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.
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)
Related Work

Sentiment analysis in Twitter

- Go et al. [2009]
  - Compared SVM, Naïve Bayes classifier, and MaxEnt classifier
  - 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
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
    • $0.4 = 0.8 - 0.4$
    • $0.4 = 0.5 - 0.1$

• 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

<table>
<thead>
<tr>
<th></th>
<th>paired score</th>
<th>single score</th>
</tr>
</thead>
<tbody>
<tr>
<td>wonky</td>
<td>0.60</td>
<td>0.30</td>
</tr>
<tr>
<td>impossible</td>
<td>0.10</td>
<td>-0.82</td>
</tr>
<tr>
<td>assemble</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>bother</td>
<td>0.25</td>
<td>-0.72</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Quantifying Tweets: (1) Paired-score sentiment lexicons

- Sentiment scores are calculated using a probabilistic approach. We define the positivity of a word $w$ as $\text{pos}(w)$, and its negativity as $\text{neg}(w)$:

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

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

Where:

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

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

And:

$$\text{df}(w) = \text{pdf}(w) + \text{ndf}(w)$$

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

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

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

- Kullback-Leibler Divergence (KLD)
- Hellinger Distance (HD)
- Bhattacharyya distance (BD)
- Jensen-Shannon divergence (JSD)
- Total Variation Distance (TVD)
- Resistor-Average Distance (RAD)

**Multivariate SVMs**

* c.f. Section III.D of the paper
### Experimental Evaluation: datasets

- **Sentiment analysis datasets**
  
<table>
<thead>
<tr>
<th>Name</th>
<th>Train + dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># pos</td>
<td># neg</td>
</tr>
<tr>
<td>SemEval2013 Task 2 A</td>
<td>4,215</td>
<td>1,798</td>
</tr>
<tr>
<td>SemEval2013 Task 9 A</td>
<td>4,215</td>
<td>1,798</td>
</tr>
<tr>
<td>SemEval2013 Task 10 A</td>
<td>4,215</td>
<td>1,798</td>
</tr>
<tr>
<td>SST</td>
<td>989</td>
<td>842</td>
</tr>
<tr>
<td>Sanders</td>
<td>418</td>
<td>54</td>
</tr>
</tbody>
</table>

- **Sentiment quantification datasets**
  
<table>
<thead>
<tr>
<th>Name</th>
<th>Train</th>
<th>Dev</th>
<th>Dev-test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># topics</td>
<td># pos</td>
<td># neg</td>
<td># total</td>
</tr>
<tr>
<td>SemEval2016 task 4 D</td>
<td>60</td>
<td>2,841</td>
<td>582</td>
<td>3,423</td>
</tr>
<tr>
<td>SemEval2017 task 4 D</td>
<td>100</td>
<td>8,212</td>
<td>2,339</td>
<td>10,551</td>
</tr>
</tbody>
</table>

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

• 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\text{perf} from T. Joachims (2005)
  – Multivariate SVM: SVM\text{perf}(KLD) from Gao and Sebastiani (2015)
Results: single scores vs paired scores

- **Single score lexicons:**
  - *Single score* \( w = \text{Pos}(w) - \text{Neg}(w) \)
  - 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: \(|\text{vocabulary}| + 7\)

- **Methodology**
  - Sentiment quantification using the baseline approach (Univariate SVM with linear kernel)
Results: sentiment quantification

- Comparison of the various multivariate SVMs against the baselines

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Metrics</th>
<th>univariate SVM</th>
<th>SVM(perf)</th>
<th>SVM(KLD)</th>
<th>SVM(HD)</th>
<th>SVM(BD)</th>
<th>SVM(JSD)</th>
<th>SVM(TVD)</th>
<th>SVM(RAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST</td>
<td>KLD</td>
<td>0.031</td>
<td>0.011</td>
<td>0.036</td>
<td>0.005</td>
<td>0.044</td>
<td>0.046</td>
<td>0.000</td>
<td>0.030</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>0.124</td>
<td>0.148</td>
<td>0.266</td>
<td>0.100</td>
<td>0.295</td>
<td>0.301</td>
<td>0.028</td>
<td>0.245</td>
</tr>
<tr>
<td></td>
<td>RAE</td>
<td>0.254</td>
<td>0.149</td>
<td>0.268</td>
<td>0.101</td>
<td>0.296</td>
<td>0.303</td>
<td>0.029</td>
<td>0.246</td>
</tr>
<tr>
<td>Sanders</td>
<td>KLD</td>
<td>0.000</td>
<td>0.004</td>
<td>0.010</td>
<td>0.001</td>
<td>0.007</td>
<td>0.028</td>
<td>0.000</td>
<td>0.007</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>0.005</td>
<td>0.088</td>
<td>0.138</td>
<td>0.037</td>
<td>0.115</td>
<td>0.230</td>
<td>0.005</td>
<td>0.115</td>
</tr>
<tr>
<td></td>
<td>RAE</td>
<td>0.010</td>
<td>0.088</td>
<td>0.138</td>
<td>0.037</td>
<td>0.115</td>
<td>0.231</td>
<td>0.005</td>
<td>0.115</td>
</tr>
<tr>
<td>SemEval</td>
<td>KLD</td>
<td>0.003</td>
<td>0.046</td>
<td>0.019</td>
<td>0.000</td>
<td>0.018</td>
<td>0.024</td>
<td>0.003</td>
<td>0.019</td>
</tr>
<tr>
<td>2013</td>
<td>AE</td>
<td>0.032</td>
<td>0.275</td>
<td>0.194</td>
<td>0.006</td>
<td>0.191</td>
<td>0.219</td>
<td>0.080</td>
<td>0.194</td>
</tr>
<tr>
<td>task A</td>
<td>RAE</td>
<td>0.081</td>
<td>0.290</td>
<td>0.204</td>
<td>0.006</td>
<td>0.201</td>
<td>0.230</td>
<td>0.084</td>
<td>0.204</td>
</tr>
<tr>
<td>SemEval</td>
<td>KLD</td>
<td>0.011</td>
<td>0.018</td>
<td>0.022</td>
<td>0.001</td>
<td>0.020</td>
<td>0.026</td>
<td>0.001</td>
<td>0.022</td>
</tr>
<tr>
<td>2014</td>
<td>AE</td>
<td>0.059</td>
<td>0.171</td>
<td>0.204</td>
<td>0.040</td>
<td>0.197</td>
<td>0.222</td>
<td>0.045</td>
<td>0.204</td>
</tr>
<tr>
<td>task A</td>
<td>RAE</td>
<td>0.208</td>
<td>0.191</td>
<td>0.228</td>
<td>0.045</td>
<td>0.221</td>
<td>0.249</td>
<td>0.050</td>
<td>0.228</td>
</tr>
<tr>
<td>SemEval</td>
<td>KLD</td>
<td>0.017</td>
<td>0.041</td>
<td>0.032</td>
<td>0.001</td>
<td>0.030</td>
<td>0.036</td>
<td>0.002</td>
<td>0.031</td>
</tr>
<tr>
<td>2015</td>
<td>AE</td>
<td>0.085</td>
<td>0.260</td>
<td>0.251</td>
<td>0.047</td>
<td>0.244</td>
<td>0.267</td>
<td>0.058</td>
<td>0.248</td>
</tr>
<tr>
<td>Task A</td>
<td>RAE</td>
<td>0.220</td>
<td>0.276</td>
<td>0.266</td>
<td>0.050</td>
<td>0.259</td>
<td>0.283</td>
<td>0.061</td>
<td>0.263</td>
</tr>
<tr>
<td>SemEval</td>
<td>KLD</td>
<td>0.099</td>
<td>0.069</td>
<td>0.010</td>
<td>0.013</td>
<td>0.014</td>
<td>0.011</td>
<td>0.018</td>
<td>0.014</td>
</tr>
<tr>
<td>2016</td>
<td>AE</td>
<td>0.130</td>
<td>0.242</td>
<td>0.098</td>
<td>0.111</td>
<td>0.108</td>
<td>0.098</td>
<td>0.136</td>
<td>0.106</td>
</tr>
<tr>
<td>Task D</td>
<td>RAE</td>
<td>1.378</td>
<td>0.266</td>
<td>0.111</td>
<td>0.125</td>
<td>0.121</td>
<td>0.111</td>
<td>0.156</td>
<td>0.119</td>
</tr>
<tr>
<td>SemEval</td>
<td>KLD</td>
<td>0.138</td>
<td>0.254</td>
<td>0.024</td>
<td>0.028</td>
<td>0.034</td>
<td>0.025</td>
<td>0.031</td>
<td>0.033</td>
</tr>
<tr>
<td>2017</td>
<td>AE</td>
<td>0.188</td>
<td>0.577</td>
<td>0.171</td>
<td>0.182</td>
<td>0.207</td>
<td>0.176</td>
<td>0.192</td>
<td>0.203</td>
</tr>
<tr>
<td>Task D</td>
<td>RAE</td>
<td>2.559</td>
<td>0.676</td>
<td>0.200</td>
<td>0.213</td>
<td>0.241</td>
<td>0.205</td>
<td>0.225</td>
<td>0.237</td>
</tr>
<tr>
<td>Average</td>
<td>KLD</td>
<td>0.041</td>
<td>0.063</td>
<td>0.022</td>
<td>0.007</td>
<td>0.024</td>
<td>0.028</td>
<td>0.008</td>
<td>0.022</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>0.089</td>
<td>0.252</td>
<td>0.189</td>
<td>0.075</td>
<td>0.194</td>
<td>0.216</td>
<td>0.078</td>
<td>0.188</td>
</tr>
<tr>
<td></td>
<td>RAE</td>
<td>0.673</td>
<td>0.276</td>
<td>0.202</td>
<td>0.082</td>
<td>0.208</td>
<td>0.230</td>
<td>0.087</td>
<td>0.202</td>
</tr>
</tbody>
</table>
Results: sentiment quantification

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM(HD)</td>
<td>0.005</td>
<td>0.001</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>0.013</td>
<td>0.028</td>
</tr>
<tr>
<td>SVM(KLD)$^1$</td>
<td>0.036</td>
<td>0.010</td>
<td>0.019</td>
<td>0.022</td>
<td>0.032</td>
<td><strong>0.010</strong></td>
<td><strong>0.024</strong></td>
</tr>
<tr>
<td>SVM(KLD)$^2$</td>
<td>0.011</td>
<td><strong>0.001</strong></td>
<td>0.029</td>
<td>0.033</td>
<td>0.076</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Stojanovski et al.$^{2,3}$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.034</td>
<td>-</td>
</tr>
<tr>
<td>Mathieu Cliché$^{2,4}$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.036</td>
</tr>
</tbody>
</table>

$^1$ 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