Multiresolution Kernels

Marco Cuturi, Kenji Fukumizu

Multiresolution Kernels

Marco Cuturi
Ecole des Mines de Paris
Fontainebleau, France;
Institute of Statistical Mathematics
Tokyo, Japan.
marco.cuturi@ensmp.fr

Kenji Fukumizu
Institute of Statistical Mathematics
Tokyo, Japan.
fukumizu@ism.ac.jp

Abstract

We present in this work a new methodology to design kernels on data which is structured with smaller components, such as text, images or sequences. This methodology is a template procedure which can be applied on most kernels on measures and takes advantage of a more detailed "bag of components" representation of the objects. To obtain such a detailed description, we consider possible decompositions of the original bag into a collection of nested bags, following a prior knowledge on the objects' structure. We then consider these smaller bags to compare two objects both in a detailed perspective, stressing local matches between the smaller bags, and in a global or coarse perspective, by considering the entire bag. This multiresolution approach is likely to be best suited for tasks where the coarse approach is not precise enough, and where a more subtle mixture of both local and global similarities is necessary to compare objects. The approach presented here would not be computationally tractable without a factorization trick that we introduce before presenting promising results on an image retrieval task.

1 Introduction

There is strong evidence that kernel methods [11] can deliver state-of-the-art performance on most classification tasks when the input data lies in a vector space. Arguably, two factors contribute to this success. First, the good ability of kernel algorithms, such as the SVM, to generalize and provide a sparse formulation for the underlying learning problem; Second, the capacity of nonlinear kernels, such as the polynomial and RBF kernels, to quantify meaningful similarities between vectors, notably non-linear correlations between their components. Using kernel machines with non-vectorial data (e.g., in bioinformatics, pattern recognition or signal processing tasks) requires more arbitrary choices, both to represent the objects and to chose suitable kernels on those representations. The challenge of using kernel methods on real-world data has thus recently fostered many proposals for kernels on complex objects, notably for strings, trees, images or graphs to cite a few.

A strategy often quoted as the generative approach to this problem takes advantage of a generative model, that is an adequate statistical model for the objects, to derive feature representations for the objects. In practice this often yields kernels to be used on the histograms of smaller components sampled in the objects, where the kernels take into account

the geometry of the underlying model in their similarity measures [7, 9, 4, 6, 3]. The previous approaches coupled with SVM's combine both the advantages of using discriminative methods with generative ones, and produced convincing results on many tasks.

One of the drawbacks of such representations is however that they implicitly assume that each component has been generated independently and in a stationary way, where the empirical histogram of components is seen as a sample from an underlying stationary measure. While this viewpoint may translate into adequate properties for some learning tasks (such as translation or rotation invariance when using histograms of colors to manipulate images [2]), it might prove too restrictive and hence inadequate for other types of problems. Namely, tasks which involve a more subtle mix of detecting *both* conditional (with respect to the location of the components for instance) and global similarities between the objects. Such problems are likely to arise for instance in speech, language, time series or image processing. In the first three tasks, this consideration is notably treated by most state-of-the-art methods through dynamic programming algorithms capable of detecting and penalizing accordingly local matches between the objects. Using dynamic programming to produce a kernel yielded fruitful results in different applications [14, 12], with the limitation that the kernels obtained in practice are not always positive definite, as reviewed in [14]. Other kernels proposed for sequences [10] directly incorporate a localization information into each component, augmenting considerably the size of the component space, and then introduce some smoothing (such as mismatches) to avoid representations that would be too sparse.

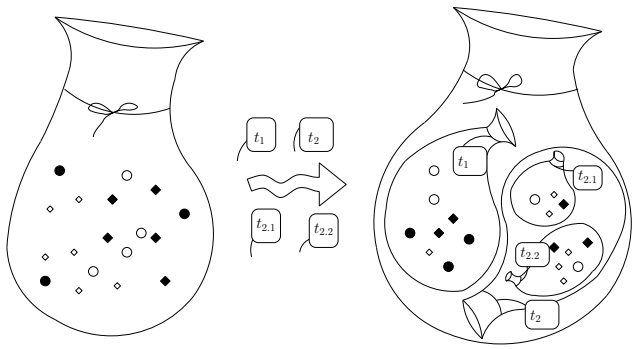

Figure 1: From the bag of components representation to a set of nested bags, using a set of conditioning events.

We propose in this work a different approach grounded on the generative approach previously quoted, managing however to combine both conditional and global similarities when comparing two objects. The motivation behind this approach is both intuitive and computational: intuitively, the global histogram of components, that is the simple bag of components representation of Figure 1, may seem inadequate if the components' appearance seem to be clearly conditioned by some external events. This phenomenon can be taken into account by considering collections (indexed on the same set of events, to be defined) of nested bags or histograms to describe the object. Kernels that would only rely on these detailed resolutions might however miss the bigger picture that is provided by the global histogram. We propose a trade-off between both viewpoints through a combination that aims at giving a balanced account of both fine and coarse perspectives, hence the name of multiresolution kernels, which we introduce formally in Section 2. On the computational side, we show how such a theoretical framework can translate into an efficient factorization detailed in Section 3. We then provide experimental results in Section 4 on an image retrieval task which shows that the methodology improves the performance of kernel based state-of-the art techniques in this field.

2 Multiresolution Kernels

In most applications, complex objects can be represented as histograms of components, such as texts as bags of words or images and sequences as histograms of colors and letters. Through this representation, objects are cast as probability laws or measures on the space $\mathcal{X}$ of components, typically multinomials if $\mathcal{X}$ is finite [9, 6, 2, 8], and compared as such through kernels on measures. An obvious drawback of this representation is that all contextual information on how the components have been sampled is lost, notably any general sense of position in the objects, but also more complex conditional information that may be induced from neighboring components, such as transitions or long range interactions.

In the case of images for instance, one may be tempted to consider not only the overall histogram of colors, but also more specialized histograms which may be relevant for the task. If some local color-overlapping in the images is an interesting or decisive feature of the learning problem, these specialized histograms may be generated arbitrarily following a grid, dividing for instance the image into 4 equal parts, and computing histograms for each corner before comparing them pairwise between two images (see Figure 2 for an illustration). If sequences are at stake, these may also be sliced into predefined regions to yield local histograms of letters. If the strings are on the contrary assumed to follow some Markovian behaviour (namely that the appearance of letters in the string is independent of their exact location but only depends on the few letters that precede them), an interesting index would translate into a set of contexts, typically a complete suffix dictionary as detailed in [4]. While the two previous examples may seem opposed in the way the histograms are generated, both methodologies stress a particular class of events (location or transitions) that give an additional knowledge on how the components were sampled in the objects. Since both these two approaches, and possibly other ones, can be applied within the framework of this paper using a unified formalism, we present our methodology using a general notation for the index of events. Namely, we note $\mathcal{T}$ for an arbitrary set of conditioning events, assuming these events can be directly observed on the object itself, by contrast with the latent variables approach of [13]. Considering still, following the generative approach, that an object can be mapped onto a probability measure μ on $\mathcal{X}$, we have that the realization of an event $t \in \mathcal{T}$ can be interpreted under the light of a joint probability $\mu(x,t)$, with $x \in \mathcal{X}$, factorized through Bayes' law as $\mu(x|t)\mu(t)$ to yield the following decomposition of μ as

$$\mu = \sum_{t \in \mathcal{T}} \mu_t,$$

where each $\mu_t \stackrel{\text{def}}{=} \mu(\cdot|t)\mu(t)$ is an element of the set of sub-probability measures $M_+^s(\mathcal{X})$, that is the set of positive measures ρ on $\mathcal{X}$ such that their total mass $\rho(\mathcal{X})$ denoted as $|\rho|$ is *less than* or equal to 1. To take into account the information brought by the events in $\mathcal{T}$, objects can hence be represented as families of measures of $M_+^s(\mathcal{X})$ indexed by $\mathcal{T}$, namely elements μ contained in $M_{\mathcal{T}}(\mathcal{X}) \stackrel{\text{def}}{=} M_+^s(\mathcal{X})^{\mathcal{T}}$.

2.1 Local Similarities Between Measures Conditioned by Sets of Events

To compare two objects under the light of their respective decompositions as sub-probability measures μ_t and μ'_t, we make use of an arbitrary positive definite kernel k on $M_+^s(\mathcal{X})$ to which we will refer to as the base kernel throughout the paper. For interpretation purposes only, we may assume in the following sections that k can be written as e^{-d^2} where d is an Euclidian distance in $M_+^s(\mathcal{X})$. Note also that the kernel is defined not only on probability measures, but also on sub-probabilities. For two elements μ, μ' of $M_{\mathcal{T}}(\mathcal{X})$ and a given element $t \in \mathcal{T}$, the kernel

$$k_t(\mu, \mu') \stackrel{\text{def}}{=} k(\mu_t, \mu'_t)$$

measures the similarity of μ and μ' by quantifying how similarly their components were generated conditionally to event t. For two different events s and t of $\mathcal{T}$, k_s and k_t can be associated through polynomial combinations with positive factors to result in new kernels, notably their sum $k_s + k_t$ or their product $k_s k_t$. This is particularly adequate if some complementarity is assumed between s and t, so that their combination can provide new insights for a given learning task. If on the contrary the events are assumed to be similar, then they can be regarded as a unique event $\{s\} \cup \{t\}$ and result in the kernel

$$k_{\{s\}\cup\{t\}}(\mu, \mu') \stackrel{\text{def}}{=} k(\mu_s + \mu_t, \mu'_s + \mu'_t),$$

which will measure the similarity of m and m' when *either* s or t occurs. The previous formula can be extended to model kernels indexed on a set $T \subset \mathcal{T}$ of similar events, through

$$k_T(m, m') \stackrel{\text{def}}{=} k\left(\mu_T, \mu'_T\right), \text{ where } \mu_T \stackrel{\text{def}}{=} \sum_{t \in T} \mu_t \text{ and } \mu'_T \stackrel{\text{def}}{=} \sum_{t \in T} \mu'_t.$$

Note that this equivalent to defining a distance between elements μ and μ' conditionned by T as $d^2_T(\mu, \mu') \stackrel{\text{def}}{=} d^2(\mu_T, \mu'_T)$.

2.2 Resolution Specific Kernels

Let P be a finite partition of $\mathcal{T}$, that is a finite family $P = (T_1, ..., T_n)$ of sets of $\mathcal{T}$, such that $T_i \cap T_j = \varnothing$ if $1 \leq i < j \leq n$ and $\bigcup_{i=1}^n T_i = \mathcal{T}$. We write $\mathcal{P}(\mathcal{T})$ for the set of all partitions of $\mathcal{T}$. Consider now the kernel defined by a partition P as

$$k_P(\mu, \mu') \stackrel{\text{def}}{=} \prod_{i=1}^{n} k_{T_i}(\mu, \mu'). \tag{1}$$

The kernel k_P quantifies the similarity between two objects by detecting their joint similarity under all possible events of $\mathcal{T}$, given an a priori similarity assumed on the events which is expressed as a partition of $\mathcal{T}$. Note that there is some arbitrary in this definition since, following the convolution kernels [5] approach for instance, a simple multiplication of base kernels k_{T_i} to define k_P is used, rather than any other polynomial combination. More precisely, the multiplicative structure of Equation (1) quantifies how two objects are similar given a partition P in a way that imposes for the objects to be similar according to all subsets T_i. If k can be expressed as a function of a distance d, k_P can be expressed as the exponential of

$$d^2_P(\mu, \mu') \stackrel{\text{def}}{=} \sum_{i=1}^{n} d^2_{T_i}(\mu, \mu'),$$

a quantity which penalizes local differences between the decompositions of μ and μ' over $\mathcal{T}$, as opposed to the coarsest approach where $P = \{\mathcal{T}\}$ and only $d^2(\mu, \mu')$ is considered.

As illustrated in Figure 2 in the case of images expressed as histograms indexed over locations, a partition of $\mathcal{T}$ reflects a given belief on how events should be associated to belong to the same set or dissociated to highlight interesting dissimilarities. Hence, all partitions contained in the set $\mathcal{P}(\mathcal{T})$ of all possible partitions[1] are not likely to be equally meaningful given that some events may look more similar than others. If the index is based on location, one would naturally favor mergers between neighboring indexes. For contexts, a useful topology might also be derived by grouping contexts with similar suffixes.

[1] which is quite a big space, since if $\mathcal{T}$ is a finite set of cardinal r, the cardinal of the set of partitions is known as the Bell Number of order r with $B_r = \frac{1}{e}\sum_{u=1}^{\infty} \frac{u^r}{u!} \underset{r\to\infty}{\sim} e^{r \ln r}$.

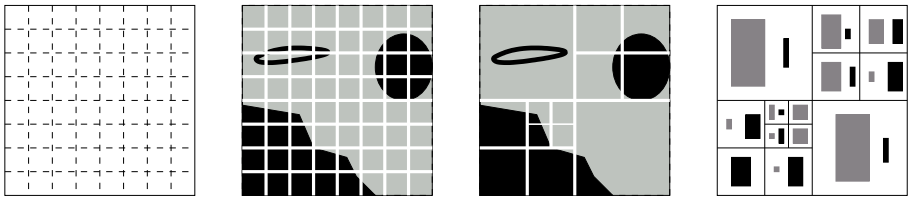

Figure 2: A useful set of events $\mathcal{T}$ for images which would focus on pixel localization can be represented by a grid, such as the 8×8 one represented above. In this case P_3 corresponds to the 4^3 windows presented in the left image, P_2 to the 16 larger square obtained when grouping 4 small windows, P_1 to the image divided into 4 equal parts and P_0 is simply the whole image. Any partition of the image obtained from sets in P_0^3, such as the one represented above, can in turn be used to represent an image as a family of sub-probability measures, which reduces in the case of two-color images to binary histograms as illustrated in the right-most image.

Such meaningful partitions can be obtained in a general case if we assume the existence of a prior hierarchical information on the elements of $\mathcal{T}$, translated into a series

$$P_0 = \{\mathcal{T}\}, .., P_D = \{\{t\}, t \in \mathcal{T}\}$$

of partitions of $\mathcal{T}$, namely a hierarchy on $\mathcal{T}$. To provide a hierarchical content, the family $(P_d)_{d=1}^D$ is such that any subset present in a partition P_d is included in a (unique by definition of a partition) subset included in the coarser partition P_{d-1}, and further assume this inclusion to be strict. This is equivalent to stating that each set T of a partition P_d is divided in P_{d+1} through a partition of T which is not T itself. We note this partition $s(T)$ and name its elements the siblings of T. Consider now the subset $\mathcal{P}_D \subset \mathcal{P}(\mathcal{T})$ of all partitions of $\mathcal{T}$ obtained by using only sets in

$$P_0^D \stackrel{\text{def}}{=} \bigcup_{d=1}^{D} P_d,$$

namely $\mathcal{P}_D \stackrel{\text{def}}{=} \{P \in \mathcal{P}(\mathcal{T}) \text{ s.t. } \forall T \in P, T \in P_0^D\}$.. The set $\mathcal{P}_D$ contains both the coarsest and the finest resolutions, respectively P_0 and P_D, but also all variable resolutions for sets enumerated in P_0^D, as can be seen for instance in the third image of Figure 2.

2.3 Averaging Resolution Specific Kernels

Each partition P contained in $\mathcal{P}_D$ provides a resolution to compare two objects, and generates consequently a very large family of kernels k_P when P spans $\mathcal{P}_D$. Some partitions are probably better suited for certain tasks than others, which may call for an efficient estimation of an optimal partition given a task. We take in this section a different direction by considering an averaging of such kernels based on a Bayesian prior on the set of partitions. In practice, this averaging favours objects which share similarities under a large collection of resolutions.

Definition 1. *Let $\mathcal{T}$ be an index set endowed with a hierarchy $(P_d)_{d=0}^D$, π be a prior measure on the corresponding set of partitions $\mathcal{P}_D$ and k a base kernel on $M_+^s(\mathcal{X}) \times M_+^s(\mathcal{X})$. The multiresolution kernel k_π on $M_{\mathcal{T}}(\mathcal{X}) \times M_{\mathcal{T}}(\mathcal{X})$ is defined as*

$$k_\pi(\mu, \mu') = \sum_{P \in \mathcal{P}_D} \pi(P)\, k_P(\mu, \mu'). \tag{2}$$

Note that in Equation (2), each resolution specific kernel contributes to the final kernel value and may be regarded as a weighted feature extractor.

3 Kernel Computation

This section aims at characterizing hierarchies $(P_d)_{d=0}^D$ and priors π for which the computation of k_π is both tractable and meaningful. We first propose a type of hierarchy generated by trees, which is then coupled with a branching process prior to fully specify π. These settings yield a computational time for expressing k_π which is loosely upperbounded by $D \times \operatorname{card} \mathcal{T} \times c(k)$ where $c(k)$ is the time required to compute the base kernel.

3.1 Partitions Generated by Branching Processes

All partitions P of $\mathcal{P}_D$ can be generated iteratively through the following rule, starting from the initial root partition $P := P_0 = \{\mathcal{T}\}$. For each set T of P:

1. either leave the set as it is in P,
2. either replace it by its siblings enumerated in $s(T)$, and reapply this rule to each sibling unless they belong to the finest partition P_D.

By giving a probabilistic content to the previous rule through a binomial parameter (i.e. for each treated set assign probability $1-\varepsilon$ of applying rule 1 and probability ε of applying rule 2) a candidate prior for $\mathcal{P}_D$ can be derived, depending on the overall coarseness of the considered partition. For all elements T of P_D this binomial parameter is equal to 0, whereas it can be individually defined for any element T of the $D-1$ coarsest partitions as $\varepsilon_T \in [0,1]$, yielding for a partition $P \in \mathcal{P}_D$ the weight

$$\pi(P) = \prod_{T \in P} (1-\varepsilon_T) \prod_{T \in \overset{\circ}{P}} (\varepsilon_T),$$

where the set $\overset{\circ}{P} = \{T \in P_0^D \text{ s.t. } \exists V \in P, V \subsetneq T\}$ gathers all coarser sets belonging to coarser resolutions than P, and can be regarded as all ancestors in P_0^D of sets enumerated in P.

3.2 Factorization

The prior proposed in Section 3.1 can be used to factorize the formula in (2), which is summarized in this theorem, using notations used in Definition 1

Theorem 1. *For two elements m, m' of $M_{\mathcal{T}}(\mathcal{X})$, define for T spanning recursively $P_D, P_{D-1}, ..., P_0$ the quantity*

$$K_T = (1-\varepsilon_T) k_T(\mu, \mu') + \varepsilon_T \prod_{U \in\ s(T)} K_U.$$

Then $k_\pi(\mu, \mu') = K_{\mathcal{T}}$.

Proof. The proof follows from the prior structure used for the tree generation, and can be found in either [1] or [4]. Figure 3 underlines the importance of incorporating to each node K_T a weighted product of the kernels K_U computed by its siblings. □

If the hierarchy of $\mathcal{T}$ is such that the cardinality of $s(T)$ is fixed to a constant α for any set T, typically $\alpha = 4$ for images as seen in Figure 2, then the computation of k_π is upperbounded by $(\alpha^{D+1}-1)c(k)$. This computational complexity may even become lower in cases where the histograms become sparse at fine resolutions, yielding complexities in linear time with respect to the size of the compared objects, quantified by the length of the sequences in [4] for instance.

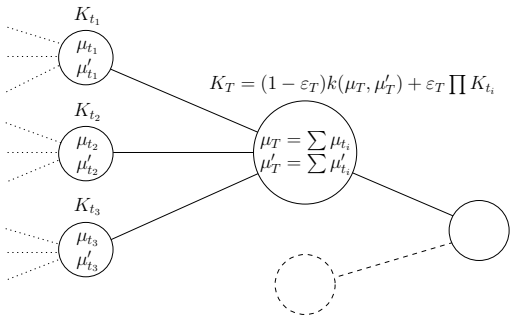

Figure 3: The update rule for the computation of $k\pi$ takes into account the branching process prior by updating each node corresponding to a set T of any intermediary partitions with the values obtained for higher resolutions in $s(T)$.

4 Experiments

We present in this section experiments inspired by the image retrieval task first considered in [2] and also used in [6], although the images used here are not exactly the same. The dataset was also extracted from the Corel Stock database and includes 12 families of labelled images, each class containing 100 color images, each image being coded as 256×384 pixels with colors coded in 24 bits (16M colors). The families depict *bears, African specialty animals, monkeys, cougars, fireworks, mountains, office interiors, bonsais, sunsets, clouds, apes* and *rocks and gems*. The database is randomly split into balanced sets of 900 training images and 300 test images. The task consists in classifying the test images with the rule learned by training 12 one-vs-all SVM's on the learning fold. The object are then classified according to the SVM performing the highest score, namely with a "winner-takes-all" strategy. The results presented in this section are averaged over 4 different random splits. We used the CImg package to generate histograms and the Spider toolbox for the SVM experiments[2].

We adopted a coarser representation of 9 bits per color for the $98,304$ pixels of each image, rather than the 24 available ones to reduce the size of the RGB color space to $8^3 = 512$ from the original set of $256^3 = 16,777,216$ colors. In this image retrieval experiment, we used localization as the conditioning index set, dividing the images into $1, 4, 4^2 = 16, 9$ and $9^2 = 81$ local histograms (in Figure 2 the image was for instance divided into $4^3 = 64$ windows). To define the branching process prior, we simply set an uniform value over all the grid of ε of $1/\alpha$, an usage motivated by previous experiments led in a similar context [4]. Finally, we used kernels described in both [2] and [6] to define the base kernel k. These kernels can be directly applied on sub-probability measures, which is not the case for all kernels on multinomials, notably the Information Diffusion Kernel [9]. We report results for two families of kernels, namely the Radial Basis Function expressed for multinomials and the entropy kernel based on the Jensen divergence [6, 3]:

$$k_{a,b,\rho}(\theta,\, \theta') = e^{-\rho \sum |\theta_i^a - \theta_i'^a|^b} \;\;,\;\; k_h(\theta,\, \theta') = e^{-h\left(\frac{\theta+\theta'}{2}\right) + \frac{1}{2}\left(h(\theta)+h(\theta')\right)}.$$

For most kernels not presented here, the multiresolution approach usually improved the performance in a similar way than the results presented in Table 1. Finally, we also report that using only the finest resolution available in each (α, D) setting, that is a branching process prior uniformly set to 1, yielded better results than the use of the coarsest histogram without achieving however the same performance of the multiresolution averaging framework, which highlights the interest of taking both coarse and fine perspectives into account.

[2] http://cimg.sourceforge.net/ and http://www.kyb.tuebingen.mpg.de/bs/people/spider/

When $a = .25$ for instance, this setting produced 16.5% and 16.2% error rates for $\alpha = 4$ and $D = 1, 2$, and 15.8% for $\alpha = 9$ and $D = 1$.

Kernel	RBF, $b = 1$, $\rho = .01$			JD
	$a = .25$	$a = .5$	$a = 1$	
global histogram	18.5	18.3	18.3	21.4
$D = 1, \alpha = 4$	15.4	16.4	18.8	17
$D = 2, \alpha = 4$	*13.9*	*13.5*	15.8	15.2
$D = 1, \alpha = 9$	14.7	14.7	16.6	15
$D = 2, \alpha = 9$	15.1	15.1	30.5	15.35

Table 1: Results for the Corel image database experiment in terms of error rate, with 4 fold cross-validation and 2 different types of tested kernels, the RBF and the Jensen Divergence.

Acknowledgments

MC would like to thank Jean-Philippe Vert and Arnaud Doucet for fruitful discussions, as well as Xavier Dupré for his help with the CImg toolbox.

References

[1] Olivier Catoni. *Statistical learning theory and stochastic optimization, Saint-Flour lecture notes*. Springer Verlag, 2001.

[2] O. Chapelle, P. Haffner, and V. Vapnik. Svms for histogram based image classification. *IEEE Transactions on Neural Networks*, (9), 1999.

[3] M. Cuturi, K. Fukumizu, and J.-P. Vert. Semigroup kernels on measures. submitted to *Journal of Machine Learning Research*.

[4] M. Cuturi and J.-P. Vert. The context-tree kernel for strings. *Neural Networks*, 2005. in press.

[5] David Haussler. Convolution kernels on discrete structures. Technical report, UC Santa Cruz, 1999. USCS-CRL-99-10.

[6] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In Z. Ghahramani and R. Cowell, editors, *Proceedings of AISTATS 2005*, January 2005.

[7] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[8] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer Academic Publishers, Dordrecht, 2002.

[9] John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163, January 2005.

[10] G. Rätsch and S. Sonnenburg. Accurate splice site prediction for caenorhabditis elegans. In B. Schoelkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.

[11] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization , Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.

[12] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in support vector machine. In *NIPS 14*, Cambridge, MA, 2002. MIT Press.

[13] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(Suppl 1):268–275, 2002.

[14] J.-P. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for protein sequences. In B. Schoelkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.