning applies for the other case, $P(z|d)$.

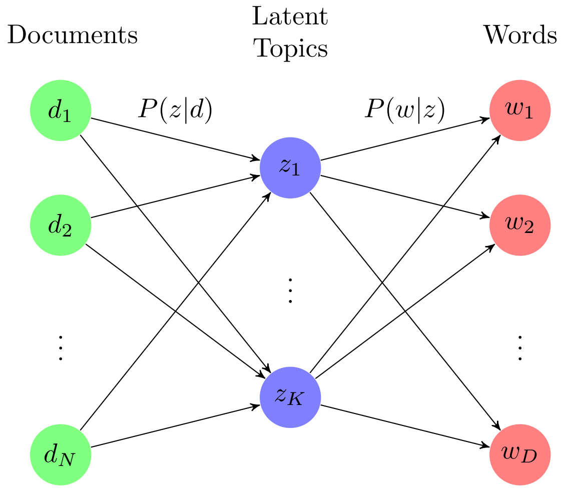


Figure 2: The general structure of pLSA model. This shows the intermediate layer of latent topics that links the documents and the words: each document can be represented as a mixture of concepts weighted by the probability $P(z|d)$ and each word expresses a topic with probability $P(w|z)$.

can be estimated via likelihood maximization, by finding those values that maximize the predictive probability for the observed word occurrences. The predictive probability of pLSA mixture model is denoted by $P(w|d)$, so the objective function is given by the following expression:

$$L = \prod_{(d,w)} P(w|d) = \prod_{d \in \mathcal{D}} \prod_{w \in \mathcal{W}} P(w|d)^{n(d,w)} \quad (3)$$

where $n(d,w)$ represents the observed frequencies, the number of times word w appears in document d. This is a non-convex optimization problem and it can be solved by using Expectation-Maximization (EM) algorithm for the log-likelihood:

$$\mathcal{L} = \log L = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d,w) \cdot \log \sum_{z \in \mathcal{Z}} P(w|z)P(z|d). \quad (4)$$

In the original paper [7], in order to avoid overfitting the author suggested an alternative heuristic approach for training—a “tempered” version of EM algorithm, similar to *deterministic annealing* [15].

2.2 Matrix factorization

Another way to present pLSA is as a matrix factorization approach. The document-word matrix that defines our dataset is a very large and sparse matrix; it has as many rows as documents N, and the number of columns is equal to the number of different words M that appear in our corpus. Its sparseness comes from the from the fact that only a small percentage of the words are used in each document depending on its particular topic. So, the idea is to somehow reduce the dimensionality of our document-word matrix as most of its entries are zero and do not offer particular information. This can be achieved by approximating the co-occurrence matrix (which it will be denoted by A) as a product of two low-rank (thinner) matrices L and R. For example:

$$A \approx \hat{A} = L \cdot R. \quad (5)$$

So, if the size of L is $N \times K$ and the size of R is $K \times M$, with $K \ll M, N$, then this will fullfil the dimensionality reduction task, because $N \cdot M \gg N \cdot K + K \cdot M$. Apart from this, we also expect that our matrices L and R reveal information about the latent structure of the data.

If we look back to Equation 1, we easily observe that what pLSA does is exactly a matrix factorization of the conditional distribution $P(w|d)$. In order to obtain the factorization of the full co-occurrence data $P(w,d)$, we use Equation 2. In terms of matrix notation, that can be rewritten as follows:

$$A = L \cdot U \cdot R. \quad (6)$$

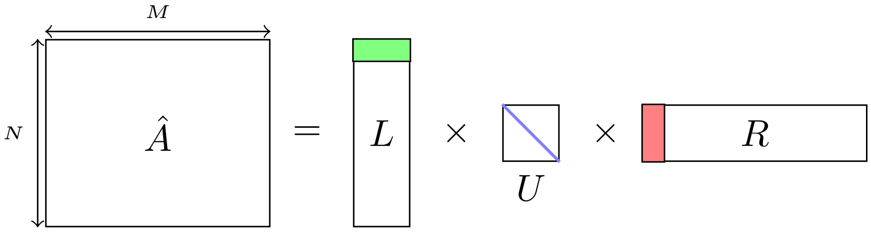


Figure 3: Alternative view of pLSA as a matrix decomposition technique. The matrix A denotes the document-term matrix. The green row represents the probabilities over a document $P(d|z)$, the blue diagonal represents the probabilities over all the topics $P(z)$ and the red column corresponds to the probabilites of a word being generated by each topic $P(w|z)$.

where he have the following relations (see Figure 3):

- L contains the document probabilities $P(d|z)$.
- U is a diagonal matrix of the prior probabilities of the topics $P(z)$.
- R corresponds to the word probability $P(w|z)$.

These matrices are non-negative and normalized, as they represent probability distribution. Consequently, these properties ensure different results from plain LSI, which uses SVD and does not impose any constraints.

3 Applications to computer vision

3.1 Object categorization

One extension of the pLSA model for computer vision applications was done by Sivic et al. [16]; they used this model on sets of images in order to extract object categories in an unsupervised manner. Also, they were able to classify novel images with the help of the learned objects and to segment images by grouping together local features that belong to a certain object.

The standard pLSA framework (described in Section 2) was used; however, instead of documents the algorithm operated on images, the words were substituted by patches/visual words, and the topic was represented by a category of an object. The most acute difference from the previous case is that the "words" are not clearly specified for a set of pictures. These can be achieved in three steps[4]:

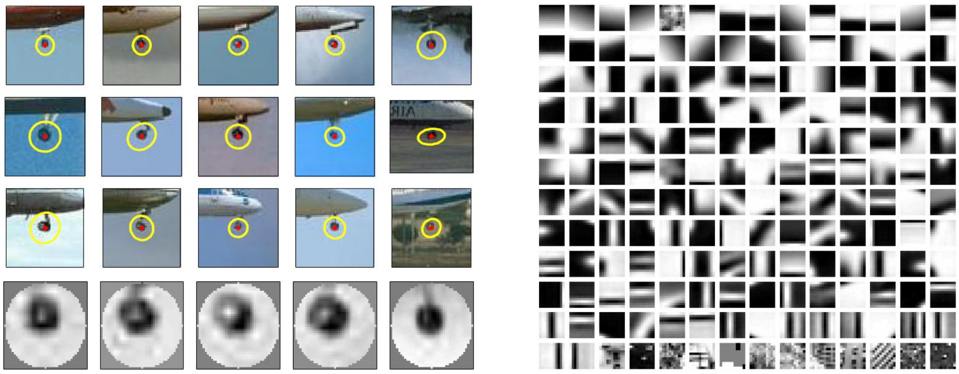
Feature detection Elliptical regions are extracted using an affine invariant interest point detector [12]—this technique is also known as *Harris-affine detector* and it "can identify similar regions between images that are related through affine transformation and have different illuminations"[5] (see Figure 4a, first 3 rows).

Feature representation The previously detected patches are scaled to circles (see Figure 4a, last row) and their scale-invariant feature transform (SIFT) [11] descriptor is computed.

Codebook generation As there are a huge number of resulted visual words (around

[4] `http://en.wikipedia.org/wiki/Bag_of_words_model_in_computer_vision` Date last accessed: 06/04/2011

[5] `http://en.wikipedia.org/wiki/Harris_affine_region_detector` Date last accessed: 06/04/2011



(a) Examples of a visual "words" (source: [16]).

(b) Example of a dictionary (source: [9]).

Figure 4: The steps performed for obtaining the set of codewords.

hundred of thousands [16]) their number is reduced through the process of vector quantization (to approximately 2000)—they are clustered using k-means algorithm and each cluster will be represented by its centroid (see Figure 4b). Thus a dictionary of "words" is determined and each image can be represented as a "bag" of these patches; consequently, the entire dataset can be represented as a co-occurrence matrix.

In the following, we briefly describe how pLSA was applied for different goals:

Object categorization The training process of pLSA yields the probabilities $P(z|d)$ and $P(w|z)$; using $P(z|d_n)$ for each image d_n, the images were classified as containing object k, where $k = \text{argmax}_{z_k \in \mathcal{Z}} P(z_k|d_n)$.

Classify unseen images New images are classified by using the so-called "fold-in" technique. First, the standard training procedure (the EM algorithm) is done on the dataset. When we have a new query image d_{new} the training algorithm is rerun, but this time $P(w|z)$ are kept fixed to their previous values, while only $P(z|d_{\text{new}})$ is updated. In this manner we obtain the mixing coefficients $P(z|d_{\text{new}})$ for the unseen image.

Segmentation Spatial segmentation can be achieved by using the posterior distribution:

$$P(z|w,d) = \frac{P(w|z,d)P(z|d)}{P(w|d)} = \\ = \frac{P(w|z)P(z|d)}{\sum_{z \in \mathcal{Z}} P(w|z)P(z|d)}.$$

This gives us the probability of every word in an image of being generated by a certain topic. In [16], they selected the words that have a probability larger than 0.8.

3.2 Auto-annotation

Monay et al. addressed the problem of image auto-annotation with pLSA model [13, 14]. In their first work [13], they use a pLSA-mixed system where every document is represented by a pair image-annotation and the words are a concatenation of visual words and textual words (that are present in the annotation). This approach is based on the fact that the latent topics are the same for both the visual words and the text. However, for datasets where the visual modality does not correspond to textual modailty, this method has drawbacks; this happens because if two pLSA models are fitted—one on images and the other on text—they learn different topics [14]. In their next paper [14], they use two linked pLSA models that share the distribution over the topics $P(z|d)$. The learning process is done in two steps: first a model is fitted only to the textual information and then the other model uses the previously obtained $P(z|d)$ and learns the distribution over the visual words $P(\text{visual words}|z)$. In this way the semantic consistency is ensured.

Further reading

- Original papers that introduce pLSA [7, 6, 8].
- *Latent Dirichlet Allocation* (Bayesian version of pLSA) [1] and its application to computer vision [9].
- Other applications of pLSA to computer vision:
  - *Scene classificaion via pLSA* [2].
  - *Multilayer pLSA for mutlimodal image retrieval* [10].

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[2] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification via pLSA. In *In Proc. ECCV*, pages 517–530, 2006.

[3] Scott Deerwester. Improving Information Retrieval with Latent Semantic Indexing. In Christine L. Borgman and Edward Y. H. Pai, editors, *Proceedings of the 51st ASIS Annual Meeting (ASIS '88)*, volume 25, Atlanta, Georgia, October 1988. American Society for Information Science.

[4] Kevin Gimpel. Modelling topics. `http://www.cs.cmu.edu/~nasmith/LS2/gimpel.06.pdf`, 2006. Date last accessed: 06/04/2011.

[5] Matthew D. Hoffman, David M. Blei, and Perry R. Cook. Finding latent sources in recorded music with a shift-invariant hdp. In *International Conference on Digital Audio Effects (DAFx)*, 2009.

[6] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[7] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[8] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42:177–196, January 2001.

[9] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[10] Rainer Lienhart, Stefan Romberg, and Eva Hörster. Multilayer pLSA for multimodal image retrieval. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, CIVR '09, pages 9:1–9:8, New York, NY, USA, 2009. ACM.

[11] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[12] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 128–142, London, UK, UK, 2002. Springer-Verlag.

[13] Florent Monay and Daniel Gatica-Perez. On image auto-annotation with latent space models. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 275–278, New York, NY, USA, 2003. ACM.

[14] Florent Monay and Daniel Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 348–351, New York, NY, USA, 2004. ACM.

[15] K. Rose, E. Gurewwitz, and G. Fox. A deterministic annealing approach to clustering. *Pattern Recogn. Lett.*, 11:589–594, September 1990.

[16] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the International Conference on Computer Vision*, 2005.