Nyström Methods for Efficient Kernel-Based Methods for Community Question Answering

Danilo Croce[1], **Simone Filice**[2] and **Roberto Basili**[1]
[1] Dept. of Enterprise Engineering
[2] Dept. of Civil Engineering and Computer Science Engineering
University of Roma, Tor Vergata, Italy
{croce,filice,basili}@info.uniroma2.it

Abstract

English. Expressive but complex kernel functions, such as Sequence or Tree kernels, are usually underemployed in NLP tasks, e.g., in community Question Answering (cQA), as for their significant complexity in both learning and classification stages. Recently, the Nyström methodology for data embedding has been proposed as a viable solution to scalability problems. By mapping data into low-dimensional approximations of kernel spaces, it positively increases scalability through compact linear representations for highly structured data. In this paper, we show that Nyström methodology can be effectively used to apply a kernel-based method in the cQA task, achieving state-of-the-art results by reducing the computational cost of orders of magnitude.

Italiano. *Metodi di apprendimento automatico basato su funzioni kernel complesse, come Sequence o Tree Kernel, rischiano di non poter essere adeguatamente utilizzati in problemi legati all'elaborazione del linguaggio naturale (come ad esempio in Community Question Answering) a causa degli alti costi computazionali per l'addestramento e la classificazione. Recentemente é stata proposta una metodologia, basata sul metodo di Nyström, per poter far fronte a questi problemi di scalabilitá: essa permette di proiettare gli esempi, osservabili in fase di addestramento e classificazione, all'interno di spazi a bassa dimensionalitá che approssimano lo spazio sottostante la funzione kernel. Queste rappresentazioni compatte permettono di applicare algoritmi di apprendimento automatico estremamente efficienti e scalabili. In questo lavoro si dimostra che é possibile applicare metodi kernel al problema di Community Question Answering, ottenendo risultati che sono lo stato dell'arte, riducendo di ordini di grandezza i costi computazionali.*

1 Introduction

Kernel methods (Shawe-Taylor and Cristianini, 2004) have been employed in several Machine Learning algorithms (Crammer et al., 2006; Vapnik, 1998) achieving state-of-the-art performances in many classification tasks. Recently, the kernel based approach presented in (Filice et al., 2016) has been applied in the community Question Answering (cQA) challenge at SemEval 2016 (Nakov et al., 2016) obtaining state-of-the-art results.

Unfortunately, when large data volumes are involved, time and space complexity required in learning and classification may prevent the adoption of expressive but complex kernel functions, such as Sequence (Cancedda et al., 2003) or Tree kernels (Collins and Duffy, 2001). In particular, the classification cost required by a kernel-based model crucially depends on its number of support vectors: classifying a new instance requires a kernel computation against all support vectors. This scalability issue is evident in many NLP and IR applications, such as in re-ranking answers in question answering (Moschitti et al., 2007; Severyn et al., 2013; Filice et al., 2016), where the number of support vectors is typically very large.

Some approaches have been defined to bound the complexity of kernel-based methods, such as (Wang and Vucetic, 2010; Vedaldi and Zisserman, 2012; Filice et al., 2014), but they are still specific to kernel formulations and learning algorithms.

In (Croce and Basili, 2016) it has been shown that a viable and more general solution to the above scalability issues is the Nyström methodology, a dimensionality reduction technique that has been applied also in kernel-based methods since (Williams and Seeger, 2001). This methodology has been designed to approximate the Gram Matrix derivable by a kernel function, enabling the projections of examples into low-dimensional spaces. The Nyström projection function is generated by using some examples called *landmarks*, whose number directly impacts on the embeddings quality; dually, costs of projecting a new example in the embedding space rise linearly with the number of landmarks, that is usually of orders of magnitude lower with respect of the number of possible support vectors that can be derived from a learning process. Once each example is projected in the dense low-dimensional space, the application of efficient linear learning methods is enabled, such as (Hsieh et al., 2008), preserving at the same time the expressiveness and effectiveness of kernel methods. This approach is highly applicable to different input data as well as to different kernels or learning algorithms, as discussed in (Croce and Basili, 2016).

In this paper we show that the Nyström method can be effectively used in the cQA task, by adopting the same kernel functions proposed in (Filice et al., 2016) and obtaining the same results w.r.t. the metrics adopted in the SemEval task, by reducing the computational cost of orders of magnitude.

In Section 2, we demonstrate the viability of the Nyström method to reduce the computational costs of kernel machines. Experimental results (obtained by adopting efficient SVM learning over the cQA task) are discussed in Section 3. Finally, Section 4 describes related work, while in Section 5 conclusions are derived.

2 Linearizing linguistic properties through Nyström Approach

Given an input training dataset $o_i \in \mathcal{D}$, a kernel function $K(o_i, o_j)$ is a similarity function that corresponds to a dot product in the implicit kernel space, i.e., $K(o_i, o_j) = \Phi(o_i) \cdot \Phi(o_j)$. The advantage of kernels is that the projection function $\Phi(o_i) = \vec{x}_i \in \mathbb{R}^n$ is never explicitly computed (Shawe-Taylor and Cristianini, 2004). In fact, this operation may be prohibitive when the dimensionality n of the underlying kernel space is extremely large. For example, Tree Kernels (Collins and Duffy, 2001) give rise to spaces whose number of dimensions is proportional to the number of possible sub-trees in a Natural Language. Kernel functions are exploited by kernel-based learning algorithms, such as SVM (Vapnik, 1998), to operate on the implicit kernel space without its explicit definition.

Let us assume that, given a kernel K, its explicit projection function ϕ over $\mathcal{D}$ is available to derive new representations $\vec{x}_i$ being the rows of the resulting matrix X. We define the Gram Matrix as $G = XX^\top$, with each single element corresponding to $G_{ij} = \Phi(o_i)\Phi(o_j) = K(o_i, o_j)$. The aim of the Nyström method is to derive a new low-dimensional embedding in a l-dimensional space, with $l \ll n$ so that $G \approx \tilde{G} = \tilde{X}\tilde{X}^\top$. This is obtained by generating an approximation of G using a subset of l columns of the matrix. This corresponds to selecting a subset L of the available examples, called *landmarks*. Suppose we randomly sample l columns of G, and let C be the $n \times l$ matrix of these sampled columns. Then, we can rearrange the columns and rows of G and define $X = [X_1 \;\; X_2]$ such that:

$$G = XX^\top = \begin{bmatrix} W & X_1^\top X_2 \\ X_2^\top X_1 & X_2^\top X_2 \end{bmatrix}$$
$$\text{and} \quad C = \begin{bmatrix} W \\ X_2^\top X_1 \end{bmatrix} \qquad (1)$$

where $W = X_1^\top X_1$, i.e., the subset of G that only considers landmarks. The Nyström approximation can be defined as:

$$G \approx \tilde{G} = CW^\dagger C^\top \qquad (2)$$

where $W^\dagger$ denotes the Moore-Penrose inverse of W. The Singular Value Decomposition (SVD) is used to obtain $W^\dagger$ as it follows. First W is decomposed so that $W = USV^\top$ where U and V are both orthogonal matrices, and S is a diagonal matrix containing the (non-zero) singular values of W on its diagonal. Since W is symmetric and positive definite $W = USU^\top$. Then $W^\dagger = US^{-1}U^\top = US^{-\frac{1}{2}}S^{-\frac{1}{2}}U^\top$ and the Equation 2 can be rewritten as

$$\begin{aligned} G \approx \tilde{G} &= CUS^{-\frac{1}{2}}S^{-\frac{1}{2}}U^\top C^\top \\ &= (CUS^{-\frac{1}{2}})(CUS^{-\frac{1}{2}})^\top = \tilde{X}\tilde{X}^\top \end{aligned} \qquad (3)$$

Given an input example $o_i \in \mathcal{D}$, a new low-dimensional representation $\tilde{\vec{x}}_i$ can be thus determined by considering the corresponding i-th item of C as

$$\tilde{\vec{x}}_i = \Theta(o_i) = \vec{c}_i US^{-\frac{1}{2}} \qquad (4)$$

where $\vec{c_i}$ corresponds to a vector whose dimensions contain the evaluation of the kernel function between o_i and each landmark $o_j \in L$. The method produces l-dimensional vectors, and no restriction is applied to the input dataset as long as a valid $K(o_i, o_j)$ is used.

Several policies have been defined to determine the best selection of landmarks to reduce the Gram Matrix approximation error. In this work the uniform sampling without replacement is adopted, as suggested by (Kumar et al., 2012), where this policy has been theoretically and empirically shown to achieve results comparable with other (more complex) selection policies.

Assuming that k is the computational cost[1] of a single kernel operation, the runtime cost of the Nyström method is $\mathcal{O}(knl + l^3 + nl^2)$ as it depends on (*i*) the computation of the $n \times l$ matrix C, i.e., $\mathcal{O}(knl)$; (*ii*) the SVD evaluation on W, which is $\mathcal{O}(l^3)$; and (*iii*) the projection of the entire dataset through the multiplication by C, i.e., $\mathcal{O}(nl^2)$. For several classes of kernels, such as Tree or Sequence Kernels (Collins and Duffy, 2001), the kernel computation cost is extremely high. Therefore, the computational cost for the construction of the matrix C dominates the overall expression.

Once an example is projected in the l-dimensional space, efficient and large-scale learning algorithm can be applied. To further control the computational cost of the training step, we addressed a class of algorithms that bounds the number of times a single instance is re-used during training. In particular, we investigated the Dual Coordinate Descent algorithm (Hsieh et al., 2008): it is a batch learning algorithm whose achievable accuracy is made inversely dependent on the number of iterations T over a training dataset. Its training time cost on a dataset of n examples in $\mathbb{R}^l$ is $\mathcal{O}(Tnl)$. Being fixed the number of iterations required to obtain an accurate model[2], such cost is negligible w.r.t. the projection cost. Therefore, a complete training process exploiting the Nyström method is simply $\mathcal{O}\big(kln\big)$, that should be compared with a traditional kernel-based SVM learning algorithm, e.g., (Chang and Lin, 2011), whose computational cost is almost $\mathcal{O}\big(kn^2\big)$, with $l \ll n$.

The computational cost of a classification step only depends on the projection of the example in the new space, i.e., $\mathcal{O}\big(kl\big)$. In fact, once a test example is projected, the final decision requires a dot product between the low-dimensional representation $\tilde{\vec{x}}_i$ and the hyperplane underlying the classification function: again, this is negligible with respect to the cost of the single kernel operations. Such cost is typically extremely lower than the cost of a pure kernel-based classification, which requires a kernel operation againts all the support vectors selected during the training process, which is usually far larger than the number of landmarks.

3 Empirical Investigation: the Community QA task

The proposed stratified Nyström method has been applied in the SemEval-2016 community Question Answering (cQA) task. In this task, participants are asked to automatically provide good answers in a community question answering setting (Nakov et al., 2016).

In particular, we focused on the Subtask A: given a question and a large collection of question-comment threads created by a user community, the task consists in (re-)ranking comments that are most useful for answering the question. This task is interesting as kernel methods achieved the highest results in the cQA task, as demonstrated by the KeLP team (Filice et al., 2016). In particular, Subtask A is modeled as a binary classification problem, where examples are generated by considering (question,comment) pairs. Each pair generates an example for a binary SVM, where the positive label is associated with a *good* comment and the negative label includes the *potential* and *bad* comments. The classification score is used to sort the instances and produce the final ranking. According to the above setting, a train and test dataset made of 20,340 and 3,270 examples are generated. In (Filice et al., 2016), a Kernel-based SVM classifier achieved state-of-the-art results by adopting a kernel combination that exploited (*i*) feature vectors containing linguistic similarities between the texts in a pair; (*ii*) shallow syntactic trees that encode the lexical and morphosyntactic information shared between text pairs; (*iii*) feature vectors capturing task-specific information.

First, a batch kernel-based SVM (Chang and Lin, 2011) learning algorithm operating on the kernel function proposed in (Filice et al., 2016) is adopted to determine the upper bound in terms

[1] Expressed in terms of basic operations, such as products.

[2] In (Croce and Basili, 2016) a number of iterations $T = 30$ obtained stable and accurate results in several tasks.

of classification quality (but with higher computational costs). Then, multiple standard Nyström methods are used to linearize the dataset by sampling different numbers of landmarks: 10 configurations have been investigated by starting from 100 landmarks and incrementally adding 100 landmarks at a time. The higher is the number of used landmarks, the higher is the quality of the approximated low-dimensional space (Drineas and Mahoney, 2005), but the higher is also the computational cost. The most complex projection function is thus based on 1,000 landmarks. Landmarks have been selected by applying a random selection without replacement, as suggested in (Kumar et al., 2012). An efficient linear SVM (Hsieh et al., 2008) is adopted on the resulting embedding space. Experiments have been carried out by using the KeLP framework[3] (Filice et al., 2015a).

Results are reported in Table 1 in terms of Mean Average Precision (MAP, the official rank of the competition), F_1 on the *good* class, and computational saving, i.e., percentage of avoided kernel operations in classification. The standard SVM model contains 11,322 Support Vectors, thus requiring more than 37M kernel operations for the complete classification of the 3,270 test instances[4]. By adopting the Nyström methodology with only 1,000 landmarks the same F_1 score (i.e., 64.4) is obtained. Moreover, a comparable MAP (i.e., 78.2%) achieved by the KeLP team is replicated with a 91.2% of saving. The speed up is impressive also when fewer landmarks are used: with 300 landmarks, 77.7 MAP is obtained by saving more that 97% of kernel computations. These results are straightforward, considering that results comparable with the state-of-the-art can be obtained by reducing of almost two orders of magnitude the computational costs. Overall, the MAP obtained by the proposed approach is still higher than the one achieved by all the other systems of the challenge, including ConvKN (Barrón-Cedeño et al., 2016) and SemanticZ (Mihaylov and Nakov, 2016), i.e., the second and third best systems, respectively.

4 Related Work

Improving the efficiency of kernel-based methods is a largely studied topic. The reduction of computational costs has been early designed by imposing a budget in the number of support vectors (Cesa-Bianchi and Gentile, 2006; Dekel and Singer, 2006; Orabona et al., 2008; Wang and Vucetic, 2010; Filice et al., 2014). However, in complicated tasks, such methods still require large budgets that systematically rely on many kernel computations. They are thus less efficient than Nyström: a classifier based on the Nyström method with l landmarks has approximately the same computational complexity of its budgeted counterpart with a budget set to l, but its accuracy is typically higher, as shown in (Croce and Basili, 2016). Alternatively, Zanzotto and Dell'Arciprete (2012) proposed Distributed Tree Kernels that approximate tree kernels (Collins and Duffy, 2001) through the explicit mapping of trees into vectors. DTKs focus on specific tree kernel functions, while the approach proposed here can be effectively applied to any kernel function. An alternative strategy is presented in (Filice et al., 2015b), where a cascade of kernel-based classifiers is proposed according to the computational cost of their kernel functions, so that more complex classifiers are invoked only on difficult instances. Their solution is strictly connected to the availability of multiple kernels that have to be sorted according to their complexity and expressiveness. Usually, it is hard to define many kernels for a given task, and consequently only few layers can be set.

Table 1: Results in CQA. Upperbound is achieved by a SVM with more than 37M kernel operations.

Landmarks	MAP	F1	Saving
100	76.0	58.6	99.1%
200	77.0	60.8	98.2%
300	77.5	62.2	97.4%
400	77.7	62.4	96.5%
500	77.9	63.1	95.6%
600	78.0	63.6	94.7%
700	78.1	63.7	93.8%
800	78.0	63.8	92.9%
900	78.1	64.2	92.1%
1000	78.2	64.4	91.2%
standard SVM	79.2	64.4	-
ConvKN	77.7	66.2	-
SemanticZ	77.6	61.8	-

5 Conclusion

This paper discussed the application of Nyström method for a significant reduction of computa-

[3] `https://github.com/SAG-KeLP`

[4] The classification time was more than 3 hours and a half on a standard machine with 4 cores i7-2600 3.4 GHz.

tional costs in kernel-based classifications in the cQA task. By projecting examples into low-dimensional embeddings, Nyström enables the adoption of efficient linear classifier, and drastically reduces the overall computational cost. Experimental results demonstrate that the proposed approach leads to a cost reduction higher than 90%, with a negligible performance drop. Future research will be devoted to the definition of a principled strategy to estimate the optimal number of layers, as well as the size of embeddings at each layer.

References

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad Al-Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 2016. Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 896–903, San Diego, California, June. Association for Computational Linguistics.

Nicola Cancedda, Éric Gaussier, Cyril Goutte, and Jean-Michel Renders. 2003. Word-sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Nicolo' Cesa-Bianchi and Claudio Gentile. 2006. Tracking the best hyperplane with a simple budget perceptron. In *In proc. of the nineteenth annual conference on Computational Learning Theory*, pages 483–498. Springer-Verlag.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS'2001)*, pages 625–632.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, December.

Danilo Croce and Roberto Basili. 2016. Large-scale kernel-based language learning through the ensemble nystrom methods. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 100–112.

Ofer Dekel and Yoram Singer. 2006. Support vector machines on a budget. In Bernhard Schlkopf, John Platt, and Thomas Hoffman, editors, *NIPS*, pages 345–352. MIT Press.

Petros Drineas and Michael W. Mahoney. 2005. On the nystrm method for approximating a gram matrix for improved kernel-based learning. *Journal of ML Research*, 6.

Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2014. Effective kernelized online learning in language processing tasks. In *Proceedings of ECIR 2014*, pages 347–358.

Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2015a. Kelp: a kernel-based learning platform for natural language processing. In *Proceedings of ACL: System Demonstrations*, Beijing, China, July.

Simone Filice, Danilo Croce, and Roberto Basili. 2015b. A Stratified Strategy for Efficient Kernel-based Learning. In *AAAI Conference on Artificial Intelligence*.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1116–1123, San Diego, California, June. Association for Computational Linguistics.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. 2008. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the ICML 2008*, pages 408–415, New York, NY, USA. ACM.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. 2012. Sampling methods for the nyström method. *J. Mach. Learn. Res.*, 13:981–1006, April.

Todor Mihaylov and Preslav Nakov. 2016. Semanticz at semeval-2016 task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 879–886, San Diego, California, June. Association for Computational Linguistics.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *Proceedings of ACL'07*.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Francesco Orabona, Joseph Keshet, and Barbara Caputo. 2008. The projectron: a bounded kernel-based perceptron. In *Proceedings of ICML '08*, pages 720–727, USA. ACM.

Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. 2013. Building structures from classifiers for passage reranking. In *Proceedings of the 22nd ACM international Conference on Information and Knowledge Management*, CIKM '13, pages 969–978, New York, NY, USA. ACM.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Andrea Vedaldi and Andrew Zisserman. 2012. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3).

Zhuang Wang and Slobodan Vucetic. 2010. Online passive-aggressive algorithms on a budget. *Journal of Machine Learning Research - Proceedings Track*, 9:908–915.

Christopher K. I. Williams and Matthias Seeger. 2001. Using the nyström method to speed up kernel machines. In *Proceedings of NIPS 2000*.

Fabio Massimo Zanzotto and Lorenzo Dell'Arciprete. 2012. Distributed tree kernels. In *Proceedings of ICML 2012*.