Improving LSA word weights for document classification

Bc. Vladimír Macko[1]
supervisor: RNDr. Kristína Malinovská, PhD.[1]

[1]Comenius University
Faculty of Mathematics, Physics and Informatics

June 13, 2018

Overview

- Introduction
- Problem outline
- Our work
- Results

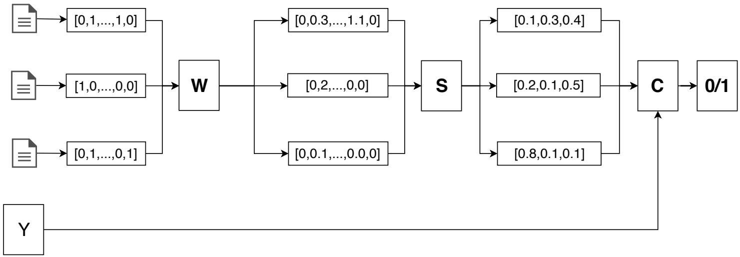
Document classification

Sentiment analysis

This was a terrible movie = negative sentiment

- create representation for words
- create representation for document
- predict

LSA



W: reweighting
S: decomposition
C: classifier

SVD

$$(\mathbf{d}_i) \rightarrow \overset{\substack{M \\ \mathbf{t}_j^T \\ \downarrow}}{\begin{bmatrix} x_{1,1} \dots x_{1,n} \\ \vdots \quad \ddots \quad \vdots \\ x_{i,1} \dots x_{i,n} \\ \vdots \quad \ddots \quad \vdots \\ x_{m,1} \dots x_{m,n} \end{bmatrix}} = \mathbf{u}_i \rightarrow \overset{U}{\begin{bmatrix} [\quad \mathbf{u}_1 \quad] \\ \vdots \\ [\quad \mathbf{u}_m \quad] \end{bmatrix}} \cdot \overset{\Sigma}{\begin{bmatrix} \sigma_1 \dots & 0 \\ \vdots & \ddots \vdots \\ 0 \dots & \sigma_l \end{bmatrix}} \cdot \overset{V^T}{\begin{bmatrix} \begin{bmatrix} \\ \mathbf{v}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{v}_n \\ \\ \end{bmatrix} \end{bmatrix}}$$

d_i: document as bag of words
u_i: word vector
M: co-occurrence matrix
v_i: document vector
d_iU: lower dimensional embedding

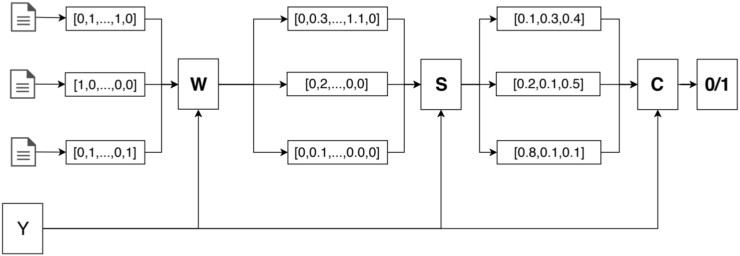
LSA problems

- Most representative features, not most discriminative
- Sensitive to preprocessing and stop words
- Sensitive to weights
- Unsupervised and can forget things

Current solutions

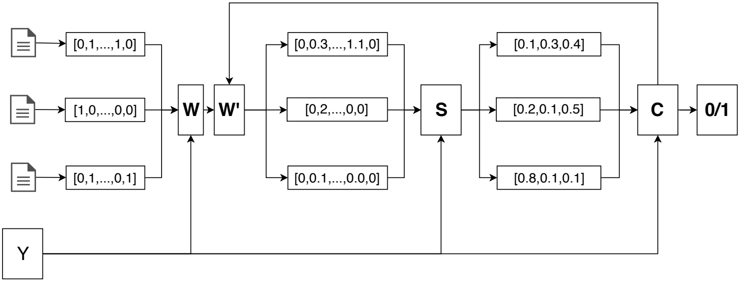
- Preprocessing
- Weight - Mutual information [Wu et al., 2017], [Deng et al., 2014]
- Supervised weights: TF-KLD [Ji and Eisenstein, 2013], [Lan et al., 2009]

Current solutions



W: reweighting
S: decomposition
C: classifier

eLSA



eLSA

- Apply weighting scheme w, rescale with w', factorize, predict
- Training the predictor, optimize w'

LSA used in similar manner in [Ionescu et al., 2015]

Gradient descent

- Co-occurrence matrix M
- Weight vector w'
- SVD: $U\Sigma V^T$
- Simple classifier: $\sigma(v\theta + b)$

- Reweighted matrix $M \circ w'$
- SVD decomposition $M \circ w' = U\Sigma V^T$
- Compute embedding $v = d \circ w'U$
- Train classifier $\hat{y} = \sigma(v\theta + b)$ to minimize $E = \frac{1}{2}(\hat{y} - y)^2$
- Compute derivative $\frac{\partial E}{\partial w'} = (\hat{y} - y)\hat{y}(1 - \hat{y})\Theta U$
- Update weights: $w' = w' - \alpha\frac{\partial E}{\partial w'}$

Evaluation

Datasets from SentEval [Conneau et al., 2017]

- Customer review dataset (CR)
- Movie review (MR)
- Subjective vs objective (SUBJ)
- Opinion polarity (MPQA)
- Questions types (TREC), actually 6 dataset

eLSA learning curves

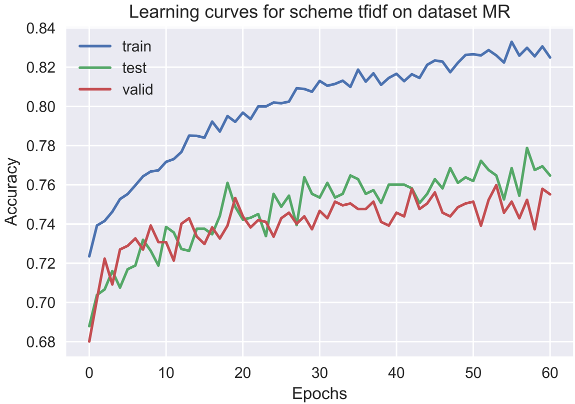


Figure 1: Learning curve for eLSA with tfidf weights on MR dataset

eLSA results

scheme	lsa	CR	MPQA	MR	SUBJ
None	200	**0.01**	**0.02**	**0.06**	**0.02**
	300	**0.02**	**0.02**	**0.05**	-0.0
	400	**0.03**	**0.01**	**0.04**	**0.01**
tfchi2	200	**0.01**	0.0	**0.01**	**0.01**
	300	0.0	-0.0	**0.02**	**0.01**
	400	**0.01**	0.0	**0.03**	**0.02**
tfgr	200	**0.01**	-0.0	**0.01**	**0.02**
	300	**0.01**	-0.0	**0.01**	**0.01**
	400	**0.03**	**0.01**	**0.01**	**0.02**

Table 1: Accuracy increase over LSA

eLSA results

scheme	lsa	CR	MPQA	MR	SUBJ
tfidf	200	**0.04**	**0.06**	**0.07**	**0.01**
	300	-0.0	**0.05**	**0.05**	0.0
	400	-0.01	**0.03**	**0.02**	**0.01**
tfig	200	0.0	**0.01**	**0.01**	-0.0
	300	0.0	**0.01**	**0.01**	**0.01**
	400	**0.03**	0.0	**0.02**	**0.01**
tfor	200	**0.01**	0.0	0.0	**0.01**
	300	0.0	0.0	-0.0	0.0
	400	-0.0	**0.02**	-0.03	**0.01**

Table 2: Accuracy increase over LSA

Insight

words	w'
is	6.25
how	5.87
what	3.73
in	3.60
mean	3.51
of	3.10
come	3.09
long	2.96
for	2.94
the	2.39

(a) Words with highest w'

words	w'
from	0.42
its	0.41
nickname	0.38
address	0.34
abbreviation	0.32
fast	0.32
term	0.25
word	0.24
between	0.04
?	0.00

(b) Words with lowest w'

Table 3: Most reweighted words on DESC dataset for scheme TFIDF

Insight

words	w'
is	7.69
are	4.52
what	3.52
mean	3.44
origin	3.42
difference	3.20
much	2.91
long	2.79
where	2.72
definition	2.71

(a) Words with highest w'

words	w'
out	1.00
name	0.98
you	0.97
does	0.93
in	0.90
who	0.83
do	0.71
?	0.59
was	0.46
the	0.00

(b) Words with lowest w'

Table 4: Most reweighted words on DESC dataset for scheme TFIG

Other experiments

- word vectors baselines
- learning rates for w'
- batch gradient descent
- stochastic gradient descent
- even more datasets

Literature I

[Altszyler et al., 2016] Altszyler, E., Sigman, M., Ribeiro, S., and Slezak, D. F. (2016).

Comparative study of lsa vs word2vec embeddings in small corpora: a case study in dreams database.

arXiv preprint arXiv:1610.01520.

[Bottou and Bousquet, 2008] Bottou, L. and Bousquet, O. (2008).

The tradeoffs of large scale learning.

In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. NIPS Foundation (http://books.nips.cc).

[Brand, 2006] Brand, M. (2006).

Fast low-rank modifications of the thin singular value decomposition.

Linear algebra and its applications, 415(1):20–30.

Literature II

[Conneau et al., 2017] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017).
Supervised learning of universal sentence representations from natural language inference data.
arXiv preprint arXiv:1705.02364.

[Deng et al., 2014] Deng, Z.-H., Luo, K.-H., and Yu, H.-L. (2014).
A study of supervised term weighting scheme for sentiment analysis.
Expert Systems with Applications, 41(7):3506–3513.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep Learning.
MIT Press.
http://www.deeplearningbook.org.

Literature III

[Ionescu et al., 2015] Ionescu, C., Vantzos, O., and Sminchisescu, C. (2015).
Training deep networks with structured layers by matrix backpropagation.
arXiv preprint arXiv:1509.07838.

[Ji and Eisenstein, 2013] Ji, Y. and Eisenstein, J. (2013).
Discriminative improvements to distributional sentence similarity.
In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 891–896.

[Lan et al., 2009] Lan, M., Tan, C. L., Su, J., and Lu, Y. (2009).
Supervised and traditional term weighting methods for automatic text categorization.
IEEE transactions on pattern analysis and machine intelligence, 31(4):721–735.

Literature IV

[Levy and Goldberg, 2014] Levy, O. and Goldberg, Y. (2014).
Neural word embedding as implicit matrix factorization.
In *Advances in neural information processing systems*, pages 2177–2185.

[Levy et al., 2015] Levy, O., Goldberg, Y., and Dagan, I. (2015).
Improving distributional similarity with lessons learned from word embeddings.
Transactions of the Association for Computational Linguistics, 3:211–225.

[Maas et al., 2011] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011).
Learning word vectors for sentiment analysis.
In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Literature V

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014).
Glove: Global vectors for word representation.
In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Rumelhart, 1986] Rumelhart, D. (1986).
Learning internal representations by error propagation.
Nature, 323:533–536.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988).
Term-weighting approaches in automatic text retrieval.
Information processing & management, 24(5):513–523.

[Wu et al., 2017] Wu, H., Gu, X., and Gu, Y. (2017).
Balancing between over-weighting and under-weighting in supervised term weighting.
Information Processing & Management, 53(2):547–557.

Thank you for your attention

Opponent's review

Notation

- "označenia bez akéhokoľvek vysvetlenia"
- "matica M"
- "SVD ako konštanta"
- "documenty alebo vety": "We consider the sentences to be basically identical to documents as they both can be considered to be sequences of words."

Opponent's review

Bibliography

- 62 citations on 6 pages
- researched other thesis (Vajdová, 2017)
- stochastic gradient descent: [19] [Goodfellow et al., 2016], [8] [Bottou and Bousquet, 2008], [55] [Rumelhart, 1986],
- TF-IDF: [56], [Salton and Buckley, 1988]

Opponent's review

Weighting schemes

- Weighting schemes [61] [29] [18]

$$
\begin{aligned}
ig = & \frac{a}{N}\log_2\frac{aN}{(a+b)(a+c)} + \frac{b}{N}\log_2\frac{bN}{(a+b)(b+d)} + \\
& \frac{c}{N}\log_2\frac{cN}{(a+c)(c+d)} + \frac{d}{N}\log_2\frac{dN}{(b+d)(c+d)}
\end{aligned}
$$

$$
gr = \frac{ig}{-\frac{a+b}{N}\log_2\frac{a+b}{N} - \frac{c+d}{N}\log_2\frac{c+d}{N}}
$$

Opponent's review

Default model parameters

- mentioned the relevant ones
- others: **penalty**, dual, tol, C, fit_intercept, intercept_scaling, class_weight, random_state, solver, max_iter, multi_class, warm_start, **kernel**, degree, gamma, coef0, shrinking, probability, cache_size, decision_function_shape, alpha, window, min_count, sample, seed, workers, min_alpha, sg, hs, negative, cbow_mean, hashfxn, iter, null_word, trim_rule, sorted_vocab, batch_words, compute_loss, callbacks, num_topics, id2word, chunksize, decay, distributed, onepass, power_iters, extra_samples

Opponent's review

Others

- "Ako sa spoja TF a IDF váhy do jednej": multiplication
- Classifier in 4.2.3: logistic regression

Opponent's questions

Constrains on w'

- We tried regularization, but results were poor
- Other constrains are extremely hard (GANS)
- In practice, results were fine

w' vs $2w'$

- In theory, no difference
- In practice the classifier may be regularized
- Experimentally, weights are centered around 1 (4.4.1.2)

Underweighting vs overweighting

- Relative change in ordering
- Notions of importance

Supervisor's review

Datasets

- Customer review dataset (CR)
- Movie review (MR)
- Subjective vs objective (SUBJ)
- Opinion polarity (MPQA)
- Questions types (TREC)
    - ABBR
    - DESC
    - ENTY
    - HUM
    - LOC
    - NUM

Supervisor's review

Time complexity

- LSA: $1 - 3$, complexity depends on SVD
- eLSA: $LSA \times epochs$, (35)
- word2vec: 5, $C \times (D + D \times \log_2(V))$

Count vs. prediction

Prediction

- extremely popular
- huge performance gains
- less memory demanding

Count

- less hyperparameters
- easier to “train”
- teoreticaly based

Count vs prediction

Glove vectors as explicit factorization

- Neural word embedding as implicit matrix factorization [Levy and Goldberg, 2014]

Hyperparameters matter

- Improving distributional similarity with lessons learned from word embeddings [Levy et al., 2015]

Does not work well on small datasets

- Comparative study of LSA vs Word2vec embeddings in small corpora [Altszyler et al., 2016]