Optimizing Statistical Distance Measures in Multivariate SVM for Sentiment Quantification

Kevin Labille and Susan Gauch
University of Arkansas, Arkansas, USA

Contact email: kclabill@uark.edu

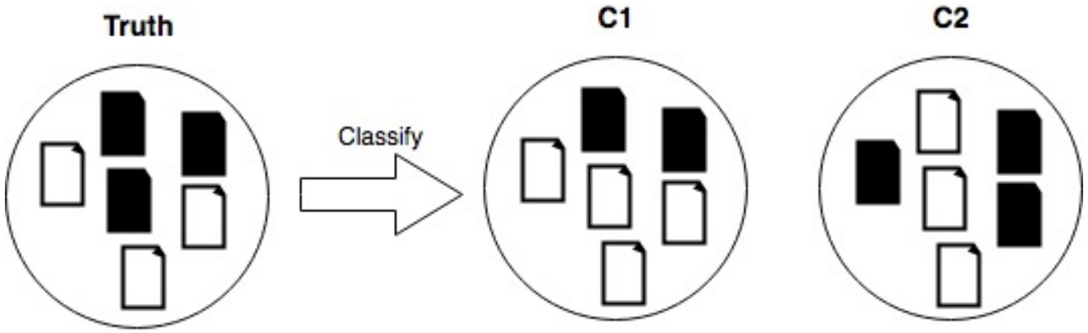
KEVIN LABILLE

Kevin Labille received his Ph.D in Computer Science in 2019 from the University of Arkansas under the supervision of Dr. Susan Gauch where he focused on text-mining and natural language processing. He then pursued a Post-doc with Dr. Xintao Wu (University of Arkansas) where his research were oriented towards dynamic recommender systems and fairness in machine learning.

Motivation

This paper focuses on sentiment quantification:

- A perfect classifier is a good quantifier
- A good classifier is not necessarily a good quantifier



- C1
  - false positive rate different than false negative rate
  - 5/6 correct
  - It is a good classifier, but poor quantifier
- C2
  - FPR = FNR
  - 2/6 correct
  - Perfect quantifier but poor classifier

Outline

- Introduction
- Related Work
- Quantifying Tweets
- Experimental Evaluation
- Results
- Conclusion

Introduction

- Introduction and background

Rapid growth of Internet
- e-commerce
- Online retailers

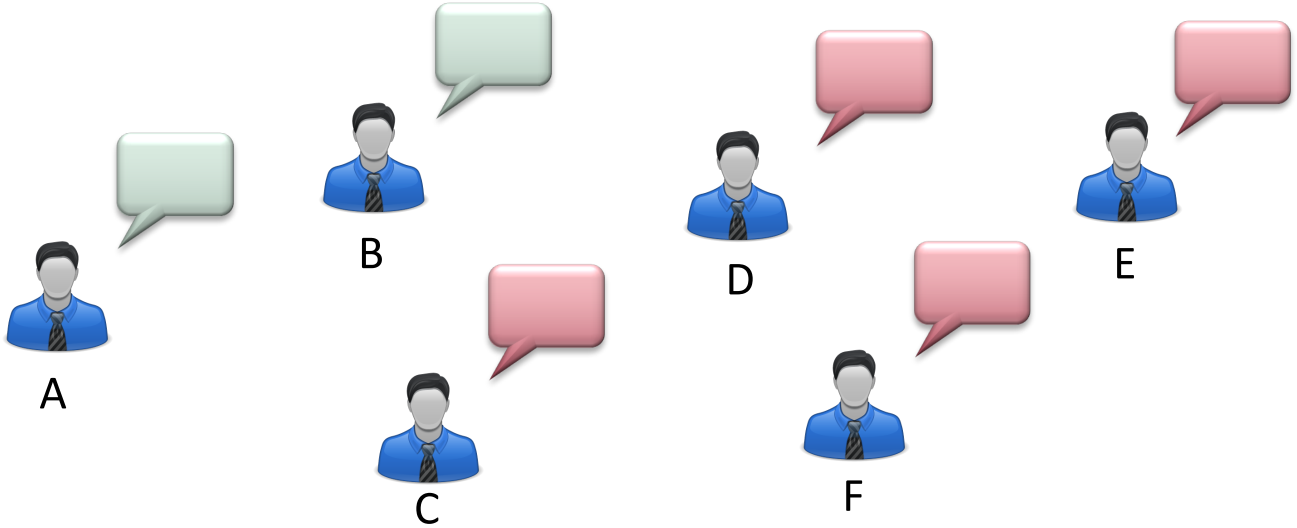
Increasing amount of online reviews

Opinions about: products, hotels, professionals…

Available data to study
- Identify trend
- Identify consensus

Introduction

" The only downside is the sound does not have a lot of bass, but honestly the quality of sound for the price is impressive "



Sentiment analysis

- The computational analysis of opinions in text
  - Who has a positive opinion? (A, B)
  - Who has a negative opinion? (C, D, E, F)

Sentiment quantification

- The estimation the proportion of document that belong to each polarity classes
  - How many have a positive opinion? (2)
  - How many have a negative opinion? (4)

Image from http://www.softicons.com

Related Work

Sentiment analysis in Twitter

- Go et al. [2009]
  - Compared SVM, Naïve Bayes classifier, and MaxEnt clssifier
  - MaxEnt performed better
  - POS tag not useful in Twitter sentiment classification
- Mohammad et al. [2013]
  - SVM classifier that uses sentiment lexicons as feature
  - Lexicons-related features improved accuracy by more than 8.5%
- Tang et al. [2014]
  - Word embedding combined with neural networks
  - Outperforms Mohammed et al. by 1.85%
- Labille et al. [2016]
  - Using information theory and probabilities for word sentiment scores

Sentiment quantification in Twitter

- Gao and Sebastiani [2015]
  - Pioneer work
  - Compare SVM(KLD) to traditional SVM
  - SVM(KLD) > traditional SVM
- Vilares et al. [2016]
  - Convolutional Neural Network to get hidden activation values
  - Train SVM(KLD) using these values
- Stojanovski et al. [2016]
  - Convolutional Neural Network + Gated Neural Network
  - Performances of CNN alone and GNN alone are very comparable
  - Outperformed Vilares et al.
- Mathieu Cliché [2017]
  - Used a deep-learning approach that uses both a Convolutional Neural Network (CNN) and a LSTM

Optimizing Statistical Distance Measures in Multivariate SVM for Sentiment Quantification

- This paper offers three contributions:
  - (1) We propose a new statistical method for building sentiment lexicons on short texts (tweets) that captures the polarity strength (score) and polarity orientation (sign) for both the **positive** and **negative components** of the words
  - (2) We use the paired-score sentiment lexicons to derive **new sentiment features** that better summarize the distribution of the positive and negative contributions of each word within the dataset
  - (3) Through a multivariate Support Vector Machine (SVM), we explore and compare numerous kernels that optimize various statistical distance measures to understand how they behave in a sentiment quantification task

Traditional sentiment lexicons vs paired-score sentiment lexicons

- Single sentiment score
  - One polarity strength
  - One polarity orientation
  - No information about how much *w* is positive and negative
    - 0.4 = 0.8 – 0.4
    - 0.4 = 0.5 – 0.1

wonky	-0.30
impossible	-0.72
assemble	0.01
bother	-0.47
…	…

- Paired sentiment score
  - Uses both the positive and negative distributions of the word
  - Catches more information than a single score
  - Could improve accuracy for quantifying

wonky	0.60	0.30
impossible	0.10	-0.82
assemble	0.02	0.01
bother	0.25	-0.72
…	…	

Quantifying Tweets: (1) Paired-score sentiment lexicons

– Sentiment scores are calculated using a probabilistic approach. We define the positivity of a word *w* as *pos(w)*,and its negativity as *neg(w):*

$$Pos(w) = \frac{pdf(w)}{N_{pos}} \times \frac{1}{df(w)}$$

$$Neg(w) = \frac{ndf(w)}{N_{neg}} \times \frac{1}{df(w)}$$

Where:

$$pdf(w) = \sum_{t \in T_{pos}} x \begin{cases} x = \frac{1}{|tweet|} & \text{if } w \in t \\ x = 0 & \text{otherwise} \end{cases}$$

$$ndf(w) = \sum_{t \in T_{neg}} x \begin{cases} x = \frac{1}{|tweet|} & \text{if } w \in t \\ x = 0 & \text{otherwise} \end{cases}$$

And:

$$df(w) = pdf(w) + ndf(w)$$

$$N_{pos} = \sum_{w \in vocab} pdf(w)$$

$$N_{neg} = \sum_{w \in vocab} ndf(w)$$

We then normalize both scores in the range [0.0, 1.0]

Quantifying Tweets: (2) Sentiment feature vectors

- *TF-IDF Bag of Words (tf-idf of the words computed from the training dataset)*
- We further derive additional numerical features that catch several **sentiment aspects** using the word's sentiment scores extracted from the paired-score lexicon

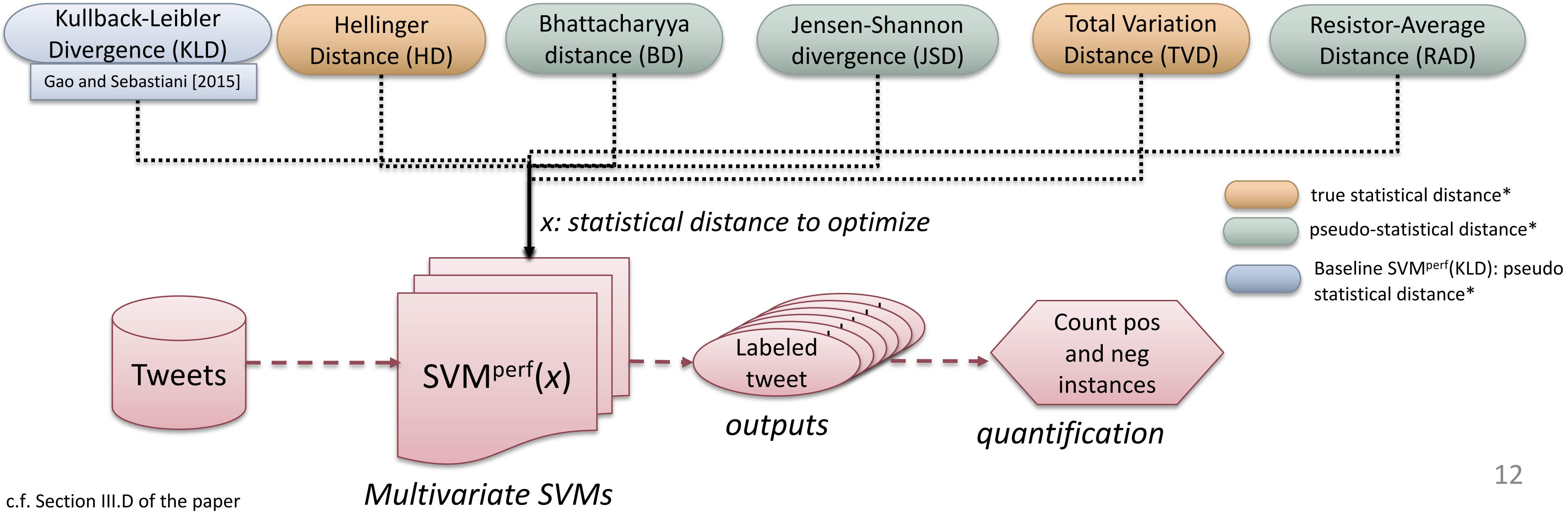
Each Tweet is therefore represented by the following features:

- BoW TF-IDF
- *Token found*: the number of words in the tweet that were found in the lexicon
- *token total*: the number of words in the tweet
- *max pos*: the maximum positive score in the tweet
- *min pos*: the minimum positive score in the tweet
- *max neg*: the maximum negative score in the tweet
- *min neg*: the minimum negative score in the tweet
- *ratio*: the ratio of *avg pos* over *avg neg*

Yielding a feature vector of size |vocabulary|+7 for each word

Quantifying Tweets: (3) Sentiment quantifier

- We use a Support Vector Machine (SVM) for multivariate performance measures (SVM^{perf})[T. Joachims, 2005] to optimize and compare several statistical distances
  - multivariate SVM allows the optimization of multivariate performance measures as opposed to univariate SVM



* c.f. Section III.D of the paper

Experimental Evaluation: datasets

- Sentiment analysis datasets[1]

Name	Train + dev			Test		
	# pos	# neg	# total	# pos	# neg	# total
SemEval2013 Task 2 A	4,215	1,798	6,013	1,475	559	2,034
SemEval2013 Task 9 A	4,215	1,798	6,013	982	202	1,184
SemEval2013 Task 10 A	4,215	1,798	6,013	1,038	365	1,403
SST	989	842	1,831	263	195	458
Sanders	418	54	872	101	118	219

- Sentiment quantification datasets [1,2]

Name	Train				Dev				Dev-test				Test			
	# topics	# pos	# neg	# total	# topics	# pos	# neg	# total	# topics	# pos	# neg	# total	# topics	# pos	# neg	# total
SemEval2016 task 4 D	60	2,841	582	3,423	20	778	279	1,057	20	893	216	1,109	100	8,212	2,339	10,551
SemEval2017 task 4 D	100	8,212	2,339	10,551	–	–	–	–	–	–	–	–	125	2,463	3,722	6,185

[1]*we ignore tweets that are labeled neutral for both training and testing*

[2]*we ignore the topics during the training phase while we test on each topic separately during the testing phase*

Experimental Evaluation: metrics and baselines

$\hat{p}$: *predicted distribution*
p: *true distribution*

- Metrics
  - Kullback-Leibler Divergence (KLD):

$$KLD(\hat{p}, p) = \sum_{c_i \in C} p(c_i) \cdot log \frac{p(c_i)}{\hat{p}(c_i)}$$

  - Mean Absolute Error (MAE):

$$MAE(\hat{p}, p) = \frac{1}{|C|} \sum_{c \in C} |\hat{p}(c) - p(c)|$$

  - Relative Absolute Error (RAE):

$$RAE(\hat{p}, p) = \frac{1}{|C|} \sum_{c \in C} \frac{|\hat{p}(c) - p(c)|}{p(c)}$$

- Baselines:
  - Univariate SVM with a linear kernel: classify each tweet then count the prevalence of both the positive and negative classes
  - Multivariate SVM: SVM^{perf} from T. Joachims (2005)
  - Multivariate SVM: SVM^{perf}(KLD) from Gao and Sebastiani (2015)

Results:
single scores vs paired scores

- Single score lexicons:
  - $Single\ score(w) = Pos(w) - Neg(w)$
  - Sentiment features derived from single score lexicon:
    - token found: the number of words in the tweet that were found in the lexicon
    - token total: the number of words in the tweet
    - max: the maximum score in the tweet
    - min: the minimum score in the tweet
    - avg: the average of the scores in the tweet
    - nb pos: the number of positive words in the tweet
    - nb neg: the number of negative words in the tweet
  - Single score feature vectors:
    - BoW TF-IDF + sentiment features
    - Size of feature vector: |vocabulary| + 7
- Methodology
  - Sentiment quantification using the baseline approach (Univariate SVM with linear kernel)

	Metrics	Single score lexicon	Paired score lexicon
SemEval2016	KLD	0.094	**0.090**
	AE	0.132	**0.130**
	RAE	1.269	**1.378**
SemEval2017	KLD	0.174	**0.138**
	AE	0.216	**0.188**
	RAE	2.972	**2.559**
Average	KLD	0.134	**0.114**
	AE	0.174	**0.159**
	RAE	2.121	**1.969**

Results

Results: sentiment quantification

- Comparison of the various multivariate SVMs against the baselines

	Metrics	univariate SVM	SVM(perf)	SVM(KLD)	SVM(HD)	SVM(BD)	SVM(JSD)	SVM(TVD)	SVM(RAD)
SST	KLD	0.031	0.011	0.036	0.005	0.044	0.046	**0.000**	0.030
	AE	0.124	0.148	0.266	0.100	0.295	0.301	0.028	0.245
	RAE	0.254	0.149	0.268	0.101	0.296	0.303	0.029	0.246
Sanders	KLD	0.000	0.004	0.010	0.001	0.007	0.028	**0.000**	0.007
	AE	0.005	0.088	0.138	0.037	0.115	0.230	0.005	0.115
	RAE	0.010	0.088	0.138	0.037	0.115	0.231	0.005	0.115
SemEval 2013 task A	KLD	0.003	0.046	0.019	**0.000**	0.018	0.024	0.003	0.019
	AE	0.032	0.275	0.194	0.006	0.191	0.219	0.080	0.194
	RAE	0.081	0.290	0.204	0.006	0.201	0.230	0.084	0.204
SemEval 2014 task A	KLD	0.011	0.018	0.022	**0.001**	0.020	0.026	0.001	0.022
	AE	0.059	0.171	0.204	0.040	0.197	0.222	0.045	0.204
	RAE	0.208	0.191	0.228	0.045	0.221	0.249	0.050	0.228
SemEval 2015 Task A	KLD	0.017	0.041	0.032	**0.001**	0.030	0.036	0.002	0.031
	AE	0.085	0.260	0.251	0.047	0.244	0.267	0.058	0.248
	RAE	0.220	0.276	0.266	0.050	0.259	0.283	0.061	0.263
SemEval 2016 Task D	KLD	0.090	0.069	**0.010**	0.013	0.014	0.011	0.018	0.014
	AE	0.130	0.242	0.098	0.111	0.108	0.098	0.136	0.106
	RAE	1.378	0.266	0.111	0.125	0.121	0.111	0.156	0.119
SemEval 2017 Task D	KLD	0.138	0.254	**0.024**	0.028	0.034	0.025	0.031	0.033
	AE	0.188	0.577	0.171	0.182	0.207	0.176	0.192	0.203
	RAE	2.559	0.676	0.200	0.213	0.241	0.205	0.225	0.237
Average	KLD	0.041	0.063	0.022	**0.007**	0.024	0.028	0.008	0.022
	AE	0.089	0.252	0.189	**0.075**	0.194	0.216	0.078	0.188
	RAE	0.673	0.276	0.202	**0.082**	0.208	0.230	0.087	0.202

Results: sentiment quantification

- Comparison of SVM(HD) against other sentiment quantification approaches
  - metric reported: KLD

	SST	Sanders	SemEval2013	SemEval2014	SemEval2015	SemEval2016	SemEval2017
SVM(HD)	**0.005**	**0.001**	**0.000**	**0.001**	**0.000**	0.013	0.028
SVM(KLD)[1]	0.036	0.010	0.019	0.022	0.032	**0.010**	**0.024**
SVM(KLD)[2]	0.011	**0.001**	0.029	0.033	0.076	-	-
Stojanovski et al.[2,3]	-	-	-	-	-	0.034	-
Mathieu Cliché [2,4]	-	-	-	-	-	-	0.036

[1] *Multivariate SVM with a KLD kernel using our approach.*

[2] *Results reported as per the authors in their respective papers. We did not reproduce their work.*

[3] *CNN combined with GNN*

[4] *CNN + LSTM*

Conclusion

- In this paper we have presented the following:
  - A new probabilistic approach to create a novel sentiment lexicon that captures and uses both the positivity and the negativity of words separately
  - We showed that such a lexicon can be used to derive sentiment features to model Tweets in the Vector Space Model
  - We showed that employing these feature vectors with a multivariate Support Vector Machine (SVM) that optimizes statistical distances metrics can improve sentiment quantification accuracy
  - Such a SVM machine achieves the good performances when optimizing the Hellinger Distance

Thank you!

- Kevin Labille -

kclabill@uark.edu