Text Classification Using String Kernels

DD2434: Machine Learning, Advanced Course
Project Report

Group 36

Fabian Huss,
Jie Deng,
Mona Dadoun,
Þorsteinn Daði Gunnarsson

fhuss@kth.se jied@kth.se dadoun@kth.se tdgu@kth.se

Machine Learning
KTH Royal Institute of Technology
Stockholm, Sweden

January 16, 2017

Abstract

The scientific article "Text Classification Using String Kernels" [1] proposes a novel approach to classifying strings and documents with the help of a "String Subsequence Kernel" (SSK). The proposed method splits strings into subsequences as a means to further compare strings in different documents. The documents are then classified using a Support Vector Machine (SVM) and compared with the standardized "Word Kernel" (WK) as well as with the "n-grams kernel" (NGK). This report aims to reproduce the results achieved in the original article as well as trying to evaluate the legitimacy of the method proposed.

1 Introduction

The authors of the paper "Text Classification Using String Kernels" [1] addressed the problem with applying machine learning algorithms to discrete instances, for example strings, trees and graphs. They presented a novel approach for categorizing text documents by using an algorithm based on kernel method called String Sub-sequence Kernel (SSK). A sub-sequence is "any ordered sequence of k characters occurring in the text though not necessarily contiguously" [1]. SSK compares string similarities using a decay factor $\lambda \in (0, 1)$ for approximating the similarities between two sub-strings of length k. This decay factor depends upon how contiguous the strings are compared to each other. The paper describes as well how the inner product between strings can be efficiently evaluated by a dynamic programming technique in order to make the computations more efficient.

2 Kernel Methods

Kernel methods are a class of algorithms that only requires a similarity function $K(d_i, d_j)$ in contrast to other algorithms which transforms raw data sets into feature vector representations. The main idea of the Kernel Methods is embedding data into a high dimensional vector space and then looking for (linear) relations in that space. If the map was chosen suitably, complex relations could be simplified, and easily detected. The substring kernel are kernels comparing documents to find out how similar they are in the way that the more sub strings in common, the more similar the documents are. [1]

2.1 About the data set used Reuters

The text data set from Reuters we retrieved are with categories of "earn", "acq", "crude" and "corn". Documents in data sets are processed by removing stop words (i.e. words such as "and", "or", "him", "her" etc.) and punctuation. To relieve the concern of heavy computation of SSK kernel, each document has been shortened by extracting and only using the first 100 words. We set the size of subset of Reuters to 132 documents, in which 100 is training data and 32 is test data. Therefore, our data set is split in four categories with training sets and test sets accordingly: "earn" 25 (8); "acq" 25 (8); "crude" 25 (8); "corn" 25 (8).

2.2 Other Kernels

We will compare the performance of the SSK with varying lengths of the subtstring (n) for the n-grams (NGK) as well as a word kernel (WK). The documents for NGK and WK are preprocessed in the same way as done for SSK in order to get comparable results.

2.2.1 N-gram

N-grams maps documents into a high dimensional feature vector. Each entry in the vector represents the number of times a contiguous sub-sequence of length n appears in the document. The feature vector was then normalized using the same variant of $tfidf$ as described in the original paper. [1]

Category	F1 (SSK)	F1 (Approx. SSK)
earn	0.857	0.857
acq	0.800	0.696
crude	0.778	0.800
corn	0.667	0.667

Table 1: Comparing results for SSK and Approximate SSK ($\lambda = 0.5$, $k = 3$, $|S| = 200$)

2.2.2 Word kernel

A word kernel (WK) also maps documents into a high dimensional feature vector. Each entry represents the number of times a particular word appears in the document. The feature vector was also normalized using the same method as for n-grams.

2.3 String Sub-sequence Kernel

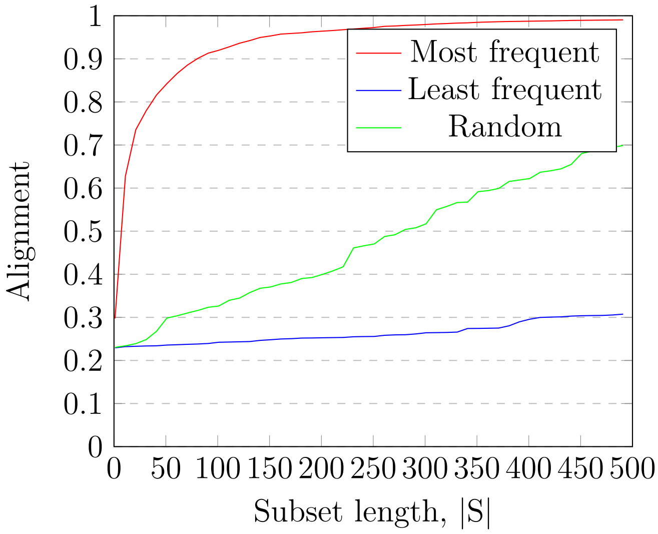
String sub-sequence kernel computes the similarity between documents like NGK and WK, but do not explicitly extract any features. The SSK requires a prohibiive amount of computations, and is according to the paper rewritten to a recursive algorithm. For more information we refere to definition 1 and 2 at the source [1].

2.4 Approximation of String Kernel

Since the SSK is heavy on computation an approximation method is needed to make it more efficient. Such a method is described in the paper [1] and works by using only a subset of all sub sequences that appear in the corpus to compute the kernel.

The approximation uses a subset S of all sub-strings present in the corpus. Choosing this subset can be done in multiple ways. We opted for choosing the most frequent subsequences as it does fairly good approximation for low values of $|S|$ data as shown in graph 2.4. More importantly the approximate kernel matrix gives similar results as the full one as seen in table 1 where F1 values for both the full SSK and approximate methods are shown.

Alignment of Kernel matrices for SSK vs Approximate SSK as length of the subset is increased.



2.5 Build Kernel Matrices and Classify

With the Reuters data set retrieved, approximated with fewer features, organized in proper data structure, and SSK kernel created, we then built up gram matrices. We used sklearn.svm.SVC to conduct the classification. With it "precomputed", gram matrix of training data is used to fit the model, and matrix for training and test data to predict the categories for test data.

3 Experiments and Results

Similarly to the paper we first ran a series of experiments varying a single variable while keeping the other ones constant and compared the methods. For the first set of experiments we looked into various values of k and $\lambda = 0.05$ for SSK and n for n-grams as can be seen table 2 and table 3. All the results were then compared including results for a WK. Secondly we ran the same experiment with a constant $k = 6$ but varying λ for SSK and compared to the previously best results of both the n-gram and WK methods. The Results for varying λ can be seen in table 4 and table 5.

All experiments were done with the approximation of SSK, with the 200 most frequent substrings, since the computation of the full SSK was expensive. As shown in section 2.4 this gives a good approximation of the results of the full SSK. The paper also performed some experiments with combining kernel methods but since they did not show any significant results we opted to focus on other things like the approximation instead. All calculations for the SSK in the results section were calculated averaging over 2 runs.

Category	Kernel	Length	F1		Precision		Recall	
			Mean	SD	Mean	SD	Mean	SD
earn	SSK	3	0.929	0.071	1.000	0.000	0.875	0.125
		4	0.877	0.058	0.785	0.091	1.000	0.000
		5	0.929	0.071	1.000	0.000	0.875	0.125
		6	**0.952**	0.067	1.000	0.000	0.917	0.118
		7	0.929	0.071	1.000	0.000	0.875	0.125
		8	0.929	0.071	1.000	0.000	0.875	0.125
		10	0.833	0.167	1.000	0.000	0.750	0.250
		12	0.851	0.082	1.000	0.000	0.750	0.125
		14	0.667	0.000	1.000	0.000	0.500	0.000
	NGK	3	0.500	0.500	0.500	0.500	0.500	0.500
		4	0.000	0.000	0.000	0.000	0.000	0.000
		5	0.000	0.000	0.000	0.000	0.000	0.000
		6	0.000	0.000	0.000	0.000	0.000	0.000
		7	0.000	0.000	0.000	0.000	0.000	0.000
		8	0.000	0.000	0.000	0.000	0.000	0.000
		10	0.000	0.000	0.000	0.000	0.000	0.000
		12	0.000	0.000	0.000	0.000	0.000	0.000
		14	0.000	0.000	0.000	0.000	0.000	0.000
	W-K		0.000	0.000	0.000	0.000	0.000	0.000
acq	SSK	3	0.785	0.090	0.704	0.171	0.938	0.062
		4	0.735	0.048	1.000	0.000	0.583	0.059
		5	0.882	0.059	0.833	0.056	0.938	0.125
		6	**0.917**	0.118	0.917	0.118	0.917	0.118
		7	0.833	0.167	0.857	0.143	0.812	0.188
		8	0.812	0.188	0.812	0.188	0.812	0.188
		10	0.670	0.030	0.527	0.056	0.938	0.062
		12	0.634	0.062	0.467	0.067	1.000	0.000
		14	0.516	0.000	0.348	0.000	1.000	0.000
	NGK	3	0.833	0.167	0.750	0.250	1.000	0.000
		4	0.833	0.167	0.750	0.250	1.000	0.000
		5	0.250	0.250	0.167	0.167	0.500	0.500
		6	0.200	0.200	0.125	0.125	0.500	0.500
		7	0.200	0.200	0.125	0.125	0.500	0.500
		8	0.200	0.200	0.125	0.125	0.500	0.500
		10	0.200	0.200	0.125	0.125	0.500	0.500
		12	0.200	0.200	0.125	0.125	0.500	0.500
		14	0.200	0.200	0.125	0.125	0.500	0.500
	W-K		0.633	0.233	0.500	0.224	0.900	0.300

Table 2: F1, Precision, and Recall for categories 'earn' and 'acq' using the three different kernels SSK, NGK, and W-K for different lengths of n. With a fixed $\lambda = 0.5$

Category	Kernel	Length	F1		Precision		Recall	
			Mean	SD	Mean	SD	Mean	SD
crude	SSK	3	0.821	0.021	0.792	0.065	0.875	0.125
		4	**0.906**	0.025	0.830	0.042	1.000	0.000
		5	0.825	0.063	0.708	0.092	1.000	0.000
		6	0.790	0.047	0.655	0.064	1.000	0.000
		7	0.729	0.033	0.574	0.041	1.000	0.000
		8	0.699	0.063	0.558	0.058	0.938	0.062
		10	0.734	0.109	0.676	0.051	0.812	0.188
		12	0.637	0.304	0.694	0.194	0.625	0.375
		14	0.400	0.000	1.000	0.000	0.250	0.000
	NGK	3	1.000	0.000	1.000	0.000	1.000	0.000
		4	0.833	0.167	0.750	0.250	1.000	0.000
		5	0.700	0.300	0.625	0.375	1.000	0.000
		6	0.200	0.200	0.125	0.125	0.500	0.500
		7	0.200	0.200	0.125	0.125	0.500	0.500
		8	0.200	0.200	0.125	0.125	0.500	0.500
		10	0.200	0.200	0.125	0.125	0.500	0.500
		12	0.200	0.200	0.125	0.125	0.500	0.500
		14	0.200	0.200	0.125	0.125	0.500	0.500
	W-K		0.967	0.100	0.950	0.150	1.000	0.000
corn	SSK	3	0.718	0.051	1.000	0.000	0.562	0.062
		4	**0.952**	0.067	1.000	0.000	0.917	0.118
		5	0.718	0.051	1.000	0.000	0.562	0.062
		6	0.701	0.048	1.000	0.000	0.542	0.059
		7	0.606	0.061	1.000	0.000	0.438	0.062
		8	0.606	0.061	1.000	0.000	0.438	0.062
		10	0.384	0.162	1.000	0.000	0.250	0.125
		12	0.384	0.162	1.000	0.000	0.250	0.125
		14	0.545	0.000	1.000	0.000	0.375	0.000
	NGK	3	1.000	0.000	1.000	0.000	1.000	0.000
		4	1.000	0.000	1.000	0.000	1.000	0.000
		5	0.000	0.000	0.000	0.000	0.000	0.000
		6	0.000	0.000	0.000	0.000	0.000	0.000
		7	0.000	0.000	0.000	0.000	0.000	0.000
		8	0.000	0.000	0.000	0.000	0.000	0.000
		10	0.000	0.000	0.000	0.000	0.000	0.000
		12	0.000	0.000	0.000	0.000	0.000	0.000
		14	0.000	0.000	0.000	0.000	0.000	0.000
	W-K		0.933	0.133	0.900	0.200	1.000	0.000

Table 3: F1, Precision, and Recall for categories ’crude’ and ’corn’ using the three different kernels SSK, NGK, and W-K for different lengths of n. With a fixed $\lambda = 0.5$

Category	Kernel	λ	F1		Precision		Recall	
			Mean	SD	Mean	SD	Mean	SD
earn	NGK		0.500	0.500	0.500	0.500	0.500	0.500
	SSK	0.01	0.929	0.071	1.000	0.000	0.875	0.125
		0.03	0.929	0.071	1.000	0.000	0.875	0.125
		0.05	0.929	0.071	1.000	0.000	0.875	0.125
		0.07	0.929	0.071	1.000	0.000	0.875	0.125
		0.09	0.844	0.044	0.829	0.029	0.875	0.125
		0.1	**0.967**	0.033	1.000	0.000	0.938	0.062
		0.3	0.895	0.038	1.000	0.000	0.812	0.062
		0.5	0.866	0.009	0.938	0.062	0.812	0.062
		0.7	0.933	0.000	1.000	0.000	0.875	0.000
		0.9	0.813	0.044	1.000	0.000	0.688	0.062
	W-K		0.000	0.000	0.000	0.000	0.000	0.000
acq	NGK		0.833	0.167	0.750	0.250	1.000	0.000
	SSK	0.01	0.853	0.147	0.833	0.167	0.875	0.125
		0.03	0.812	0.188	0.812	0.188	0.812	0.188
		0.05	0.833	0.167	0.800	0.200	0.875	0.125
		0.07	**0.853**	0.147	0.833	0.167	0.875	0.125
		0.09	0.641	0.026	0.650	0.150	0.750	0.250
		0.1	0.500	0.100	0.750	0.250	0.500	0.250
		0.3	0.733	0.067	0.679	0.179	0.875	0.125
		0.5	0.661	0.053	0.650	0.183	0.750	0.125
		0.7	0.536	0.010	0.727	0.273	0.500	0.125
		0.9	0.691	0.059	0.648	0.102	0.750	0.000
	W-K		0.633	0.233	0.500	0.224	0.900	0.300
crude	NGK		1.000	0.000	1.000	0.000	1.000	0.000
	SSK	0.01	0.754	0.088	0.633	0.094	0.938	0.062
		0.03	0.739	0.103	0.614	0.114	0.938	0.062
		0.05	0.721	0.121	0.614	0.114	0.875	0.125
		0.07	**0.754**	0.088	0.633	0.094	0.938	0.062
		0.09	0.517	0.183	0.542	0.042	0.562	0.312
		0.1	0.614	0.026	0.513	0.042	0.812	0.188
		0.3	0.675	0.039	0.667	0.167	0.750	0.125
		0.5	0.628	0.012	0.635	0.165	0.750	0.250
		0.7	0.598	0.042	0.485	0.015	0.812	0.188
		0.9	0.618	0.018	0.500	0.000	0.812	0.062
	W-K		0.967	0.100	0.950	0.150	1.000	0.000

Table 4: F1, Precision, and Recall for categories 'earn', 'acq', and 'crude' using the three different kernels SSK, NGK, and W-K for different values of λ with a fixed $n = 6$

Category	Kernel	λ	F1		Precision		Recall	
			Mean	SD	Mean	SD	Mean	SD
corn	NGK		1.000	0.000	1.000	0.000	1.000	0.000
	SSK	0.01	0.718	0.051	1.000	0.000	0.562	0.062
		0.03	0.718	0.051	1.000	0.000	0.562	0.062
		0.05	0.718	0.051	1.000	0.000	0.562	0.062
		0.07	0.718	0.051	1.000	0.000	0.562	0.062
		0.09	0.615	0.000	0.800	0.000	0.500	0.000
		0.1	**0.718**	0.051	1.000	0.000	0.562	0.062
		0.3	0.667	0.000	1.000	0.000	0.500	0.000
		0.5	0.606	0.061	1.000	0.000	0.438	0.062
		0.7	0.718	0.051	1.000	0.000	0.562	0.062
		0.9	0.583	0.083	0.875	0.125	0.438	0.062
	W-K		0.933	0.133	0.900	0.200	1.000	0.000

Table 5: F1, Precision, and Recall for categories ’corn’ using the three different kernels SSK, NGK, and W-K for different values of λ with a fixed $n = 6$

4 Discussion

Different results has been retrieved than shown in the original paper since we used a shorter document length than the Reuters dataset provided and we used smaller training and test sets. The results are never the less interesting since they show that the SSK method seems to outperform the WK and n-grams over all. WK does especially poorly in some cases since they don’t have as many features to work with in such short documents. Possibly because the vocabulary of the documents is too general but does better for classes with a more distinct vocabulary or specialized words. The n-grams however have more to work with as they consider substrings similarly to the SSK but do still not show as good results as the SSK method. While, as the original paper shows and concludes, that SSK does not give a significant advantage for longer documents compared to n-grams or WK it outperforms both in smaller documents. A significant downside to the SSK method is how slow it is, and even with the approximation it is still much slower than both WK and n-grams. Another thing that gives WK and NGK some advantage is that they produce feature vectors that can thus both be linearly classified or used in combination with a kernel method such as Polynomial or Radial basis function kernels.

Since we are extracting only the first 100 words from each document, we get results differing from the original paper. In order to make the computations efficient we had to make all the optimizations named in the paper as well as only taking the 100 words. This results are made on incomplete documnets, which makes them more dissimilar.

References

[1] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text classification using string kernels,” *Journal of Machine Learning Research*, vol. 2, no. Feb, pp. 419–444, 2002.