Efficient evaluation of large sequence kernels

Pavel P. Kuksa[1], Vladimir Pavlovic[2]
[1]NEC Laboratories America, Inc [2]Department of Computer Science, Rutgers University

Classification of sequences drawn from a finite alphabet using a family of string kernels with inexact matching (e.g., spectrum or mismatch) has shown great success in machine learning [6, 3, 9, 4]. However, selection of optimal mismatch kernels for a particular task is severely limited by inability to compute such kernels for long substrings with potentially many mismatches. We extend prior work on algorithms for computing (k, m) mismatch string kernels and introduce a new method that allows us to evaluate kernels for large k, m. This makes it possible to explore a larger set of kernels with a wide range of kernel parameters, opening a possibility to better model selection and improved performance of the string kernels. To investigate the utility of large (k,m) string kernels, we consider several sequence classification problems, including protein remote homology detection, and music classification. Our results show that increased k-mer lengths with larger substitutions can improve classification performance.

Background. A number of state-of-the-art approaches to classification of sequences over finite alphabet Σ rely on fixed-length representations $\Phi(X)$ of sequences as the *spectra* ($|\Sigma|^k$-dimensional histogram) of counts of short substrings (k-mers), contained, possibly with up to m mismatches, in a sequence, c.f., spectrum/mismatch methods [6, 7, 3]. However, computing similarity scores, or kernels, $K(X,Y)=\Phi(X)^T\Phi(Y)$ using these representations can be challenging, e.g., efficient $O(k^{m+1}|\Sigma|^m(|X|+|Y|))$ trie-based mismatch kernel algorithm [7] strongly depends on the alphabet size and the number of mismatches m.

More recently, [4] introduced linear time algorithms with *alphabet-independent* complexity $O(c_{k,m}(|X|+|Y|)$ applicable to computation of a large class of existing string kernels. The authors show that it is possible to compute an inexact (k, m) kernel as

$$K(X,Y|m,k) = \sum_{a\in X}\sum_{b\in Y}\mathcal{I}(a,b) = \sum_{i=0}^{\min(2m,k)} M_i\mathcal{I}_i, \quad (1)$$

where $\mathcal{I}(a, b)$ is the number of common substrings in the intersection of the mutation neighborhoods of a and b, $\mathcal{I}_i$ is the size of the intersection of k-mer mutational neighborhood for Hamming distance i, and M_i is the number of observed k-mer pairs in X and Y having Hamming distance i.

This result however requires that the number of identical substrings in (k, m)-mutational neighborhoods of k-mers a and b (the intersection size) be known in advance, for every possible pair of m and the Hamming distance d between k-mers (k and $|\Sigma|$ are free variables). Obtaining the closed form expression for the intersection size for arbitrary k, m is challenging, with no clear systematic way of enumerating the intersection of two mutational neighborhoods. Closed form solutions obtained in [4] were only provided for cases when m is small ($m \leq 3$). No systematic way of obtaining these intersection sizes has been proposed in [4].

In this work we introduce a systematic and efficient procedure for obtaining intersection sizes that can be used for large k and m and arbitrary alphabet size $|\Sigma|$. This will allow us to effectively explore a much larger class of (k, m) kernels in the process of model selection which could further improve performance of the string kernel method as we show experimentally.

Efficient evaluation of large sequence kernels. For large values of k and m finding intersection sizes needed for kernel computation can be problematic. This is because while for smaller values of m combinatorial closed form solution can be found easily, for larger values of m finding it becomes more difficult due to an increase in the number of combinatorial possibilities as the mutational neighborhood increases (exponentially) in size. On the other hand, direct computation of the intersection by trie traversal algorithm is computationally difficult for large k and m as the complexity of traversal is $O(k^{m+1}|\Sigma|^k)$, i.e. is exponential in both k and m. The above mentioned issues do not allow for efficient kernel evaluation for large k and m.

Reduction-based computation of intersection size coefficients. We will now show that it is possible to efficiently compute the intersection sizes by reducing $(k, m, |\Sigma|)$ intersection size problem to a set of less complex intersection size computations and solving linear systems of equations. We discuss this approach below.

The number of k-mers at the Hamming distance of at most m from both k-mers a and b, $\mathcal{I}(a, b)$, can be found in a weighted form

$$\mathcal{I}(a,b) = \sum_{i=0}^{m} w_i(|\Sigma|-1)^i. \quad (2)$$

Coefficients w_i depend only on the Hamming distance $d(a, b)$ between k-mers a and b for fixed k, m, $|\Sigma|$.

For every Hamming distance $0 \leq d(a, b) \leq 2m$, the corresponding set of coefficients $w_i, i = 0, 1, \ldots, m$ can be found by solving a linear system $Aw = \mathcal{I}$ of $m + 1$ equations with each equation corresponding to a particular alphabet size $|\Sigma| \in \{2, 3, \ldots, m + 2\}$. The left-hand side matrix A is an (m+1,m+1) matrix with elements $a_{ij} = i^{j-1}$, $i = 1, \ldots, m + 1$, $j = 1, \ldots, m + 1$.

$$A = \begin{pmatrix} 1^0 & 1^1 & 1^2 & \ldots & 1^m \\ 2^0 & 2^1 & 2^2 & \ldots & 2^m \\ & & & \ldots & \\ (m+1)^0 & (m+1)^1 & (m+1)^2 & \ldots & (m+1)^m \end{pmatrix}$$

The right-hand side $\mathcal{I} = (I_0, I_1, \ldots, I_m)^T$ is a vector of intersection sizes for a particular setting of k, m, d, $|\Sigma| = 2, 3, \ldots, m+2$. Here, $I_i, i = 0 \ldots m$ is the intersection size for a pair of k-mers over alphabet size $i+2$. Note that I_i need only be computed for small alphabet sizes, up to $m+2$. Hence, this vector can feasibly be computed using a trie traversal for a pair of k-mers at Hamming distance d even for moderately large k as the size of the trie is only $(m+2)^k$ as opposed to $|\Sigma|^k$. This allows now to evaluate kernels for large k and m as the traversal is performed over much smaller tries, e.g., even in case of relatively small protein alphabet with $|\Sigma| = 20$, for $m = 6$ and $k = 13$, the size of the trie is $20^{13}/8^{13} = 149011$ times smaller. Coefficients w obtained by solving $Aw = \mathcal{I}$ do not depend on the alphabet size $|\Sigma|$. In other words, once found for a particular combination of values (k, m), these coefficients can be used to determine intersection sizes for any given finite alphabet $|\Sigma|$ using Eq. 2.

Experimental evaluation. We evaluate the utility of large (k, m) computations as a proxy for model selection, by allowing a significantly wider range of kernel parameters to be investigated during the selection process. Such large range evaluation is the first of its kind, made possible by our efficient kernel evaluation algorithm. In these evaluations we follow the experimental settings considered in [5] and [4]. We use standard benchmark datasets: the SCOP dataset (7329 sequences, 54 experiments) [9] for remote protein homology detection, and music genre data[1] (10 classes, 1000 seqs) [8] for multi-class genre prediction.

Table 1: Remote homology. Classification performance of the mismatch kernel method

Kernel	Mean ROC	Mean ROC50
mismatch(5,1)	87.75	41.92
mismatch(5,2)	90.67	49.09
mismatch(6,2)	90.74	49.66
mismatch(6,3)	90.98	49.36
mismatch(7,3)	91.31	52.00
mismatch(7,4)	90.84	49.29
mismatch(9,4)	91.45	53.51
mismatch(10,5)	**91.60**	**53.78**
mismatch(13,6)	90.98	50.11

Table 2: Multi-class music genre recognition. Classification performance of the mismatch method

Kernel	Error	Top-2 Error	F1	Top-2 F1
mismatch(5,1)	34.8	18.3	65.36	81.95
mismatch(5,2)	32.6	18.0	67.51	82.21
mismatch(6,3)	31.2	**17.2**	68.92	**83.01**
mismatch(7,4)	**31.1**	18.0	**68.96**	82.16
mismatch(9,3)	31.4	18.0	68.59	82.33
mismatch(9,4)	32.2	17.8	67.83	82.36
mismatch(10,3)	32.3	18.0	67.65	82.12
mismatch(10,4)	31.7	19.1	68.29	81.04

Results of mismatch kernel classification for the remote homology detection problem are shown in Table 1. We observe that larger values of k and m perform better compared to typically used values of k=5-6, m=1-2. For instance, (k=10,m=5)-mismatch kernel achieves significantly higher average ROC50 score of 53.78 compared to ROC50 of 41.92 and 49.02 for the (k=5,m=1)- and (k=5,m=2)- mismatch kernels. The utility of such large mismatch kernels was not possible to investigate prior to this study.

We also note that the results for per-family or per-superfamily based parameter selection suggest the need for model selection and the use of multiple kernels, e.g., per-family kernel selection results in much higher ROC50 of 60.32 compared to 53.78 of the best single kernel.

For the music genre classification task (Table 2), parameter combinations with moderately long k and larger values of m tend to perform better than kernels with small m. As can be seen from results, larger values of m are important for achieving good classification accuracy and outperform setting with small values of m.

Conclusions. In this work we proposed a new systematic method that allows evaluation of inexact string family kernels for long substrings k with large number of mismatches m. The method finds the intersection set sizes by explicitly computing them for *small* alphabet size $|\Sigma|$ and then generalizing this to arbitrary *large* alphabets. We show that this enables one to explore a larger set of kernels which as we demonstrate experimentally can further improve performance of the string kernels.

References

[1] Chris H.Q. Ding and Inna Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358, 2001.

[2] Eugene Ie, Jason Weston, William Stafford Noble, and Christina Leslie. Multi-class protein fold recognition using adaptive codes. In *ICML '05*, pages 329–336, New York, NY, USA, 2005. ACM.

[3] Rui Kuang, Eugene Ie, Ke Wang, Kai Wang, Mahira Siddiqi, Yoav Freund, and Christina S. Leslie. Profile-based string kernels for remote homology detection and motif extraction. In *CSB*, pages 152–160, 2004.

[4] Pavel Kuksa, Pai-Hsi Huang, and Vladimir Pavlovic. Scalable algorithms for string kernels with inexact matching. In *NIPS*, 2008.

[5] Pavel P. Kuksa and Vladimir Pavlovic. Spatial representation for efficient sequence classification. In *ICPR*, 2010.

[6] Christina Leslie and Rui Kuang. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.*, 5:1435–1455, 2004.

[7] Christina S. Leslie, Eleazar Eskin, Jason Weston, and William Stafford Noble. Mismatch string kernels for SVM protein classification. In *NIPS*, pages 1417–1424, 2002.

[8] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *SIGIR '03*, pages 282–289, New York, NY, USA, 2003. ACM.

[9] Jason Weston, Christina Leslie, Eugene Ie, Dengyong Zhou, Andre Elisseeff, and William Stafford Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

[1] `http://opihi.cs.uvic.ca/sound/genres`