Computational Intelligence Laboratory

Lecture 4

Non-Negative Matrix Factorization

Thomas Hofmann

ETH Zurich – cil.inf.ethz.ch

15 March 2019

Section 1

Motivation

Introduction: Topic Models

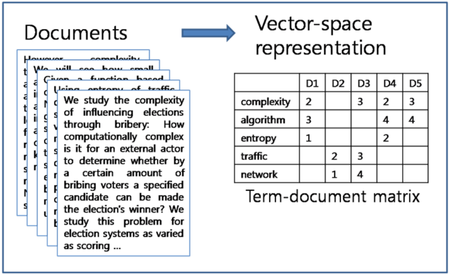
- Challenge
  - given: corpus of text documents (e.g. web pages)
  - goal: find low-dimensional document representation in **semantic space** of topics or concepts – **aboutness** of documents
  - also known as **topic models**
- Approach: predictive model
  - log-liklihood of predicting words in document
  - MLE: probabilistic Latent Semantic Analysis (pLSA)
  - Bayesian: Latent Dirichlet Allocation (LDA)
  - related to non-negative matrix decomposition

Document Representation: Pre-Processing

- Vocabulary
  - all "meaningful" words (=terms) in a language
  - extracted from corpus documents via **tokenization**
  - large cardinality (e.g. $\sim$1-100 million)
- Term filtering
  - exclude **stop words** ("the", "is", "at", "which", etc.).
  - exclude infrequent words, misspellings, tokenizer errors, etc.
- Term normalization
  - **stemming** (optionally): reduce word to stem/lemma
  - example: "argue", "argued", "argues", "arguing", and "argus" reduce to the stem "arg"

Document Representation: Bag-of-Words

- Bag-of-word Representation
  - ignore order of words in sentences/document
  - reduce data to co-occurrence counts
  - see previous lecture: word context = entire document
  - document = M-dimensional vector of counts, very **sparse**!



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

Section 2

Probabilistic LSA

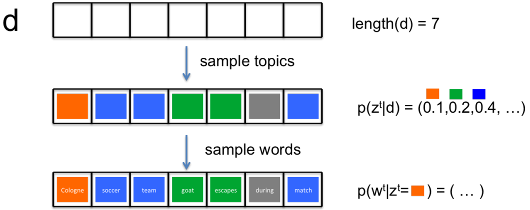
Probabilistic LSA: Topic Model

- Topic parameters = word distribution
- Document = **mixture of topics**
  - $\neq$ probabilistic assignment
  - example: document on *soccer world cup 2022 in Dubai*
    - soccer vocabulary (e.g. "teams","play", "soccer", "match")
    - political vocabulary (e.g. "labor", "corruption", "president")
    - mixing weights $\neq$ uncertainty about correct topic
- Goal: Discover topics in an unsupervised fashion.

Probabilistic LSA: Two-Stage Sampling

- Two-stage (hierarchical) sampling:
  - (1) sample topic for each token
  - (2) sample token, given sampled topic

- Model parameters
  - each document = specific mix of topics (colors): $p(z|d)$
  - each topic (color) = specific distribution of words: $p(w|z)$



Probabilistic LSA: Basic Model

- **Context model**:
  occurrence of word w in context/document d

$$p(w|d) = \sum_{z=1}^{K} p(w|z)p(z|d)$$

  - identify topics with integers $z \in \{1, \ldots, K\}$ (K: pre-specified)
  - relative to a fixed "slot" (i.e. fixed position in document)
  - identical distribution for every slot

- **Conditional independence** assumption (*)

$$p(w|d) = \sum_{z} p(w, z|d) = \sum_{z} p(w|d, z)p(z|d) \stackrel{*}{=} \sum_{z} p(w|z)p(z|d)$$

  - topics represent regularities common to the entire collection

Probabilistic LSA: Log-Likelihood

- Summarize data into co-occurrence counts $\mathbf{X} = x_{ij}$ (# occurrences of w_j in document d_i)
- Alternatively: multiset $\mathcal{X}$ over index pairs (i, j)
- **Log-likelihood**

$$\ell(\mathbf{U}, \mathbf{V}) = \sum_{i,j} x_{ij} \log p(w_j|d_i) = \sum_{(i,j)\in\mathcal{X}} \log \sum_{z=1}^{K} \underbrace{p(w_j|z)}_{=:v_{zj}} \underbrace{p(z|d_i)}_{=:u_{zi}}$$

  - two types of parameters:
  - $u_{zi} \geq 0$ such that $\sum_z u_{zi} = 1$ $(\forall i)$
  - $v_{zj} \geq 0$ such that $\sum_j v_{zj} = 1$ $(\forall z)$

Expectation Maximization for pLSA

- Missing data $Q_{zij} \in \{0, 1\}$: w_j in d_i generated via z, $\sum_z Q_{zij} = 1$
- Variational parameters $q_{zij} = \Pr(Q_{zij} = 1)$, $\sum_z q_{zij} = 1$
- Lower bound from Jensen's inequality

$$\log \sum_{z=1}^{K} q_{zij} \frac{u_{zi} v_{zj}}{q_{zij}} \geq \sum_{z=1}^{K} q_{zij} \left[\log u_{zi} + \log v_{zj} - \log q_{zij}\right]$$

- Solve for optimal q (**Expectation Step**)

$$q_{zij} = \frac{u_{zi} v_{zj}}{\sum_{k=1}^{K} u_{ki} v_{kj}} = \frac{p(w_j|z)p(z|d_i)}{\sum_{k=1}^{K} p(w_j|k)p(k|d_i)}$$

  - $\Longrightarrow$ posterior of Q_{zij} under model

Expectation Maximization for pLSA (cont'd)

- Solve for optimal parameters (**Maximization** Step)

$$u_{zi} = \frac{\sum_j x_{ij} q_{zij}}{\sum_j x_{ij}}, \qquad v_{zj} = \frac{\sum_i x_{ij} q_{zij}}{\sum_{i,l} x_{il} q_{zil}},$$

  - numerator: simple weighted counts
  - denominator: ensure proper normalization
- EM for MLE in pLSA ;-)
  - guaranteed convergence (cf. mixture models)
  - **not** guaranteed to find global optimum

Topics Discovered by pLSA

"segment 1"	"segment 2"	"matrix 1"	"matrix 2"	"line 1"	"line 2"	"power 1"	"power 2"
imag	speaker	robust	manufactur	constraint	alpha	**POWER**	load
SEGMENT	speech	**MATRIX**	cell	**LINE**	redshift	spectrum	memori
texture	recogni	eigenvalu	part	match	**LINE**	omega	vlsi
color	signal	uncertainti	**MATRIX**	locat	galaxi	mpc	**POWER**
tissue	train	plane	cellular	imag	quasar	hsup	systolic
brain	hmm	linear	famili	geometr	absorp	larg	input
slice	source	condition	design	impos	high	redshift	complex
cluster	speakerind.	perturb	machinepart	segment	ssup	galaxi	arrai
mri	**SEGMENT**	root	format	fundament	densiti	standard	present
volume	sound	suffici	group	recogn	veloc	model	implement

Table: Eight selected topics from a 128 topic decomposition. The displayed word stems are the 10 most probable words in the class-conditional distribution $p(\text{word}|\text{topic})$, from top to bottom in descending order.

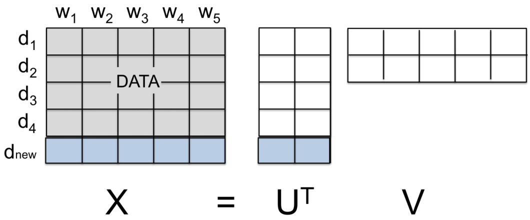
Hofmann, Thomas. Probabilistic latent semantic indexing. ACM SIGIR Forum. Vol. 51. No. 2. ACM, 2017. (re-print from 1999)

Section 3

Latent Dirichlet Allocation

Generative Document Model

- Probabilistic LSA: both dimensions of matrix are fixed
- Generative document model: how to sample new document?
- Co-occurrence matrix: how to sample additional row of $\mathbf{X}$?



- Need to be able to sample topic weights $\mathbf{u}_i = (u_{1i}, \dots, u_{Ki})^\top$ for a new document
- Combine with existing $\mathbf{V}$ to predict new data row

Latent Dirichlet Allocation (LDA)

- $\mathbf{u}_i$ is a probability vector, "simplest" (conjugate) distribution = **Dirichlet distribution**

$$p(\mathbf{u}_i|\alpha) \propto \prod_{z=1}^{K} u_{zi}^{\alpha_z - 1}$$

  - given α parameters (K dim.), can generate topic weights
  - but, we can do more ...
- Bayesian view: treat $\mathbf{U}$ as nuisance parameters
  - $\mathbf{U}$ needs to be averaged out
  - $\mathbf{V}$ are real parameters, $\mathbf{U}$ can be re-constructed, if needed
  - advantages in terms of model averaging

Latent Dirichlet Allocation: Bayesian View

- LDA model (fixed document length $l = \sum_j x_j$)
  - **multinomial** observation model ($\mathbf{x}$ = word count vector)

$$p(\mathbf{x}|\mathbf{V}, \mathbf{u}) = \frac{l!}{\prod_j x_j!} \prod_j \pi_j^{x_j}, \quad \pi_j := \sum_z v_{zj} u_z$$

  - Bayesian averaging over $\mathbf{u}$

$$p(\mathbf{x}|\mathbf{V}, \alpha) = \int p(\mathbf{x}|\mathbf{V}, \mathbf{u})\, p(\mathbf{u}|\alpha)\, d\mathbf{u}$$

- Generative model
  - for each d_i: sample $\mathbf{u}_i \sim$ Dirichlet$(\alpha) \implies$ integrate out
  - for each word slots w^t, $1 \leq t \leq l_i \implies$ iid. $=$ product
    - sample topic $z^t \sim$ Multi$(\mathbf{u}_i) \implies$ latent, sum out
    - then sample $w^t \sim$ Multi$(\mathbf{v}_{z^t}) \implies$ observable

Latent Dirichlet Allocation: Algorithms

- Learning algorithms
  - variational expectation maximization
  - Markov Chain Monte Carlo (MCMC): collapsed Gibbs sampling
  - distributed, large-scale implementations (100Ms of documents)
  - (beyond the scope of this lecture...)

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of Machine Learning Research, 2003, pp. 993-1022.

Latent Dirichlet Allocation: Examples

Example from Blei, 2012

Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.

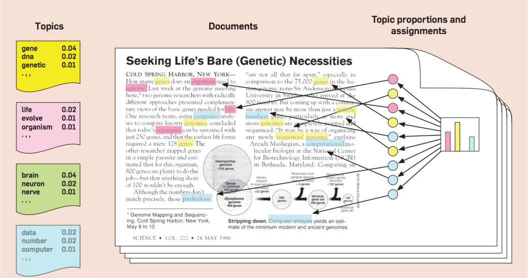
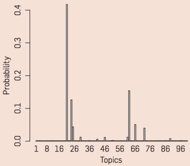


Figure 2. Real inference with LDA. We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left are the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Section 4

Non-Negative Matrix Factorization

Non-Negative Matrix Factorization

- Count matrix $\mathbf{X} \in \mathbb{Z}_{\geq 0}^{N \times M}$
- Non-negative matrix factorization (NMF) of $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{U}^\top \mathbf{V}, \quad x_{ij} = \sum_z u_{zi} v_{zj} = \langle \mathbf{u}_i, \mathbf{v}_j \rangle$$

  - constraints on matrix factors $\mathbf{U}$ and $\mathbf{V}$
    - non-negativity – as all parameters are probabilities
    - normalization – $\mathbf{U}, \mathbf{V}$ are L_1 column-normalized
  - approximation quality measured via log-likelihood
  - dimension reduction: $N \cdot M \gg (N + M)K - N - M$

NMF for Quadratic Cost Function

- pLSA: just one instance of a non-negative matrix factorization
- Variation: non-negative data $\mathbf{X}$ with quadratic cost function
  $=$ non-negative matrix approximation

$$\min_{\mathbf{U},\mathbf{V}} \quad J(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{U}^\top \mathbf{V}\|_F^2.$$

$$\text{s.t.} \quad u_{zi}, v_{zj} \geq 0 \quad (\forall i, j, z) \quad (\text{non-negativity})$$

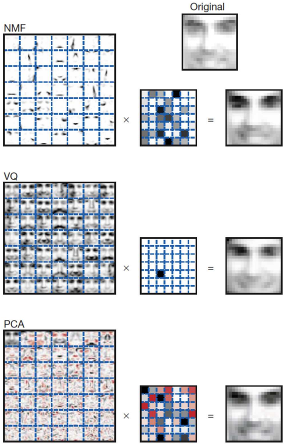
- Similar as pLSA, but ...
  - different sampling model: Gaussian vs. multinomial
  - different objective: quadratic instead of KL divergence
  - different constraints (not normalized)

Part-Based Representation of Faces

- NMF is useful when modelling non-negative data (e.g. images = non-negative intensities)

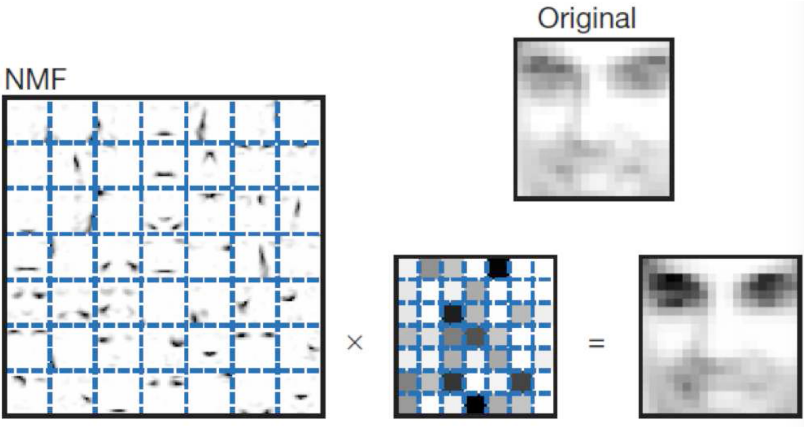
- Additive superpositions without cancellations $\Longrightarrow$ NMF leads to part-based representations

- vs. vector quantization, K-means: combination of multiple basis images



D.D. Lee & H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature, 40, 1999.

Part-Based Representation of Faces (zoom-in)



NMF Algorithm: Quadratic Costs

- Alternating least squares
  - convex in $\mathbf{U}$ given $\mathbf{V}$ and vice versa, but not jointly in $(\mathbf{U}, \mathbf{V})$
  - $\Rightarrow$ alternate optimization of $\mathbf{U}$ and $\mathbf{V}$, keeping the other fixed
  - normal equations in matrix notation

$$\left(\mathbf{U}\mathbf{U}^\top\right)\mathbf{V} = \mathbf{U}\mathbf{X}, \quad \text{and} \quad \left(\mathbf{V}\mathbf{V}^\top\right)\mathbf{U} = \mathbf{V}\mathbf{X}^\top$$

  - solved via QR-decomposition or gradient descent
  - project in between alternations – non-negativity!

$$u_{zi} = \max\{0, u_{zi}\}, \quad v_{zj} = \max\{0, v_{zj}\}$$

- More detailed discussion of algorithms for NMF see:

  Berry, M.W. et al.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics & Data Analysis, 52(1), 2007, pp.155-173.

pLSA & NMF: Discussion

- Matrix factorization obeying non-negativity and (optionally, pLSA) normalization constraints
- Different cost functions: multinomial likelihood, quadratic loss
- Iterative optimization (EM algorithm, projected ALS)
- Interpretability of factors: topics, parts, etc.
- Wide range of applications