Final Project

Detection of Negative Reviews in Online Stores

Team #8

Mikhail Sidorenko

Egor Baryshnikov

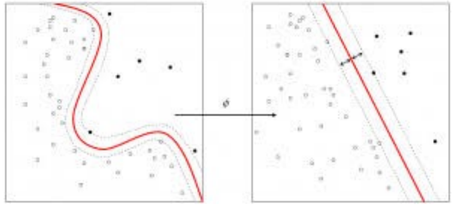
Roman Teplykh

Mentor

Evgeny Frolov

Possible applications of sentiment analysis

Possible workflow may be look like this



Data → Preprocessing → Feature Extraction → Classification

Data we used

Amazon Reviews dataset*

*http://jmcauley.ucsd.edu/data/amazon/

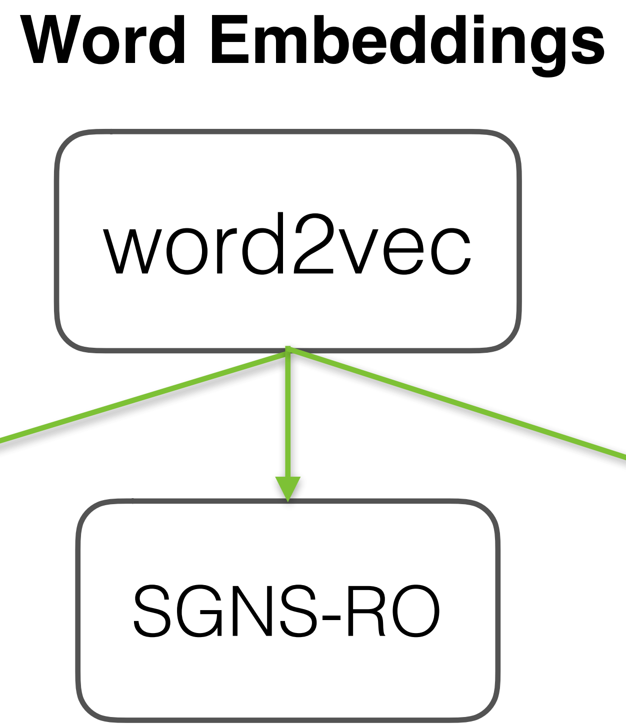
- 24 different categories of items
- Include ratings, reviews and other information
- «Cell Phones and Accessories» 194k reviews

Data preprocessing

Summary

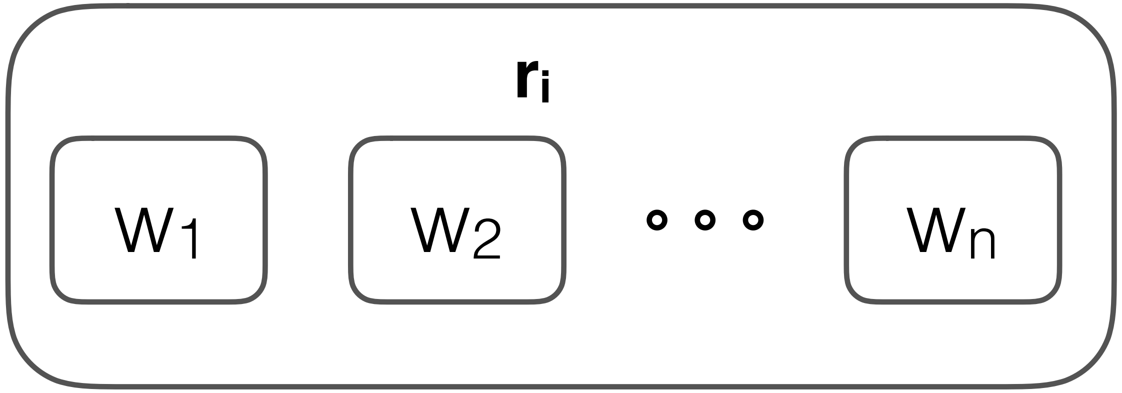
- vocabulary size: 3723 words
- sliding window size: 2
- two labels: positive/negative review

Feature extraction



Feature extraction

i-th review



$$r_i = \frac{\sum w_j}{n}$$

word2vec algorithms

SPPMI-SVD

Idea: find W and C using SVD decomposition of SPPMI matrix

Disadvantage: such approach doesn't lead to minimization of SGNS objective

SGNS-RO

Idea: optimize SGNS objective directly on the low-rank matrices space

Disadvantage: works in assumption of independence of **wc** values

word2vec algorithms

SGNS-EMF

Idea: explicitly factorize co-occurence matrix

Disadvantage: too many cycles

Algorithm 1: Alternating minimization for explicit matrix factorization

---

Input: Co-occurrence matrix $\mathbf{D}$, step-size of gradient descent η, maximum number of iterations K

Output: $\mathbf{C}_K$, $\mathbf{W}_K$

```
initialize C_i and W_i randomly, i = 1;
while i ≤ K do
    W_i = W_{i-1};
    //minimize over W;
    repeat
        W_i = W_i − η C_{i-1} (E_{D'|W_i,C_{i-1}} D' − D);
    until Convergence;
    C_i = C_{i-1};
    //minimize over C;
    repeat
        C_i = C_i − η (E_{D'|W_i,C} D' − D) W_i^T;
    until Convergence;
    i = i + 1;
```

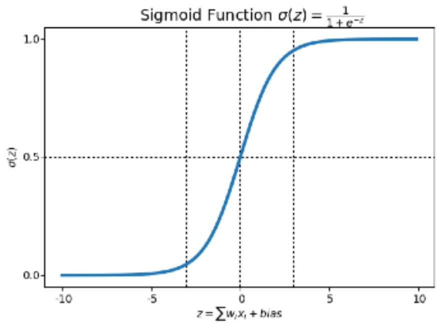
where the updates are

$$\mathbf{W}_i = \mathbf{W}_i - \eta\, \mathbf{C}_{i-1} \left(\mathbb{E}_{\mathbf{D}'|\mathbf{W}_i,\mathbf{C}_{i-1}} \mathbf{D}' - \mathbf{D}\right);$$

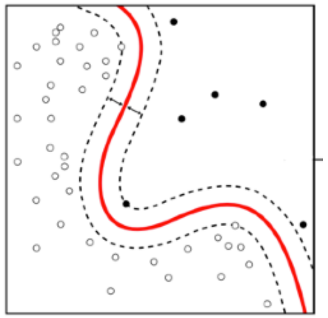
$$\mathbf{C}_i = \mathbf{C}_i - \eta \left(\mathbb{E}_{\mathbf{D}'|\mathbf{W}_i,\mathbf{C}} \mathbf{D}' - \mathbf{D}\right) \mathbf{W}_i^T;$$

**Li, et al., 2015, «Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective»*

Classification



Logistic Regression



SVM

Summary

- two simple classifiers were used «out of the box»

- classification metric: f1-score

Similarity test's results

Spearman's correlation between predicted similarities and the manually assessed ones (k = 5, alpha=0.5), simlex999 dataset

d=100	SVD-SPPMI	0.13284
	SGNS-RO	**0.13466**
	SGNS-EMF	0.03252
d=200	SVD-SPPMI	0.12277
	SGNS-RO	**0.13051**
	SGNS-EMF	0.06966
d=500	SVD-SPPMI	0.18781
	SGNS-RO	**0.18920**
	SGNS-EMF	0.06119

SGNS's objective function values

The values of SGNS objective function at the optimal point (all values are multiplied by 10^-9, k=5, alpha=0.5)

	SVD-SPPMI	SGNS-RO	SGNS-EMF
d=100	-0.2383	**-0.2321**	-0.3841
d=200	-0.2381	**-0.2316**	-0.5406
d=500	-0.2357	**-0.2300**	-0.8484

SGNS's objective function values

The values of SGNS objective function at the optimal point (all values are multiplied by 10^-9, d=200, alpha=0.5)

	SVD-SPPMI	SGNS-RO	SGNS-EMF
k=1	**-0.0758**	**-0.0742**	**-0.3467**
k=5	-0.2381	-0.2316	-0.5406
k=15	-0.6354	-0.6157	-0.6779

Classification results

F1-score values (k=5, alpha=0.5)

		LR	SVC
d=100	SVD-SPPMI	**0.87892**	**0.87901**
	SGNS-RO	0.87890	0.87888
	SGNS-EMF	0.86754	0.86849
d=200	SVD-SPPMI	0.88341	0.88338
	SGNS-RO	**0.88345**	**0.88341**
	SGNS-EMF	0.87446	0.87492
d=500	SVD-SPPMI	0.89012	0.89016
	SGNS-RO	**0.89019**	**0.89023**
	SGNS-EMF	0.88568	0.88580

Thank you for your attention!

Any questions?

https://github.com/Bullldoger/NLA-Project