ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Advances in Learning with Kernels: Theory and Practice in a World of growing Constraints

Luca Oneto[1], Nicolò Navarin[2], Michele Donini[2],
Fabio Aiolli[2], and Davide Anguita[3]

1 - DITEN - University of Genova
Via Opera Pia 11A, I-16145 Genova - Italy

2 - Department of Mathematics - University of Padua
Via Trieste, 63, I-35121 Padova - Italy

3 - DIBRIS - University of Genova
Via Opera Pia 13, I-16145 Genova - Italy

Abstract. Kernel methods consistently outperformed previous generations of learning techniques. They provide a flexible and expressive learning framework that has been successfully applied to a wide range of real world problems but, recently, novel algorithms, such as Deep Neural Networks and Ensemble Methods, have increased their competitiveness against them. Due to the current data growth in size, heterogeneity and structure, the new generation of algorithms are expected to solve increasingly challenging problems. This must be done under growing constraints such as computational resources, memory budget and energy consumption. For these reasons, new ideas have to come up in the field of kernel learning, such as deeper kernels and novel algorithms, to fill the gap that now exists with the most recent learning paradigms. The purpose of this special session is to highlight recent advances in learning with kernels. In particular, this session welcomes contributions toward the solution of the weaknesses (e.g. scalability, computational efficiency and too shallow kernels) and the improvement of the strengths (e.g. the ability of dealing with structural data) of the state of the art kernel methods. We also encourage the submission of new theoretical results in the Statistical Learning Theory framework and innovative solutions to real world problems.

1 Introduction

Kernel methods are a family of machine learning algorithms and they represent the solution in terms of pairwise similarity between input examples and do not work on an explicit representation of the examples [1]. This function for computing similarity has to be a kernel function. Given a set X and a function $K : X \times X \to \mathbb{R}$, we say that K is a *kernel* on $X \times X$ if K is symmetric and positive-semidefinite. It is easy to see that if each $x \in X$ can be represented as $\phi(x) = \{\phi_n(x)\}_{n \geq 1}$ such that the value returned by K is the ordinary dot product $K(x, y) = \langle \phi(x), \phi(y) \rangle = \sum_n \phi_n(x)\phi_n(y)$ then K is a kernel. If X is a countable set, the converse is also always true. The vector space induced by ϕ is called the *feature space*. It is easy to show that kernels are closed under positive weighted summation. This simple property is exploited in the Multiple Kernel

Learning algorithms [2, 3] where a new kernel is created by using a weighted sum of *base* kernels with positive coefficients. This is only an example and kernels have other important properties that justify their success in the machine learning community [4].

Kernel learning algorithms search for (linear) relations in the *feature space*, where it is more likely for examples to be linearly separable [5]. In these methods, the learning algorithm is formulated as an optimization problem that, if the adopted function is a kernel, is convex and has a global minimum.

Kernel methods have been broadly studied in the last few years, from a theoretical point of view [6, 7, 8, 9, 10], and also for the wide range of applications they have been applied to because of their computational efficiency and the high predictive performance they are able to reach. Recently, novel algorithms, such as Deep Neural Networks and Ensemble Methods, have increased their competitiveness against them. Due to the current data growth in size, heterogeneity and structure, the new generation of algorithms are expected to solve increasingly challenging problems. This must be done under growing constraints such as computational resources, memory budget and energy consumption. For these reasons, new ideas have to come up in the field of kernel learning, such as deeper kernels and novel algorithms, to fill the gap that now exists with the most recent learning paradigms.

The papers accepted in the special session can be broadly divided in three categories: in Section 2, two applications of kernel methods to relevant real-world problems are presented. Section 3 presents an analysis of kernels for graph-structured data, and a novel kernel for trees. Finally, in Section 4, three novel algorithms are presented dealing with constraints and the efficiency of kernel methods.

2 Applications

The special session contribution [11] (C1) proposes an efficient kernel-based collaborative filtering technique for large scale top-N recommendation. Collaborative filtering (CF) is one of the most popular methods used in recommendation systems. In its typical setting a CF method, in order to make recommendation, only requires the so-called rating matrix ($\mathbf{R}^{n\times m}$) in which are defined the preferences of n users for m items. In particular, C1 focuses on implicit feedback where users preferences are expressed in a binary form (i.e, 1 for user-item interaction, 0 otherwise). The increasing interest on implicit feedbacks is justified by the fact that users are usually reluctant to give explicit opinions for items, while the number of interactions (i.e., implicit feedbacks) are continuously growing. Since these implicit feedbacks do not represent ratings, the recommendation task to solve is a top-N ranking of items for each user.

Recently, Kaggle organized a challenge, the Million Songs Dataset challenge (MSD) [12], on top-N recommendation with implicit feedback that was defined on a very large dataset with over 1M users and 380K items for a total of roughly 50M interactions. The winning method [13] is an extension of the item-based nearest-neighbors algorithm [14] which exploits the asymmetric cosine similarity. The main problem of this solution is that it is not theoretically well founded. In [15] a more theoretically grounded algorithm (CF-OMD) for top-N recommendation is proposed. The method, inspired by preference learning, explicitly optimizes the AUC and results show very good performance on the MovieLens dataset. The drawback of this approach is its complexity, since it requires the optimization of n quadratic problems defined on m variables.

C1 proposes a variant of CF-OMD based on linear kernel that makes it applicable to very large datasets. The method exploits the typical data sparsity of CF datasets by reducing the complexity of each single optimization problem by order of magnitudes. This formulation is then generalized to other kernels preserving the efficiency. The efficiency of the method is guaranteed only if the kernel is sparse. C1 shows how to create a sparse polynomial kernel starting from sparse data. Results shows that the proposed methods achieve good performance on the MSD with an execution time about 5 time faster than the state-of-the-art method.

The special session contribution [16] (C2) proposes an application of kernel methods to a problem from the Bioinformatics domain, i.e. the problem of RNA inverse folding. RNA polymers are an important class of molecules: not only they are involved in a variety of biological functions, from coding to decoding, from regulation to expression of genes, but crucially, they are nowadays easily synthesizable, opening interesting application scenarios in biotechnological and biomedical domains. C2 proposes a constructive machine learning framework to aid in the rational design of such polymers. The paper proposes a pipeline where a central role is played by a graph kernel [17] (see Section 3) in a supervised setting, that is used as a linear discriminant estimator. The adopted graph kernel allows for an explicit (sparse) feature space representation. This allows to explicitly access the discriminative importance of each feature which will be later used to define the notion of part importance over molecular parts. Then the set of most important parts is converted into specific sequence and structure constraints. Finally an inverse folding algorithm, based on ant colony optimization [18], uses these constraints to compute the desired RNA sequence.

3 Kernel Functions for non-vectorial data

Traditional machine learning models, e.g. neural networks, are suitable for dealing with data represented in a vectorial form. Therefore, when applied on real-world data, they need to resort to a fixed size vectorial representation of the data, that may involve a number of drawbacks such as information loss or the need of domain experts to design such representation at hand. More complex structures, for example graphs, may be a more suitable tool to describe relations in many real-world domains such as chemistry, molecular biology or speech and text processing. Kernel methods are particularly suited for these tasks, since all they need is a kernel function to be defined between two examples, in whatever form they are represented. For this reason, machine learning for structured data

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

is an active area of research [19, 20, 21] Indeed, several kernel functions have been defined for sequences [22, 23], trees [21] or graphs [24, 25, 26].

In the following we present two papers dealing with kernels for non-vectorial data. The first work analyzes kernels for graphs (C3), while the second one defines a kernel for trees defined on the state space of an Echo State Network (C4).

The special session contribution [27] (C3) deals with the problem of proposing a theoretically grounded and efficiently computable notion of expressiveness of a kernel in order to analyze the expressiveness of different graph kernels. Among the different machine learning techniques applicable to graphs, kernel methods are a well established solution. In fact graph kernels relieves the user from the definition of a vectorial representation of the data, a time consuming and task-specific operation. Several instances of graph kernels have been presented in literature. A recent advance in the field are fast kernels (near-linear time) that allow for an explicit, sparse feature space representation that can be successfully applied to big graph datasets [28, 29]. Each kernel considers as features different small substructures of the original graph. Empirical comparisons among different kernels can be found [17, 25] but, with few exceptions [24], no theoretical comparison is present. Moreover, usually kernels depend on one or more user-specified parameters, that control the resulting computational complexity, and change the induced hypothesis space. The selection of an appropriate kernel (and kernel parameters) can be a critical phase for achieving satisfying predictive performance on a specific task. In particular, different kernels induce different hypothesis spaces. In the context of graph kernels, the expressiveness of a kernel is defined as its ability to distinguish between non-isomorphic examples. In [30] it is shown that complete graph kernels (kernels that map each non-isomorphic graph in a different point in the feature space) are hard to compute. Thus, the kernels that we consider (and the ones that are used in practice) are not complete, but it is difficult to characterize their expressiveness, even in a relative way. If the non-zero features generated by different kernels are independent to each other, then it is easy to see that the more non-zero features a kernel generates, the more it is able to discriminate between examples, and so the more it is expressive. However, this is not the case with structural features, where there are strong dependency relationships among them, i.e. a feature can be non-zero only if some specific features are too [31]. In this case, there is no easy way to assess how expressive a kernel is. Consequently the contribution of C3 results to be quite important in this field of research. In particular, the result is interesting since it builds upon the Rademacher Complexity (RC), a powerful notion of complexity of an hypothesis space, which is used in the finite sample analysis of the generalization error of and hypothesis chosen in a space of possible ones during the learning process [6, 7, 8, 9, 10]. The results reported in C3 on real world dataset and state-of-the-art graph kernels confirm some empirically known expressivity properties and support them with an adequate theoretical background. C3 proposes also a future application of the proposed approach. In fact the proposed measure can, in future, be applied to perform kernel/parameters selection. Moreover, C3 paves the way to the exploitation of more complex measures of complexity such as the Local Rademacher complexity.

The special session contribution [32] (C4) defines a kernel for trees as an *activation kernel* over the *reservoir* state space of a Tree Echo State Network [33] (TreeESN). As for standard Echo State Networks, the architecture of a TreeESN is composed of an untrained recurrent non-linear reservoir and a linear readout that can be trained by efficient linear methods. C4 exploits the recursive encoding of the (tree-)structure in the state activations of the untrained recurrent layer to define a kernel over trees. The intuition is that the dense recursive encoding defined by the untrained reservoir of a TreeESN can provide a rich representation of the structural knowledge that allows defining an efficient kernel over very small reservoirs. The paper discusses how the contractive property of the reservoir induces a topographic organization of the state space that can be used to compute structural matches in terms of pairwise distances between points in the state space. More in detail, to produce a mapping for a tree, C4 uses a TreeESN to project each substructure (subtree) t_u of an input tree t to a N-dimensional point x_u corresponding to the reservoir activation for the tree node acting as root of the substructure. Thus, a tree t comprising T nodes is transformed into T vectors, one for each node u of the tree. Evaluating the similarity between two structures, in this context, becomes a matter of computing distances between points in the reservoir state space. This distance can be defined as a (thresholded) Euclidean distance. The experimental analysis shows that the proposed kernel is capable of achieving competitive classification results by relying on very small reservoirs comprising as little as 10 sparsely connected recurrent neurons.

4 Learning Algorithms

The special session contribution [34] (C5) deals with one of the main problems that arise when one has to learn from data. In particular any learning procedure is subjected to many constraints [6, 35] which can be grouped in two main families: hard and soft constraints [36]. Hard constraints cannot be violated under any circumstance while soft constraints can be violated at the cost of some penalization. Hard constraints are often more expressive respect to the soft ones when it comes to formalize the learning procedure and to solve the associated optimization problem [37]. Unfortunately, from a computational point of view, dealing with hard constraints results in more difficult problems (e.g. NP-Complete or NP-Hard problems) respect to deal with the soft ones [38, 39]. For this reason, often, hard-constrained learning problems are solved through one or more soft-constrained problems. The motivation behind this approach is that, under suitable hypothesis, the optimal solutions of those soft-constrained learning problems tend to those of the original hard-constrained ones [40]. Consequently the challenge is to find families of hard-constrained problems which can be addressed by using this quite promising approach. In particular, by building on [40] and by extending [36], C5 focuses on a particular set of constraints, the pointwise constraints (PWCs). PWCs are associated to a finite set of sample, in which each element of the set defines one such constraint. PWCs are very often used in ML problems, since they are able to model very general learning conditions. The main contribution of the C5 is that it shows that the optimal solution to the learning problem with hard bilateral and linear PWCs can be obtained as the limit of the sequence of optimal solutions to the related learning problems with soft bilateral and linear PWCs, when the penalty parameter tends to infinity. In particular, in the paper, the optimal solutions to the two problems, obtained in the particular cases in which there are only hard linear constraints or only soft linear constraints, is compared. Moreover, the limit behavior of the optimal solution to the soft constrained learning problem, when the penalty parameter tends to infinity, is studied. The paper also discusses on the future direction of this promising field or research where the challenge is to extend the proposed approach to learning problems characterized by mixed hard/soft constraints, when all the hard constraints should be replaced by soft constraints.

The special session contribution [41] (C6) is a simple and effective kernel approximation approach. In real applications, the computational cost of kernel methods could be prohibitive. Specifically, the cost of computation and storage of kernel matrix for n examples with d features is $O(n^2 d)$ and $O(n^2)$, respectively. Moreover, all the learning algorithms require several matrix-vector multiplications with a computational cost of $O(n^3)$ each.
For these reasons, finding the best kernel approximation is an interesting and challenging task. The most popular method is the Nystrom approximations [42]. This method attempts to observe some columns and corresponding rows of the matrix in order to recover the kernel matrix. Different extensions of Nystrom method have been analyzed in the past [43], using non-uniform sampling to select the columns. For example, K-means clustering of input data can be exploited to sample the columns as in [44]. These methods show improvement over the standard Nystrom [43]. Recently, in [45], a memory efficient kernel approximation has been proposed. This approach works by finding blocks in the kernel matrix followed by approximation of each block. The blocks are composed by clustering the data in the input space. MEKA uses the same space as the rank r approximation of the Nystrom method and is able to perform a rank cr approximation, where c is the number of clusters.
C6 exploits the alternating least squares (ALS) [46, 47] and proposes an algorithm for rank r approximation of the kernel matrix by computing only a subset Ω of all the entries of the kernel (KALS). KALS solves a non-convex optimization problem followed by a matrix completion step using ALS. KALS has a complexity of $O(|\Omega| r^2)$ and shows better performance than baseline and state-of-the-art kernel approximation methods on different benchmark datasets. In C6, a complete theoretical analysis extends the current guarantees of ALS for kernel approximation. The convergence of KALS to the optimal SVD solution is proved under a *coherence* assumption. As future work, it will be possible to evaluate different efficient sampling schemes in order to remove the *coherence*

assumption generalizing the KALS algorithm.

The last contribution to the special session [48] (C7) focuses on developing a new learning approach, based on Gaussian processes (GP), for dealing with times series prediction in the case of structured data. The paper builds upon a series of results, coming from different fields in ML, in order to solve a quite challenging task. In fact time series prediction is a classic topic in ML with has been successfully applied to a wide range of applications [49]. This task is becoming increasingly challenging due to the rise of new complex data structure like sequences, trees or graphs in many real world applications such social network analysis or intelligent tutoring systems [50, 51]. Classical learning paradigms are not able to handle or take advantage of the structure of the data since they handle just vectorial data [49]. Nevertheless, GPs have shown to be state-of-the-art tools for dealing with time series prediction [52, 53]. Moreover, GP are based on kernel values for the given data as input and, as shown in the C7, a special choice of the prior allows to express the predictions provided by GPs as an affine combination of given data. Based on this consideration the C7 shows that it is possible to build upon the vast literature of distance measures and kernels for structured data, such as alignment distances, tree and graph kernels, to access structured data instead of vectors as time series entries [54, 55]. Moreover, as shown in C7, it is possible rely on established embeddings of the space of structured objects, which is a discrete data space in itself, in a smooth kernel or pseudo-Euclidean space, and it is possible access such outputs of a GP for structured data e.g. via efficient distance computations [56, 57]. An additional contribution of the C7, is to propose a way to face an additional challenge posed by the high computational complexity of GPs with respect to the number of data points, and the structure kernel computation. As a speed-up, C7 applies state-of-the-art approximation methods for the Gaussian Processes [58] as well as the dissimilarity and kernel data [59], obtaining good-quality predictions in linear time. Results on real world dataset shows that GP seem promising to predict time series of structured data, or relational data in general. By returning an affine combination, they enable further processing, such as classification and clustering. C7 also shows some future possible improvements of the proposed method. In particular, usual hyperparameter optimization techniques depend on a vectorial data representation [58] and one has to adapt them for a relational case. Moreover, an affine combination might not be a sufficient data representation of the predicted point for some applications. For such cases, an inverse problem has to be solved: finding the original point that maps to the affine combination in the pseudo-Euclidean space. These are two interesting challenges of this field of research.

References

[1] T. Hofmann, B. Scholkopf, and A. J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

[2] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, July 2011.

[3] F. Aiolli and M. Donini. Easymkl: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169:215 – 224, 2015.

[4] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge university press, 2004.

[5] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 1:326–334, 1965.

[6] V. N. Vapnik. *Statistical learning theory.* Wiley–Interscience, 1998.

[7] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

[8] L. Oneto, A. Ghio, S. Ridella, and D. Anguita. Global rademacher complexity bounds: From slow to fast convergence rates. *Neural Processing Letters*, pages (in–press), 2015.

[9] P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[10] L. Oneto, A. Ghio, S. Ridella, and D. Anguita. Local rademacher complexity: Sharper risk bounds with and without unlabeled samples. *Neural Networks*, 65:115–125, 2015.

[11] F. Aiolli and M. Polato. Kernel based collaborative filtering for very large scale top-n item recommendation. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[12] B. McFee, T. Bertin-Mahieux, D. P. W. Ellis, and G. R. G. Lanckriet. The million song dataset challenge. In *International conference companion on World Wide Web*, 2012.

[13] F. Aiolli. Efficient top-N recommendation for very large scale binary rated datasets. In *ACM Recommender Systems Conference*, 2013.

[14] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transaction Infformayion System*, 22(1):143–177, 2004.

[15] F. Aiolli. Convex AUC optimization for top-N recommendation with implicit feedback. In *ACM Recommender Systems Conference*, 2014.

[16] F. Costa, P. Kohvaei, and R. Kleinkauf. Rnasynth: constraints learning for rna inverse folding. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[17] F. Costa and K. De Grave. Fast neighborhood subgraph pairwise distance kernel. In *International Conference on Machine Learning*, 2010.

[18] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.

[19] T. Gärtner. *Kernels for Structured Data.* PhD thesis, University of Bonn, 2005.

[20] Q. Shi, J. Petterson, G. Dror, J. Langford, A. J. Smola, and S. V. N. Vishwanathan. Hash Kernels for Structured Data. *Journal of Machine Learning Research*, 10:2615–2637, 2009.

[21] G. Da San Martino and A. Sperduti. Mining Structured Data. *IEEE Computational Intelligence Magazine*, 5(1):42–49, 2010.

[22] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text Classification using String Kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[23] C. S. Leslie, E. Eskin, and W. S. Noble. The Spectrum Kernel: A String Kernel for SVM Protein Classification. In *Pacific Symposium on Biocomputing*, 2002.

[24] G. Da San Martino, N. Navarin, and A. Sperduti. A Tree-Based Kernel for Graphs. In *Proceedings of the Twelfth SIAM International Conference on Data Mining*, 2012.

[25] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 2011.

[26] F. Orsini, P. Frasconi, and L. D. Raedt. Graph invariant kernels. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015.

[27] L. Oneto, N. Navarin, M. Donini, A. Sperduti, F. Aiolli, and D. Anguita. Measuring the expressivity of graph kernels through the rademacher complexity. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[28] G. Da San Martino, N. Navarin, and A. Sperduti. A memory efficient graph kernel. In *International Joint Conference on Neural Networks*, 2012.

[29] G. Da San Martino, N. Navarin, and A. Sperduti. Exploiting the ODD framework to define a novel effective graph kernel. In *ESANN*, 2015.

[30] T. Gartner, P. Flach, S. Wrobel, and T. Gärtner. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Computational Learning Theory*, 2003.

[31] F. Aiolli, M. Donini, N. Navarin, and A. Sperduti. Multiple graph-kernel learning. In *IEEE Symposium Series on Computational Intelligence*, 2015.

[32] D. Bacciu, C. Gallicchio, and A. Micheli. A reservoir activation kernel for trees. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[33] C. Gallicchio and A. Micheli. Tree Echo State Networks. *Neurocomputing*, 101:319–337, 2013.

[34] G. Gnecco, M. Gori, S. Melacci, and M. Sanguineti. Learning with hard constraints as a limit case of learning with soft constraints. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[35] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[36] G. Gnecco, M. Gori, S. Melacci, and M. Sanguineti. Learning with mixed hard/soft pointwise constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2019–2032, 2015.

[37] L. Oneto, A. Ghio, S. Ridella, and D. Anguita. Learning resource-aware models for mobile devices: from regularization to energy efficiency. *Neurocomputing*, In Press.

[38] F. Gieseke, K. Lars Polsterer, C. E. Oancea, and C. Igel. Speedy greedy feature selection: Better redshift estimation via massive parallelism. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2014.

[39] L. Oneto, S. Ridella, and D. Anguita. Learning hardware-friendly classifiers through algorithmic stability. *ACM Transaction on Embedded Computing*, In Press.

[40] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.

[41] B. Piyush and K. Harish. Efficient low rank approximation via alternating least squares for scalable kernel learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[42] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Neural Information Processing Systems*, pages 682–688, 2001.

[43] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

[44] K. Zhang, I. W. Tsang, and J. T. Kwok. Improved nyström low-rank approximation and error analysis. In *International conference on Machine learning*, 2008.

[45] S. Si, C. J. Hsieh, and I. Dhillon. Memory efficient kernel approximation. In *International Conference on Machine Learning*, 2014.

[46] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[47] S. Bhojanapalli, P. Jain, and S. Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *ACM-SIAM Symposium on Discrete Algorithms*, 2015.

[48] B. Paassen, C. Gopfert, and B. Hammer. Gaussian process prediction for time series of structured data. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[49] R. H. Shumway and D. S. Stoffer. *Time series analysis and its applications.* Springer Science & Business Media, 2013.

[50] K. R. Koedinger, E. Brunskill, R. SJd Baker, Elizabeth A McLaughlin, and John Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

[51] N. Santoro, W. Quattrociocchi, P. Flocchini, A. Casteigts, and F. Amblard. Time-varying graphs and social network analysis: Temporal indicators and metrics. *arXiv preprint arXiv:1102.0629*, 2011.

[52] J. Wang, A. Hertzmann, and D. M. Blei. Gaussian process dynamical models. In *Neural information Processing Systems*, 2005.

[53] S Roberts, M Osborne, M Ebden, S Reece, N Gibson, and S Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984):1–25, 2013.

[54] G. Da San Martino and A. Sperduti. Mining structured data. *IEEE Computational Intelligence Magazine*, 5(1):42–49, 2010.

[55] F. Aiolli, G. Da San Martino, and A. Sperduti. An efficient topological distance-based tree kernel. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):1115–1120, 2015.

[56] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity data sets. *Neural Computation*, 22(9):2229–2284, 2010.

[57] D. Hofmann, F. M. Schleif, B. Paaßen, and B. Hammer. Learning interpretable kernelized prototype-based models. *Neurocomputing*, 141:84–96, 2014.

[58] M. Deisenroth and J. W. Ng. Distributed gaussian processes. In *International Conference on Machine Learning*, 2015.

[59] A. Gisbrecht and F. M. Schleif. Metric and non-metric proximity transformations at linear costs. *Neurocomputing*, 167:643–657, 2015.