Applied Mathematics, 2015, 6, 484-495
Published Online March 2015 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2015.63046

Scientific Research Publishing

Finding the Asymptotically Optimal Baire Distance for Multi-Channel Data

Patrick Erik Bradley[1], Andreas Christian Braun[2]

[1]Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Remote Sensing and Landscape Information Systems, University of Freiburg, Freiburg, Germany
Email: patrick.bradley@kit.edu, andreas.braun@felis.uni-freiburg.de

Received 16 February 2015; accepted 6 March 2015; published 10 March 2015

Abstract

A novel permutation-dependent Baire distance is introduced for multi-channel data. The optimal permutation is given by minimizing the sum of these pairwise distances. It is shown that for most practical cases the minimum is attained by a new gradient descent algorithm introduced in this article. It is of biquadratic time complexity: Both quadratic in number of channels and in size of data. The optimal permutation allows us to introduce a novel Baire-distance kernel Support Vector Machine (SVM). Applied to benchmark hyperspectral remote sensing data, this new SVM produces results which are comparable with the classical linear SVM, but with higher kernel target alignment.

Keywords

***p*-Adic Numbers, Ultrametrics, Baire Distance, Support Vector Machine, Classification**

1. Introduction

The Baire distance was introduced to classification in order to produce clusters by grouping data in "bins" by [1]. In this way, they seek to find inherent hierarchical structure in data defined by their features. Now, if there are many different features associated with data, then it is reasonable to sort the feature vector by some criterion which ranks their contribution to this inherent hierarchical structure. We will see that there is a natural Baire distance associated to any given permutation of features. Hence, it is natural to ask for this task to be performed in reasonable time. In general, there is no efficient way of sorting n variables, but if the task is to find a permutation satisfying some optimality condition, then often a gradient descent algorithm can be applied. In that case, the run-time complexity is decreased considerably.

In this paper we introduce a permutation-dependent Baire distance for data with n features, and we define a

How to cite this paper: Bradley, P.E. and Braun, A.C. (2015) Finding the Asymptotically Optimal Baire Distance for Multi-Channel Data. *Applied Mathematics*, **6**, 484-495. http://dx.doi.org/10.4236/am.2015.63046

linear cost function depending on the pairwise Baire distances for all possible permutations. The Baire distance we use depends on a parapmeter ϵ, and we argue that the precise value of this parameter is seldom to be expected of interest. On the contrary, we believe that it practically makes more sense to vary this parameter and to study the limiting case $\epsilon \to 0$. Our theoretical result is that there is a gradient-descent algorithm which can find the asymptotic minimum for $\epsilon \to 0$ with a runtime complexity of $O\left(dn^2\right)$, where d is the number of all data pairs.

The Support Vector Machine (SVM) is a well known technique for kernel based classification. In kernel based classification, the similarity between input data is modelled by kernel functions. These functions are employed to produce kernel matrices. Kernel matrices can be seen as similarity matrices of the input data in reproducing kernel Hilbert spaces. Via optimization of a Lagrangian minimization problem, a subset of input points is found, which is used to produce a separating hyperplane for the data of various classes. The final decision function is dependent only on the position of these data in the feature space and does not require estimation of first or second order statistics on the data. The user has a lot of freedom on how to produce the kernel functions. This offers the option of producing individual kernel functions for the data.

As an application of our theoretical result, we introduce the new class of Baire-distance kernels which are functions of our parametrized Baire distance. For the asymptotically optimal permutation, the resulting Baire distance SVM yields results comparable with the classical linear SVM on the AVIS Indian Pine dataset. The latter is a well known hyperspectral remote sensing dataset. Furthermore, the kernel target alignment [2] represents an a priori quality assessment and favours our new Baire distance multi-kernel SVM constructed from Baire distance kernels at difference feature resolutions. This new multi-kernel combines in a sense our first approach with the approach of [1], as it combines the different resolutions defined by their method of "bin" grouping. As our preliminary practical result, we obtain greater completeness in many of our clusters than with the classical linear SVM clusters.

2. Ultrametric Distances for Multi-Channel Data

After a short review on the ultrametric parametrized Baire distance, it is shown how to find for n variables their asymptotically optimal permutation for a linear cost function defined by permutation-dependent Baire distances. It has quadratic run-time complexity, if the data size is fixed.

2.1. Baire Distance

Let x, y be words over an Alphabet A. Then the Baire distance is

$$d\left(x,y\right) = 2^{-\ell(x,y)}$$

where $\ell\left(x,y\right)$ is the length of the longest common initial subword, as depicted in **Figure 1**. The length of a word is defined as the number of letters from A (with multiple occurrences). The reason for choosing $\frac{1}{2}$ as the basis in the Baire distance is pure arbitrariness, at least to our opinion. Hence, $\frac{1}{2}$ can be replaced by any fixed ϵ in the interval $\left(0,1\right)$.

Definition 2.1. *The expression*

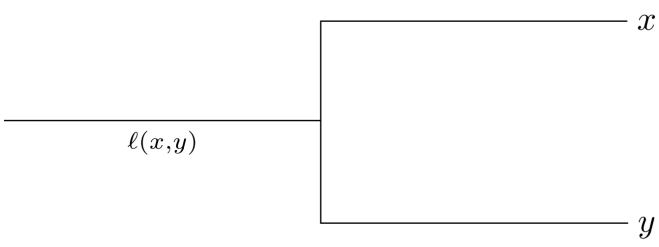

Figure 1. Two words with common initial subword.

$$d_\epsilon(x,y) = \epsilon^{\ell(x,y)}$$

is the ϵ *-Baire distance*.

Later on, we will study the limiting case $\epsilon \to 0$.

Remark 2.2. *The metrics* d_ϵ *are all equivalent in the sense that they generate the same topologies.*

The Baire distance is important for classification, because it is an ultrametric. In particular, the strict triangle inequality

$$d(x,y) \le \max\{d(x,z), d(z,y)\}$$

holds true. This is shown to lead to efficient hierarchical classification with good classification results [1] [3] [4].

Data representation is often related to some choice of alphabet. For instance, the distinction "Low" and "High" leads to $A = \{0,1\}$ and is used in [4]. The decimal representation of numbers yields $A = \{0,1,\cdots,9\}$ for the method in [1]. A very general encoding with arithmetic flavour is given by subsets $A \subseteq O_K$ inside the ring of integers inside a p-adic number field K, with all $a \in A$ different modulo $\mathfrak{m}_K$ [5]. No knowledge of p-adic number theory is required for what comes after the following Example 2.3. However, the interested reader may consult [6] for a first application of such mathematics in classification.

Example 2.3. The simplest example of p-adic number fields K in data representation is given by taking K as the field of 2-adic numbers $\mathbb{Q}_2$. Then $O_K = \mathbb{Z}_2$ is the ring of 2-adic integers, and as alphabet $A = \{0,1\}$. The numbers 0.1 represent the finite field $\mathbb{F}_2$ in a standard way which is often used when 2-adic numbers are written out as power series in 2, *i.e.* as finite or infinite binary numbers.

The role of the parameter ϵ in classification can be described as follows. Let $X \subseteq A$ be a set of words. Then X defines a unique dendrogram $D(X)$, and d_ϵ defines a metric dendrogram $D_\epsilon(X) = (D(X), d_\epsilon)$. Observe that $D_\epsilon(X)$ depends only on the metric d_ϵ. By equivalence of the Baire metrics, dendrograms $D_\epsilon(X)$ are tree-isomorphic for all ϵ. However, optimal classification results in general do depend on ϵ, as has been observed in Theorem 2 of [7], where the result is formulated for p-adic ultrametrics.

2.2. Optimal Baire Distance

Given data X and attributes $p = (p_1, \cdots, p_n)$ with possible values $V = \{v_1, \cdots, v_m\}$, then a permutation $\sigma \in S_n$, where S_n is the symmetric group of all permutations of the set $\{1, \cdots, n\}$, defines the expression

$$p^\sigma(x) := p_{\sigma(1)}(x)\, p_{\sigma(2)}(x) \cdots p_{\sigma(n)}(x)$$

i.e. a word with letters from the alphabet V. This yields the Baire distance $d_\epsilon^\sigma(x,y)$.

In order to determine a suitable permutation for the data, consider the average Baire distance. A high average Baire distance will arise if there is a large number of singletons, and branching is high up in the hierarchy. On the other hand, if there are lots of common initial features, then the average Baire distance will be low. In that case, clusters tend to have a high density, and there are few singletons. From these considerations, it follows that the task is to find a permutation $\sigma \in S_n$ such that

$$E^\sigma(\epsilon) = \sum_{x,y \in X} d_\epsilon^\sigma(x,y)$$

is minimal, leading to the optimal Baire distance d_ϵ^σ. Any method attempting to fulfil this task must overcome the problem that $|S_n| = n!$ is quite large for large n.

Let $P = (p_1, \cdots, p_n)$, written as $\{1, \cdots, n\}$. Expanding $E^\sigma(\epsilon)$ into powers of ϵ yields:

$$E^\sigma(\epsilon) = \sum_{\nu=0}^{n} \alpha_\nu^\sigma \epsilon^\nu \quad with \quad \alpha_\nu^\sigma = \sum_{k=0}^{n} \sum_{J \in \Sigma_\nu^k} m_{\sigma(J)}, \tag{1}$$

where $m_{\sigma(J)}$ is the number of data pairs (x,y) with identical values exclusively in the set $\sigma(J) = \{\sigma(j) \mid j \in J\}$. The inner sum is taken over the set

$$\Sigma_\nu^k = \left\{ J \in 2^P \middle| |J| = k, \lambda(J) = \nu \right\} \tag{2}$$

where $J = \{j_1 < \cdots < j_k\}$, and $\lambda(J) = \ell\left(w_{j_1} \cdots w_{j_k}, w_1 \cdots w_n\right)$ is the length of the common initial subword with the standard word $w_1 \cdots w_n$ obtained by defining an ordering on any arbitrary alphabet $A = \{w_1, \cdots, w_n\}$.

Some first properties of Σ_ν^k are listed in the following:

1. $\Sigma_\nu^k = \varnothing$ if $\nu > k$
2. $\Sigma_0^0 = \{\varnothing\}$
3. $\Sigma_0^1 = \{\{\{2\}, \{3\}, \cdots, \{n\}\}\}$
4. $\Sigma_0^2 = \{\{2,3\}, \cdots, \{2,n\}; \{3,4\}, \cdots, \{3,n\}; \cdots, \{n-1,n\}\}$
5. $\Sigma_n^n = \{\{1, \cdots, n\}\}$

These properties follow from Equation (2) above, and they imply some first properties of α_ν^σ:

$$\alpha_0^\sigma = m_\varnothing + \sum_{i \neq 1} m_{\sigma(\{i\})} + \sum_{2 \le i < j \le n} m_{\sigma(\{i,j\})} + \sum_{2 \le i < j < k \le n} m_{\sigma(\{i,j,k\})} + \cdots \tag{3}$$

$$\alpha_{n-1}^\sigma = m_{\{\sigma(1), \cdots, \sigma(n-1)\}} \tag{4}$$

$$\alpha_n^\sigma = m_{\{1, \cdots, n\}} \tag{5}$$

An important observation is that α_ν^σ depends only on the first $\nu+1$ permuted values $\sigma(1), \cdots, \sigma(\nu+1)$. This will be exploited in the following section, where it is shown how optimal permutations σ can be computed.

The following two examples list all values of

$$\alpha_{\nu,k}^\sigma := \sum_{J \in \Sigma_\nu^k} m_{\sigma(J)}$$

in the case $\sigma = \mathrm{id}$ for $n = 3, 4$. By effecting the permutation σ, one obtains the corresponding matrices $\left(\alpha_{\nu,k}^\sigma\right)$, and summing over the row labelled ν yields α_ν^σ.

Example 2.4. *Table for* $n = 3$ *and* $\sigma = \mathrm{id}$:

$n=3$	0	1	2	3
0	$m_\varnothing$	$m_{\{2\}} + m_{\{3\}}$	$m_{\{2,3\}}$	
1		$m_{\{1\}}$	$m_{\{1,3\}}$	
2			$m_{\{1,2\}}$	
3				$m_{\{1,2,3\}}$

Example 2.5. *Table for* $n = 4$ *and* $\sigma = \mathrm{id}$:

$n=4$	0	1	2	3	4
0	$m_\varnothing$	$m_{\{2\}} + m_{\{3\}} + m_{\{4\}}$	$m_{\{2,3\}} + m_{\{2,4\}} + m_{\{3,4\}}$	$m_{\{2,3,4\}}$	
1		$m_{\{1\}}$	$m_{\{1,3\}} + m_{\{1,4\}}$	$m_{\{1,3,4\}}$	
2			$m_{\{1,2\}}$	$m_{\{1,2,4\}}$	
3				$m_{\{1,2,3\}}$	
4					$m_{\{1,2,3,4\}}$

2.3. Finding Optimal Permutations

Let Δ be the simplex of n channels labelled by the set $N := \{1, \cdots, n\}$. The faces are given by subsets of N or, equivalently, by elements of the power set $\mathfrak{P}(N)$.

The function

$$E^{\sigma}(\epsilon)=\sum_{\nu\geq 0}\alpha_{\nu}^{\sigma}\epsilon^{\nu}$$

from Equation (1) is to be minimised, where $\sigma\in S_n$ is a permutation of the set N. A combinatorialtopological point of view appears to be helpful in the task. Namely, view the simplex Δ as a (combinatorial) simplicial complex. A *star* of an i-face $x\in\Delta$ is the set of ν-faces attached to x with $\nu\geq i$ (including x itself). The weak topology on Δ is generated by the stars.

To Δ is associated a graph Γ_{Δ} whose vertices are the faces, and an edge is given by a pair (v,v') consisting of an i-face v and an $i+1$-face v' such that v is a face of v'.

The counts $m_{J(\sigma)}$ appearing in Equation (1) define a function $m:\mathfrak{P}(N)\to N$, and this in turn yields weights on Γ_{Δ} in the following way:

$$w(v)=\sum_{U\in\mathrm{Star}(v)}m_U \qquad (\text{vertex weights}) \tag{6}$$

$$w(e)=w(v)-w(v'), \qquad \text{if } e=(v,v')(\text{edge weights}) \tag{7}$$

Observe that all edge weights are non-negative:

$$w(e)\geq 0$$

because $\mathrm{Star}(v')\subseteq\mathrm{Star}(v)$. The graph Γ_{Δ} is a directed acyclic graph with origin vertex $v_{\varnothing}$ and terminal vertex v_N.

An injective path $\gamma:v_{\varnothing}\rightsquigarrow v_J$ in Γ_{Δ} has a natural ϵ-length

$$\ell_{\epsilon}(\gamma)=\sum_{\mu=0}^{\nu-1}w(e_{\mu})\epsilon^{\mu}$$

where γ is given by the sequence of edges $(e_0,\cdots,e_{\nu-1})$.

Definition 2.6. *A permutation $\sigma\in S_n$ is said to be compatible with an injective path $\gamma:v_{\varnothing}\rightsquigarrow v_J$, if*

$$\{\sigma(i)\}=J_i\setminus J_{i-1} \tag{8}$$

where γ is given by the sequence of sets $J_0=\varnothing,\cdots,J_{\nu}=J$.

Lemma 2.7. *If σ is compatible with γ, then*

$$\ell_{\epsilon}(\gamma)=\sum_{\mu=0}^{\nu-1}\alpha_{\mu}^{\sigma}\epsilon^{\mu}$$

where the path γ is given as in Definition 2.6.

Proof. Let $e_{\mu}=(v_{\mu},v_{\mu+1})$ be an edge on γ given by the pair of sets J_{μ}, $J_{\mu+1}$. Then

$$w(e_{\mu})=\sum_{U\in\mathrm{Star}(v_{\mu})\setminus\mathrm{Star}(v_{\mu+1})}m_u$$

Assume that $\sigma=\mathrm{id}$ is compatible with γ. Then

$$\mathrm{Star}(v_{\mu})\setminus\mathrm{Star}(v_{\mu+1})=\left\{I\subseteq 2^N \,\middle|\, I\supseteq\{1,\cdots,\mu\},\mu+1\notin I\right\}=\bigcup_{k=0}^{n}\Sigma_{\mu}^{k} \tag{9}$$

from which the assertion follows for $\sigma=\mathrm{id}$ by summation over the edges along γ. For arbitrary σ compatible with γ the proof is analogue to this case. □

The following is an immediate consequence:

Corollary 2.8. *Let $\gamma:v_{\varnothing}\rightsquigarrow v_N$, and σ compatible with γ. Then*

$$\ell_{\epsilon}(\gamma)+m_N\epsilon^N=E^{\sigma}(\epsilon)$$

The minimising $E^{\sigma}(\epsilon)$ can be found by travelling along a shortest path from $v_{\varnothing}$ to v_E. One method for finding such shortest paths is given by the well known Dijkstra algorithm.

Corollary 2.9. *Dijkstra's shortest path algorithm on* Γ_Δ *finds the global minima for* $E^\sigma(\epsilon)$ *with any given* $\epsilon \in (0,1)$.

The main problem with applying Corollary 2.9 is the size of Γ_Δ for large n. However, we believe that it is of practical interest to consider $E^\sigma(\epsilon)$ for sufficiently small ϵ. We will show below that in this case, the following *gradient descent* finds the global minimum in an exhaustive manner. Given an edge $e=(v,w)$, the expression $o(e)$ will denote the origin vertex v, and $t(e)$ means the terminal vertex w.

Algorithm 2.10. (Gradient descent) *Input.* (Γ_Δ, w).

Step 0. Set $V_0 := \{v_\varnothing\}$ and $E_0 = \{\text{all edges of } \Gamma_\Delta\}$.

Step 1. Collect in E_1 all edges e with $o(e) \in V_0$ having smallest weight $w(e)$, and set $V_1 := \{t(e) | e \in E_1\}$.

Step $\nu > 1$. Collect in E_ν all edges $e \in E_{\nu-1}$ with $o(e) \in V_{\nu-1}$ having smallest weight, and set $V_\nu := \{t(e) | e \in E_\nu\}$.

Output. The subgraph of Γ_Δ containing all paths with smallest sum of edge weights from $v_\varnothing$ to v_N.

This algorithm clearly terminates after n steps. The paths $\gamma : v_\varnothing \rightsquigarrow v_N$ correspond bijectively to a set $\mathcal{S}$ of permutations $\sigma \in S_n$.

Lemma 2.11. *Let* $\sigma \in \mathcal{S}$ *be a permutation derived from gradient descent,* $\epsilon \in (0,1)$, *and* $\tau \in S_n$ *such that* $E^\tau(\epsilon)$ *is minimal. Then there exists a constant* $C_\tau > 0$ *such that for all* $0 < \eta < C_\tau$ *it holds true that* $E^\sigma(\eta) \le E^\tau(\eta)$.

Proof. We may assume that there exists some $\nu \in \{1, \cdots, n\}$ such that

$$\alpha_\nu^\tau < \alpha_\nu^\sigma \tag{10}$$

as otherwise $C_\tau := \epsilon$ can be chosen. Assume now further that ν be minimal with property (10). Still further, we may assume that there exists some $\mu \in \{1, \cdots, \nu-1\}$ such that

$$\alpha_\mu^\sigma < \alpha_\mu^\tau \tag{11}$$

as otherwise σ could not be derived by gradient descent. The reason is that at step ν that method would descend down to $\tau(\nu)$ instead of to $\sigma(\nu)$, since ν is the first occurrence of property (10). Let now μ be minimal with (11). All this implies that

$$P_\tau(t) = \frac{E^\tau(t) - E^\sigma(t)}{t^\mu}$$

is a polynomial with real coefficients such that $P_\tau(0) > 0$. Hence, by continuity of $P_\tau(t)$, there exists a small neighbourhood of 0 on which $P_\tau(t)$ is still positive. This neighbourhood defines the desired constant C_τ. □

An immediate consequence of the lemma is that gradient descent is asymptotically the method of choice:

Theorem 2.12. *There exists a constant* $C \in (0,1)$ *such that gradient descent on* Γ *finds a global minimum for the cost function* $E^\sigma(\epsilon)$ *whenever* $0 < \epsilon < C$.

Proof. Let T be the set of all $\tau \in S_n$ for which $E^\tau(\epsilon)$ is minimal with some fixed ϵ. Then $C := \min\{C_\tau | \tau \in T\}$ has the desired property. □

The competitiveness of the gradient descent method is manifest in the following Remarks:

Remark 2.13. *Algorithm* 2.10 *is of run-time complexity at most* $O(n^2)$.

Proof. In the first step, there are n choices for possible edges to follow, and after n steps the possible permutations are found. Finding the minimal edge in step ν can be done with complexity $O(\nu)$. This proves the upper bound. □

Notice that the efficiency holds only for the case that the weights w of Γ_Δ are already given. However, this cannot be expected in general. Therefore, we investigate here the computational cost for $w(\gamma)$ for a gradient descent path γ in Γ_Δ. The following is immediate:

Lemma 2.14. *Let* $\gamma : v_0 \rightsquigarrow v_J$. *Then* $\mathrm{Star}(v_\mu) \setminus \mathrm{Star}(v_{\mu+1})$ *is a* (*combinatorial*) *simplex of dimension* $n-1-\mu$.

We will write $\Delta(e)$ for the simplex coming from an edge $e = (v_\mu, v_{\mu+1})$ as in Lemma 2.14. An immediate consequence is

$$w(e) = \sum_{x \in \Delta(e)} m(x) \tag{12}$$

the computation of which seems at first sight exponential in the dimension of $\Delta(e)$. In particular, the weights of the n very first edges $(v_\varnothing, v_1)$ look to be very cumbersome to compute. The problem is the function $m : 2^N \to \mathbb{N}$ with its computational cost $O(nd)$ for each $I \in 2^N$, where $d = \left| X^2 \right|$. Slightly more efficient is the function

$$c : 2^N \to \mathbb{N}, \quad I \mapsto \left| \left\{ x = (x_1, x_2) \in X^2 \,\middle|\, \forall i \in I : i(x_1) = i(x_2) \right\} \right|$$

which counts all pairs $x \in X^2$ on which the channels in I coincide. A trivial, but important observation is

$$c(I) = \sum_{J \supseteq I} m(J) \tag{13}$$

as this allows to define a nice way of computing the weight $w(e)$:

Lemma 2.15. *Let* $I \in 2^N$ *be a vertex. Then for any edge* e *with origin* $o(e) = I$ *and terminus* $t(e) = J$ *it holds true that*

$$w(e) = c(I) - c(J)$$

Proof. This is an immediate consequence of the identity

$$\Delta(e) = \mathrm{Star}(I) \setminus \mathrm{Star}(J)$$

which follows from Lemma 2.14. □

Assume now that we are given for each pair $x = (x_1, x_2) \in X^2$ the subset $I_x \in 2^N$ on which x_1 and x_2 coincide. Let $Z := X^2$ be the set of all pairs, and $Y \subseteq Z$. Then define for $I \in 2^N$ the set of pairs

$$Y(I) := \left\{ x \in Y \,\middle|\, I(x_1) = I(x_2) \right\}$$

and its corresponding cardinality

$$c_Y(I) := \left| Y(I) \right|$$

together with the conventions

$$Y(i, \cdots, j) := Y(\{i, \cdots, j\}), \quad c_Y(i, \cdots, j) := c_Y(\{i, \cdots, j\})$$

Then the identity

$$c_Z(i) - c_Z(i, j) = c_{Z(i)}(\varnothing) - c_{Z(i)}(j) \tag{14}$$

is immediate. Its usefulness is that the right hand side is computed more quickly than the left hand side:

Lemma 2.16 *The cost of* $c_Y(I)$ *is at most* $O(|Y| \cdot |I|)$.

Proof. Take each $i \in I$ and check coincidence of each $y \in Y$ in channel i . □

Algorithm 2.17 *Input.* $Z = X^2$, N , Γ_Δ , $J_0 = \varnothing$.

Step 1. Find minimal edge $e : \varnothing \rightsquigarrow \{i\}$ with

$$w(e) = c_Z(\varnothing) - c_Z(i) \tag{15}$$

minimal. Set $Z := Z(i)$, $N := N \setminus \{i\}$, $J_1 := J_0 \cup \{j\}$.

Step $\nu > 1$. Repeat Step 1 with current values of Z and N , if both sets are non-empty. Set $J_\nu = J_{\nu-1} \cup \{i\}$ with current value of i .

Output. Path $\gamma : \varnothing \rightsquigarrow J_\mu$ for some $\mu \le n$.

Theorem 2.18 *Algorithm* 2.17 *has run-time complexity at most* $O\left(n^2 \cdot d\right)$.

Proof. The complexity in Step ν is at most $O\left(d_\nu \cdot (n-\nu)\right)$ for $d_\nu = |Z_\nu|$ with the Z_ν being the Z at that step. The reason is that, according to (15) and Lemma 2.16, $w(e)$ has complexity $O\left(|Z|_\nu\right)$, and there are $n-\nu$ edges going out of vertex v_ν. Bounding the cardinalities of Z_ν by d from above, and summing the costs yields the desired bound $O\left(n^2 \cdot d\right)$. □

Notice that the constant C of Theorem 2.12 can be very close to zero. That would mean that the gradient descent method yields only a local minimum for most values of ϵ. However, we believe that there is no polynomial-time algorithm which finds a minimum which is global for all ϵ, or at least for all ϵ below a predescribed threshold.

3. Combining Ultrametrics with SVM

Within this section the potential of integrating ultrametrics into state-of-the art classifiers—the Support Vector Machine (SVM) as introduced by [8]—is presented. SVM has been intensely applied for classification tasks in remote sensing and several methodological comparisons have been established in previous work of the authors [9] [10]. At first, our methodology is outlined. Secondly, a classification result for a standard benchmark from hyperspectral remote sensing is shown.

3.1. Methodology

Kernel matrices are the representation of similarity between the input data used for SVM classification. To integrate ultrametrics into SVM classification the crucial step is therefore to create a new kernel function [11] [12]. Instead of representing the Euclidean distance between input data, this new kernel function represents the Baire distance between them. To have an optimal kernel based on the Baire distance, at first an optimal permutation σ is found as outlined in Section 2.3 by using Algorithm 2.17. The new kernel is thus given as

$$K_\sigma\left(x_i, x_j\right) = d^\sigma\left(x_i, x_j\right)^{-1} \tag{16}$$

where $d^\sigma = d_\epsilon^\sigma$ for some choice of $\epsilon \in (0,1)$ sufficiently small, and we call it a *Baire distance kernel*.

This new kernel function could be used for classification directly. However, one feature of kernel based classification is that multiple kernel functions can be combined to increase classification performance [13]. The Baire distance is dependent on the resolution (bitrate) of the data. Two very similar features will maintain a large ℓ_σ-value at high bit depths, while the value of ℓ_σ of less similar features will deteriorate at higher bit-rates. Thus, by varying the bit depth of the data, one obtains additional information about the similarity of the data. Therefore, a kernel is to be created which incorporates the information about similarity at each resolution. At first, data with 8-bit depth are used. An optimal σ_8 is computed as described in Section 2.3. Afterwards, a kernel K_{σ_8} is computed, which includes the Baire distance between features for the given σ at 8 bit. In the next step, data are compressed to 7-bit depth. Again, an optimal σ_7 is found, a new kernel is computed and the kernels are summed up. For bit depths $b \in \{1,\cdots,8\}$ kernels are computed and summed to the multiple Kernel K_{mult}.

$$K_{\text{mult}} = \sum_{b=1}^{8} K_{\sigma_b}\left(x_i, x_j\right) \tag{17}$$

This multiple kernel also belongs to the new class of Baire distance kernels and has the advantage of incorporating the similarity at different bit depths. It is compared against the standard linear kernel frequently used for SVM:

$$K_{\text{lin}} = \left\langle x_i, x_j \right\rangle \tag{18}$$

where the bracket $\langle \cdot, \cdot \rangle$ denotes the standard scalar product on the Euclidean space into which the data is mapped.

3.2. Application

Within this section, a comparison on a standard benchmark dataset from hyperspectral remote sensing is presented, cf. also [14]. The AVIRIS Indian Pines dataset consists of a 145×145 pixel hyperspectral image

with 220 spectral channels (**Figure 2**). It is well known due to the complexity of the classification problem it represents. The 16 land use classes consisting mainly of crop classes are to be separated. These are difficult to separate since they are spectrally very similar (due to the early phenological stage of the vegetation).

Although our implementation of Algorithm 2.17 is capable to process 220 features, only the first six principal components are considered. The reason is that there are two sources of coincidences. The first is coincidence due to spectral similarity of land cover classes (signal), the second is coincidence due to noise. For this work, only the coincidence of signal is relevant. Since the algorithm is not fit to distinguish between the two sources, only the six first principal components are considered relevant. They explain 99.66% of the sum of eigenvalues and are therefore believed to contribute considerably to coincidences due to signal and only marginally to coincidence due to noise.

At first, the dataset is classified with a linear kernel SVM as given in Equation (18). A visual result can be seen in **Figure 3** (left). The overall accuracy yielded is 53.5% and the κ-coefficient is 0.44. As can be seen, the dataset requires more complex kernel functions than linear ones. Then, a multiple kernel K_{mult} of the form (16) is computed as described in Section 3.1. The dataset is again classified using an SVM, and a visual result can be seen in **Figure 3** (right). The overall accuracy yielded is 53.7% and the κ-coefficient is 0.45.

Figure 2. Hyperspectral image.

(a) (b)

Figure 3. (a) Linear SVM; (b) Multi-Baire-kernel SVM.

The overall accuracy is the percentage of correctly classified pixels from the reference data. The κ-coefficient is a statistical measure of the agreement, beyond chance, between the algorithm's results and the manual labelling in the reference data. Both are global measurements of performance.

As can be seen, both results have a lot of resemblance in the major part. However, the result produced with the linear kernel tends to confuse the brown crop classes in the north with green pasture classes. On the other hand, the linear kernel SVM better recognizes the street in the Western part of the image.

The kernel target alignment between these kernels and the ideal kernel

$$K_{\text{ideal}}\left(x_i, x_j\right) = \delta_{L_i, L_j}$$

was computed. The ideal kernel is defined via the label L associated to each pixel, and has value 1 if the labels coincide, otherwise its value is zero. Note that the kernel target alignment proposed by [2] represents an a-priori quality assessment of a kernel's suitability. It is defined as

$$A\left(K, K_{\text{ideal}}\right) := \frac{\left\langle K, K_{\text{ideal}}\right\rangle_G}{\sqrt{\left\langle K, K\right\rangle_G \cdot \left\langle K_{\text{ideal}}, K_{\text{ideal}}\right\rangle_G}}$$

where

$$\left\langle M, N\right\rangle_G = \sum_{i,j=1}^{n} M\left(x_i, x_j\right) N\left(x_i, x_j\right)$$

denotes the usual scalar product between Gram matrices.

The kernel target alignment takes values in the interval $[-1, +1]$ with one being the best. The kernel target alignment of K_{lin} was 0.37. The K_{mult} yielded a higher alignment of 0.47 thus giving reason for expecting a higher overall performance of the latter. The producers' accuracies (pa) and users' accuracies (ua) for the individual classes are shown in **Table 1** and **Table 2**.

The users' accuracy shows what percentage of a particular ground class was correctly classified. The producers' accuracy is a measure of the reliability of an output map generated from a classification scheme which tells what percentage of a class truly corresponds to a class in the reference. Both are local (*i.e.* class-dependent) measurements of performance.

Table 1. Hyperspectral image (channels R:25, G:12, B:1).

Value	C1	C2	C3	C4	C5	C6	C7	C8
pa(Kmult)	0	46.6	12.6	1.2	17.5	82.9	0	99.1
pa(Klin)	0	39.9	0.8	2.4	49.1	80.4	0	99.1
pa(Kmult) − pa(Klin)	0	6.7	11.8	−1.2	−31.6	2.5	0	0
ua(Kmult)	0	43.0	33.4	20.0	74.3	72.8	0	75.8
ua(Klin)	0	38.9	45.4	28.5	57.1	78.5	0	84.5
ua(Kmult) − ua(Klin)	0	4.1	−12.0	−8.5	17.2	−5.7	0	−8.7

Table 2. Hyperspectral image (continued).

Value	C9	C10	C11	C12	C13	C14	C15	C16
pa(Kmult)	0	10.1	80.6	4.6	90.5	90.2	15.4	63.6
pa(Klin)	0	0.1	88.0	1.1	91.8	86.5	15.0	84.8
pa(Kmult) − pa(Klin)	0	10.0	−7.4	3.5	−1.3	3.7	0.4	−21.2
ua(Kmult)	0	38.9	46.3	32.2	54.0	68.8	45.5	93.3
ua(Klin)	0	50.0	43.7	12.8	56.6	72.5	65.5	86.1
ua(Kmult) − ua(Klin)	0	−11.1	2.6	19.4	−2.6	−3.7	−20.0	7.2

As had to be expected, each classification approach outperformed the other for some classes. The approach based on K_{mult} yields higher producers' accuracy values than K_{lin} in seven cases. For five cases, K_{lin} is superior. For users' accuracy, K_{mult} is superior in five cases, K_{lin} in eight cases.

Since producers' accuracy outlines which amount of pixels from the reference are found in the classification (completeness) while users' accuracy outlines which amount of the pixels in one class are correct, it can be concluded, that the proposed approach produces more complete results for many classes than with the standard linear kernel approach. Of course, due to the low overall accuracy values yielded, the approach should be extended by applying e.g. Gaussian functions over the similarity matrices.

4. Conclusion

Finding optimal Baire distances defined by permutations of n variables can be done in quadratic time, if the data size is fixed and a gradient descent algorithm is used. For the Baire distance parametrised by the base ϵ, this becomes the global minimum if ϵ is sufficiently small. In practice the outcome will be not a unique permutation, but a more or less large set $\mathcal{S}$ of optimal permutations σ. The σ can be viewed in a natural way as words over some alphabet. This implies that the symmetric group S_n of the n variables has a well-defined dendrogram $D(S_n)$ in which we can view $\mathcal{S}$ as a cluster. The common initial word $w_{\mathcal{S}}$ of $\mathcal{S}$ defines a ranking of $\ell = \ell(w_{\mathcal{S}}) \leq n$ of the variables which we conjecture to contain the most relevant inherent hierarchical information of the dataset X, after removing variables with very small variation. We expect further hierarchical information about X by finding optimal classifications of $\mathcal{S}$ with respect to the ultrametric defined by its dendrogram $D(\mathcal{S})$. Apart from theoretically providing an algorithm which finds the optimal permutation, the applicability of the methodology was demonstrated. To this end, an initial experiment to integrate the Baire distance into state-of-the-art SVM classification is provided. By defining a new multiple kernel function based on Baire distances, classification accuracy on a benchmark dataset is increased. This finding emphasizes the usefulness of the optimal Baire distance in classification. In future work, Gaussian kernels based on the Baire distance will be studied. Furthermore, unsupervised classification algorithms using the permutation-dependent ultrametrics will be dealt with in future work.

Acknowledgements

This work has grown out of a talk given at the International Conference on Classification (ICC 2011) and the discussions afterwards. The first author is funded by Deutsche Forschungsgemeinschaft (DFG), and the second author by the Deutsches Zentrum für Luft-und Raumfahrt e.V. (DLR). Thanks to Fionn Murtagh, Roland Glantz, and Norbert Paul for valuable conversation, as well as Fionn Murtagh and David Wishart for the organising of the International Conference on Classification (ICC 2011) in Saint Andrews, Scotland. The article processing charge was funded by the German Research Foundation (DFG) and the Albert Ludwigs University Freiburg in the funding programme Open Access Publishing.

References

[1] Contreras, P. and Murtagh, F. (2010) Fast Hierarchical Clustering from the Baire Distance. In: Locarek-Junge, H. and Weighs, C., Eds., *Classification as a Tool for Research*, Springer Berlin Heidelberg, Germany, 235-243. http://dx.doi.org/10.1007/978-3-642-10745-0_25

[2] Cristianini, N., Shawe-Taylor, J., Elisseeff, A. and Kandola, J. (2006) On Kernel Alignment. In: Holmes, D.E., Ed., *Innovations in Machine Learning*, Springer Berlin Heidelberg, Germany, 205-256.

[3] Murtagh, F. (2004) On Ultrametricity, Data Coding, and Computation. *Journal of Classification*, **21**, 167-184. http://dx.doi.org/10.1007/s00357-004-0015-y

[4] Benois-Pineau, J., Khrennikov, A.Y. and Kotovich, N.V. (2001) Segmentation of Images in *p*-Adic and Euclidean Metrics. *Doklady Mathematic*, **64**, 450-455.

[5] Bradley, P.E. (2010) Mumford Dendrograms. *The Computer Journal*, **53**, 393-404. http://dx.doi.org/10.1093/comjnl/bxm088

[6] Bradley, P.E. (2008) Degenerating Families of Dendrograms. *Journal of Classification*, **25**, 27-42. http://dx.doi.org/10.1007/s00357-008-9009-5

[7] Bradley, P.E. (2009) On *p*-Adic Classification. *p-Adic Numbers, Ultrametric Analysis, and Applications*, **1**, 271-285.

[8] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297.

http://dx.doi.org/10.1007/BF00994018

[9] Braun, A.C. (2010) Evaluation of One-Class SVM for Pixe-Based and Segment-Based Classification in Remote Sensing. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **38**, 160-165.

[10] Braun, A.C., Weidner, U., Jutzi, B. and Hinz, S. (2011) Integrating External Knowledge into SVM Classification—Fusing Hyperspectral and Laserscanning Data by Kernel Composition. In: Heipke, C., Jacobsen, K., Rottensteiner, F., Müller, S. and Sörgel, U., Eds., *High-Resolution Earth Imaging for Geospatial Information*, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **38**, Part 4/W19 (on CD).

[11] Amari, S. and Wu, S. (1999) Improving Support Vector Machine Classifiers by Modifying Kernel Functions. *Neural Networks*, **12**, 783-789. http://dx.doi.org/10.1016/S0893-6080(99)00032-5

[12] Braun, A.C., Weidner, U., Jutzi, B. and Hinz, S. (2012) Kernel Composition with the One-against-One Cascade for Integrating External Knowledge into SVM Classification. *Photogrammetrie-Fernerkundung-Geoinformation*, **2012**, 371-384. http://dx.doi.org/10.1127/1432-8364/20/0124

[13] Sonnenburg, S., Rätsch, G., Schäfer, C. and Schölkopf, B. (2006) Large Scale Multiple Kernel Learning. *The Journal of Machine Learning Research*, **7**, 1531-1565.

[14] Braun, A.C., Weidner, U. and Hinz, S. (2010) Support Vector Machines for Vegetation Classification? A Revision. *Photogrammetrie-Fernerkundung-Geoinformation*, **2010**, 273-281. http://dx.doi.org/10.1127/1432-8364/2010/0055