Abstracts of the FLLL/SCCH Master and PhD Seminar

Johannes Kepler University, HS13

June 29, 2005

Software Competence Center Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz-Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

Session 1 (Chair: Mario Drobics) 14:00–15:30

14:00 Ulrich Bodenhofer, Mustafa Demirci:
Strict Fuzzy Orderings in a Similarity-Based Setting

14:30 Bernhard Moser:
On the Interrelationship of Kernels in Machine Learning and Fuzzy Similarity Relations

15:00 Werner Groißböck:
A Nonlinear Approximation Formula Generator for Very High Dimensional Data Based on Variable Selection and Genetic Programming

15:30 Coffee Break

Session 2 (Chair: Werner Groißböck) 16:00–17:30

16:00 Mario Drobics, János Botzheim:
An Bacterial Evolutionary Algorithm for Feature Selection

16:30 Leila Muresan, Bettina Heise:
Analysis of Microarray Images

17:00 Bettina Heise, Leila Muresan:
Image Segmentation for DIC Images of Cells

Technical Report
SCCH–TR–0504

Strict Fuzzy Orderings in a Similarity-Based Setting

Ulrich Bodenhofer
Software Competence Center Hagenberg
ulrich.bodenhofer@scch.at

Mustafa Demirci
Dept. of Mathematics, Faculty of Sciences and Arts
Akdeniz University, Antalya, Turkey
demirci@akdeniz.edu.tr

Abstract — This paper introduces and justifies a similarity-based concept of strict fuzzy orderings and provides constructions how fuzzy orderings can be transformed into strict fuzzy orderings and vice versa. We demonstrate that there is a meaningful correspondence between fuzzy orderings and strict fuzzy orderings. Unlike the classical case, however, we do not obtain a general one-to-one correspondence. We observe that the strongest results are achieved if the underlying t-norm induces a strong negation, which, in particular, includes nilpotent t-norms and the nilpotent minimum.

Key words — *fuzzy equivalence relations, fuzzy orderings, strict fuzzy orderings.*

Proc. Joint 4th Int. Conf. in Fuzzy Logic and Technology and 11th French Days on Fuzzy Logic and Applications, Barcelona,
September 7–9, 2005. (accepted).

Kplus
Kompetenzzentren-Programm

1 Introduction

In the classical case, there is a one-to-one correspondence between partial orderings, i.e. reflexive, antisymmetric, and transitive relations, and strict orderings, i.e. irreflexive and transitive relations. The only trivial component that distinguishes these two concepts is equality. From that point of view, it makes no fundamental difference whether we consider one or the other [20].

Orderings and strict orderings have been studied in the theory of fuzzy relations already as well [11, 17, 18, 23]. Partial fuzzy orderings in the sense of Zadeh [23], however, have severe shortcomings that were finally resolved by replacing the crisp equality by a fuzzy equivalence relation, thereby maintaining the well-known classical fact that orderings are obtained from preorderings by factorization [1, 2, 3, 10, 13]. Strict fuzzy orderings based on such a similarity-based setting, however, have not yet been considered so far. This paper aims at filling this gap. We introduce similarity-based strict fuzzy orderings and provide constructions how fuzzy orderings can be transformed into strict fuzzy orderings and vice versa. We will see that, unlike the classical case, the two concepts remain independent to some extent in the sense that there is no general one-to-one correspondence. The reason is for that is twofold: (1) the underlying fuzzy equivalence relation is a much richer structure than the classical equality; (2) the underlying logical operations do not form a Boolean algebra, thus, we do not have the guarantee that all constructions are reversible.

2 Preliminaries

All (fuzzy) relations considered in this paper are binary (fuzzy) relations on a given non-empty domain X. For simplicity, we consider the unit interval $[0,1]$ as our domain of truth values in this paper. Note that most results, with only minor and obvious modifications, also hold for more general structures [9, 10, 12, 13, 15, 19]. The symbol T denotes a left-continuous t-norm [16]. Correspondingly, $\vec{T}$ denotes the unique residual implication of T. Furthermore, we denote the residual negation of T with $N_T(x) = \vec{T}(x,0)$. If the residual negation N_T of T is a strong negation (i.e. a continuous, strictly decreasing, and involutive negation), we denote the dual t-conorm (w.r.t. the residual negation N_T) with

$$S_T(x,y) = N_T(T(N_T(x),N_T(y))).$$

In any case, we assume that the reader is familiar with the basic concepts and properties of triangular norms and related operations [11, 16].

Definition 1. A binary fuzzy relation E is called *fuzzy equivalence relation*[1] with respect to T, for brevity T-*equivalence*, if the following three axioms are fulfilled for all $x,y,z \in X$:

1. Reflexivity: $E(x,x) = 1$
2. Symmetry: $E(x,y) = E(y,x)$
3. T-transitivity: $T(E(x,y),E(y,z)) \leq E(x,z)$

Definition 2. A binary fuzzy relation L is called *fuzzy ordering* with respect to T and a T-equivalence E, for brevity T-E-*ordering*, if it fulfills the following three axioms for all $x,y \in X$:

[1] Note that various diverging names for this class of fuzzy relations appear in literature, like similarity relations, indistinguishability operators, equality relations, and several more [5, 9, 14, 15, 19, 21, 23]

1. E-reflexivity: $E(x,y) \leq L(x,y)$
2. T-E-antisymmetry:
$$T(L(x,y),L(y,x)) \leq E(x,y)$$
3. T-transitivity: $T(L(x,y),L(y,z)) \leq L(x,z)$

Definition 3. A fuzzy relation R is called *strongly complete* if $\max(L(x,y),L(y,x)) = 1$ for all $x,y \in X$ [4, 11, 17]. R is called T-linear if $N_T(L(x,y)) \leq L(y,x)$ for all $x,y \in X$ [4, 13].

Note that strong completeness implies T-linearity, regardless of the choice of T [4]. If N_T is a strong negation, then a fuzzy relation R is T-linear if and only if $S_T(R(x,y),R(y,x)) = 1$ holds for all $x,y \in X$ [4].

3 Strict Fuzzy Orderings

In the crisp case, strict orderings are defined as irreflexive and transitive relations. It is more than obvious how to translate this definition to a fuzzy setting [11, 18]. In order to take the underlying fuzzy equivalence relation into account, we add extensionality.

Definition 4. A binary fuzzy relation R is called *strict fuzzy ordering* with respect to T and a T-equivalence E, for brevity *strict T-E-ordering*, if it fulfills the following axioms for all $x,x',y,y',z \in X$:

1. Irreflexivity: $R(x,x) = 0$
2. T-transitivity: $T(R(x,y),R(y,z)) \leq R(x,z)$
3. E-extensionality:
$$T(E(x,x'),E(y,y'),R(x,y)) \leq R(x',y')$$

Note that, under the assumption of T-transitivity, irreflexivity implies T-asymmetry, i.e. that $T(R(x,y),R(y,x)) = 0$ for all $x,y \in X$, where the converse holds only if T does not have zero divisors. In other words, irreflexivity can be replaced equivalently by T-asymmetry if T does not have zero divisors. Furthermore, we can conclude that $T(E(x,y),R(x,y)) = 0$ holds for all $x,y \in X$ and any strict T-E-ordering R.

Example 1. It is a well-known fact that

$$E(x,y) = \max(1 - |x-y|, 0)$$

is a $T_{\mathbf{L}}$-equivalence on $\mathbb{R}$ [7, 21], with $T_{\mathbf{L}}(x,y) = \max(x+y-1,0)$ being the Łukasiewicz t-norm. It is easy to show that

$$L(x,y) = \max(\min(1-x+y,1),0)$$

is a strongly complete $T_{\mathbf{L}}$-E-ordering [2, 3] and that

$$R(x,y) = \max(\min(y-x,1),0)$$

is a strict $T_{\mathbf{L}}$-E-ordering.

E-extensionality as defined above is nothing else but a straightforward translation of the trivial crisp assertion

$$(x = y \wedge x' = y' \wedge x < y) \rightarrow x' < y'.$$

In case that E is the classical crisp equality, E-extensionality is trivially fulfilled and we end up in the more traditional concept of a strict fuzzy ordering [11, 18]. Conversely, given an irreflexive and T-transitive fuzzy relation, we can make it E-extensional by the following proposition.

Proposition 1. *Let R be an irreflexive and T-transitive fuzzy relation. Then the following fuzzy relation (the* extensional interior of R w.r.t. E*) is a strict T-E-ordering:*

$$\begin{aligned}&\mathrm{Int}_{T,E}[R](x,y)\\&\quad= \inf_{x',y' \in X} \vec{T}(T(E(x,x'),E(y,y')),R(x',y'))\end{aligned}$$

Note that, as the following example suggests, the extensional interior as used in Proposition 1 does not necessarily give a meaningful non-trivial result.

Example 2. The classical strict linear ordering of real numbers $<$ is, of course, an irreflexive and T-transitive fuzzy relation (no matter what t-norm T we choose). Given E from Example 1, we obtain $\mathrm{Int}_{T_{\mathbf{L}},E}[<] = R$ (with R from Example 1). Now let us consider the product t-norm $T_{\mathbf{P}}(x,y) = x \cdot y$. It is well-known that

$$E'(x,y) = \exp(-|x-y|)$$

is a $T_{\mathbf{P}}$-equivalence [7]. However, we obtain that $\mathrm{Int}_{T_{\mathbf{P}},E'}[<]$ is the empty relation, i.e., for all $x,y \in X$,

$$\mathrm{Int}_{T_{\mathbf{P}},E'}[<](x,y) = 0.$$

4 From Fuzzy Orderings to Strict Fuzzy Orderings and Back

In the crisp case, the mutual definability of strict orderings from partial orderings and vice versa is a trivial matter: Given a partial ordering $\leq$, the corresponding strict ordering can be defined as

$$x \leq y \wedge x \neq y$$

or equivalently

$$x \leq y \wedge y \not\leq x.$$

Conversely, given a strict ordering $<$, the relation

$$x < y \vee x = y$$

is a partial ordering. These two constructions are exactly inverse to each other. The question arises whether and how these simple constructions can still be preserved in the more general fuzzy case. The following proposition clarifies the first direction.

Proposition 2. *Consider a T-equivalence E and a T-E-ordering L. Then the following fuzzy relation is a strict T-E-ordering:*

$$\mathrm{Str}_{T,E}[L](x,y) = \min(L(x,y), N_T(L(y,x)))$$

If T does not have zero divisors, the equality $\mathrm{Str}_{T,E}[L](x,y) = \min(L(x,y), N_T(E(y,x)))$ holds additionally.

As a first important property, we obtain that a given T-E-ordering L and the inverse of its induced strict T-E-ordering are disjoint.

Proposition 3. *With the assumptions of Proposition 2, the following equality holds for all $x,y \in X$:*

$$T(L(x,y), \mathrm{Str}_{T,E}[L](y,x)) = 0$$

The definition of $\mathrm{Str}_{T,E}[L]$ is obviously a straightforward translation of the construction $x \leq y \wedge y \not\leq x$ (being equivalent to $x \leq y \wedge x \neq y$ in case that T does not have zero divisors), but it need not be the only possibility to translate this construction to the fuzzy case (e.g. one could use the t-norm T instead of the minimum). Therefore, let us try to investigate whether $\mathrm{Str}_{T,E}[L]$ has some specific properties and, consequently, justifications. We could consider all strict T-E-orderings contained in a T-E-ordering L, but this is not a reasonable assumption. In the crisp case, we would at least assume the following obvious kind of montonicity:

$$(x \leq y \wedge y < z) \rightarrow x < z$$
$$(x < y \wedge y \leq z) \rightarrow x < z$$

These properties can be translated into the fuzzy setting in an obvious way.

Definition 5. A fuzzy relation R is called *monotonic* w.r.t. a given T-E-ordering L if and only if the following holds for all $x,y,z \in X$:

$$T(L(x,y), R(y,z)) \leq R(x,z)$$
$$T(R(x,y), L(y,z)) \leq R(x,z)$$

The next theorem shows that $\mathrm{Str}_{T,E}[L]$ is the greatest strict T-E-ordering contained in a given T-E-ordering L that fulfills monotonicity with respect to L.

Theorem 1. *Let E be a T-equivalence and let L be a T-E-ordering. Then $\mathrm{Str}_{T,E}[L]$ is the largest strict T-E-ordering that is monotonic w.r.t. L.*

As we are, of course, interested in the most specific information available, i.e. a minimal loss of information, we conclude that $\mathrm{Str}_{T,E}[L]$ is the most appropriate choice how to define a strict T-E-ordering from a given T-E-ordering L. Note that this loss of information can still be severe, as the following example demonstrates.

Example 3. Let us reconsider the $T_{\mathbf{L}}$-equivalence $E(x,y) = \max(1 - |x-y|, 0)$ and the $T_{\mathbf{L}}$-E-ordering $L(x,y) = \max(\min(1-x+y, 1), 0)$. Then we obtain

$$\mathrm{Str}_{T_{\mathbf{L}},E}[L](x,y) = \max(\min(y-x, 1), 0),$$

which is exactly R from Example 1. Now reconsider the $T_{\mathbf{P}}$-equivalence $E'(x,y) = \exp(-|x-y|)$ and the $T_{\mathbf{P}}$-E'-ordering $L'(x,y) = \min(\exp(y-x),1)$. Then we obtain $\mathrm{Str}_{T_{\mathbf{P}},E'}[L'](x,y) = 0$, i.e. there is no non-trivial strict $T_{\mathbf{P}}$-E'-ordering contained in L' that is monotonic w.r.t. L. Obviously, this is due to the fact that $L(x,y) > 0$ for all $x,y \in \mathbb{R}$ while we have

$$N_T(x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{otherwise.} \end{cases}$$

In such a case, therefore, we can never obtain a meaningful strict ordering.

Now let us try to clarify the other direction. The following proposition provides the necessary foundation.

Proposition 4. *Consider a T-equivalence E and a strict T-E-ordering R. Then the following fuzzy relation is a T-E-ordering:*

$$\mathrm{Ref}_{T,E}[R](x,y) = \max(R(x,y), E(x,y))$$

Again the question arises why exactly this choice is appropriate and how it is justified.

Proposition 5. *With the assumptions of Proposition 4, R is monotonic w.r.t.* $\mathrm{Ref}_{T,E}[R]$. *Moreover,* $\mathrm{Ref}_{T,E}[R]$ *is the smallest T-E-ordering extending R.*

Now we turn to the question under which conditions the correspondence is one-to-one.

Theorem 2. *Consider a T-equivalence E and a T-E-ordering L. Then the inequality*

$$\mathrm{Ref}_{T,E}[\mathrm{Str}_{T,E}[L]](x,y) \leq L(x,y)$$

holds. The equality

$$\mathrm{Ref}_{T,E}[\mathrm{Str}_{T,E}[L]](x,y) = L(x,y)$$

holds if and only if, for each pair $x,y \in X$, either $T(L(x,y),L(y,x)) = 0$ or $L(x,y) = E(x,y)$ holds.

Theorem 3. *Consider a T-equivalence E and a strict T-E-ordering R. Then the inequality*

$$R(x,y) \leq \mathrm{Str}_{T,E}[\mathrm{Ref}_{T,E}[R]](x,y)$$

holds. If T does not have zero divisors, we even have equality, i.e.

$$R(x,y) = \mathrm{Str}_{T,E}[\mathrm{Ref}_{T,E}[R]](x,y).$$

5 Linearity

Finally, let us approach the question whether linearity (completeness) is preserved by the transformations introduced in the previous section. The concepts of T-linearity and strong completeness as mentioned in Definition 3 are designed for T-E-orderings and are not meaningful for irreflexive relations. Hence, the next definition proposes a straightforward generalization of the well-known property of strict linearity

$$x \neq y \to (x < y \vee y < x). \qquad (1)$$

Definition 6. A fuzzy relation R is called *strictly T-E-linear* (with E being a T-equivalence) if the following inequality holds for all $x,y \in X$:

$$N_T(E(x,y)) \leq \max(R(x,y),R(y,x))$$

Based on this definition, it is possible to prove the following two theorems:

Theorem 4. *Assume we are given a T-equivalence E and a T-E-ordering L. If L is T-linear and fulfills* min-*E-antisymmetry*[2]*, then* $\mathrm{Str}_{T,E}[L]$ *is strictly T-E-linear.*

Note that strong completeness is a sufficient condition that T-linearity and min-E-antisymmetry are fulfilled simultaneously.

For the case of a t-norm inducing a strong negation we are able to prove the following stronger results.

Theorem 5. *Suppose we are given a T-equivalence E and a T-E-ordering L and furthermore assume that T induces a strong negation N_T. If L is T-linear, then the following two assertions holds for all $x,y \in X$:*

$$S_T(\mathrm{Str}_{T,E}[L](x,y),\mathrm{Str}_{T,E}[L](y,x)) \geq N_T(E(x,y))$$
$$S_T(\mathrm{Str}_{T,E}[L](x,y),E(x,y),\mathrm{Str}_{T,E}[L](y,x)) = 1$$

The first assertion in Theorem 5 can be understood as a slightly weakened strict T-E-linearity. The second assertion is an important result which is a straightforward generalization of the well-known fact that, in the crisp case, the following holds for any linear ordering $\leq$ (with $<$ being the corresponding strict ordering):

$$x < y \vee x = y \vee y < x$$

Note that this is, of course, an equivalent formulation of (1).

Finally, let us turn to the converse direction.

Theorem 6. *Assume we are given a T-equivalence E and a strict T-E-ordering R. Suppose further that T does not have zero divisors or that T induces a strong negation. If R is strictly T-E-linear, then* $\mathrm{Ref}_{T,E}[R]$ *is T-linear.*

6 Conclusion

We have introduced and justified a new concept of similarity-based strict fuzzy orderings. Meaningful correspondences between fuzzy orderings and strict fuzzy orderings have been established, but we have not obtained a general one-to-one correspondence. From this point of view, fuzzy orderings and strict fuzzy orderings are not fully equivalent concepts. Hence, the study of both concepts remains interesting and irredundant. Although t-norms without zero divisors give rise to some results that look nice at first glance (see Proposition 2, Theorem 3, and Theorem 6), the examples suggest that this is a rather restrictive and not very intuitive setting. On the other hand, the examples as well as results like Theorems 5 and 6 suggest that t-norms inducing strong negations (in particular, including nilpotent t-norms and the nilpotent minimum) have nice and intuitive properties in this context. This once more confirms the viewpoint that such t-norms are most adequate choices in fuzzy relations theory, fuzzy preference modeling and related fields [4, 6, 8, 22].

[2] i.e. L is a fuzzy ordering in the sense of Bělohlávek [1].

Acknowledgements

Ulrich Bodenhofer gratefully acknowledges support by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the K*plus* Competence Center Program. Support by COST Action 274 "TARSKI" is also gratefully acknowledged.

References

[1] R. Bělohlávek. *Fuzzy Relational Systems. Foundations and Principles*. IFSR Int. Series on Systems Science and Engineering. Kluwer Academic/Plenum Publishers, New York, 2002.

[2] U. Bodenhofer. A similarity-based generalization of fuzzy orderings preserving the classical axioms. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 8(5):593–610, 2000.

[3] U. Bodenhofer. Representations and constructions of similarity-based fuzzy orderings. *Fuzzy Sets and Systems*, 137(1):113–136, 2003.

[4] U. Bodenhofer and F. Klawonn. A formal study of linearity axioms for fuzzy orderings. *Fuzzy Sets and Systems*, 145(3):323–354, 2004.

[5] D. Boixader, J. Jacas, and J. Recasens. Fuzzy equivalence relations: Advanced material. In D. Dubois and H. Prade, editors, *Fundamentals of Fuzzy Sets*, volume 7 of *The Handbooks of Fuzzy Sets*, pages 261–290. Kluwer Academic Publishers, Boston, 2000.

[6] B. De Baets and J. Fodor. Towards ordinal preference modelling: the case of nilpotent minimum. In *Proc. 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume I, pages 310–317, Paris, 1998.

[7] B. De Baets and R. Mesiar. Pseudo-metrics and T-equivalences. *J. Fuzzy Math.*, 5(2):471–481, 1997.

[8] B. De Baets, B. Van de Walle, and E. E. Kerre. A plea for the use of Łukasiewicz triplets in the definition of fuzzy preference structures. (II). The identity case. *Fuzzy Sets and Systems*, 99(3):303–310, 1998.

[9] M. Demirci. On many-valued partitions and many-valued equivalence relations. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 11(2):235–253, 2003.

[10] M. Demirci. A theory of vague lattices based on many-valued equivalence relations—I: general representation results. *Fuzzy Sets and Systems*, 151(3):437–472, 2005.

[11] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, 1994.

[12] P. Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 1998.

[13] U. Höhle and N. Blanchard. Partial ordering in L-underdeterminate sets. *Inform. Sci.*, 35:133–144, 1985.

[14] F. Klawonn. Fuzzy sets and vague environments. *Fuzzy Sets and Systems*, 66:207–221, 1994.

[15] F. Klawonn and J. L. Castro. Similarity in fuzzy reasoning. *Mathware Soft Comput.*, 3(2):197–228, 1995.

[16] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000.

[17] S. V. Ovchinnikov. Similarity relations, fuzzy partitions, and fuzzy orderings. *Fuzzy Sets and Systems*, 40(1):107–126, 1991.

[18] S. V. Ovchinnikov and M. Roubens. On strict preference relations. *Fuzzy Sets and Systems*, 43:319–326, 1991.

[19] A. Pultr. Fuzziness and fuzzy equality. *Comment. Math. Univ. Carolin.*, 23(2):249–267, 1982.

[20] J. G. Rosenstein. *Linear Orderings*, volume 98 of *Pure and Applied Mathematics*. Academic Press, New York, 1982.

[21] E. Trillas and L. Valverde. An inquiry into indistinguishability operators. In H. J. Skala, S. Termini, and E. Trillas, editors, *Aspects of Vagueness*, pages 231–256. Reidel, Dordrecht, 1984.

[22] B. Van de Walle, B. De Baets, and E. E. Kerre. A plea for the use of Łukasiewicz triplets in the definition of fuzzy preference structures. (I). General argumentation. *Fuzzy Sets and Systems*, 97(3):349–359, 1998.

[23] L. A. Zadeh. Similarity relations and fuzzy orderings. *Inform. Sci.*, 3:177–200, 1971.

Technical Report
SCCH–TR–0508

On the Relationship of Kernels in Machine Learning and Fuzzy Similarity Relations

Bernhard Moser
Software Competence Center Hagenberg
A-4232 Hagenberg, Austria
bernhard.moser@scch.at

Abstract — In this paper, we present a view of kernels from a fuzzy set theoretical perspective. Indeed, it turns out that kernels which are positive definite functions have to fulfill a consistency property given by the so-called T-transitivity of a fuzzy T-equivalence relation with respect to the triangular norm T. As a result, we introduce a triangular norm T_{Cos} which is characterized as being the greatest one for which all kernels are T_{Cos}-equivalences. Finally, a way of constructing kernels by means of fuzzy sets is outlined.

Key words — *Machine learning, t-norms, fuzzy set, fuzzy equivalence relation, positive definiteness, Hilbert space*

Kompetenzzentren-Programm

1 Introduction

Kernels are two-placed symmetric real functions that can be reproduced as inner products of points of an Hilbert space. It is a classical result of linear algebra that a symmetric positive definite matrix $A = (a_{ij}) \in \mathbb{R}^n \times \mathbb{R}^n$ has this property as it can be decomposed by virtue of its eigenvectors ϕ_1, ..., ϕ_n and its positive eigenvalues λ_1, ..., λ_n,

$$A = (\phi_1, \dots, \phi_n)^T \Gamma (\phi_1, \dots, \phi_n) \tag{1}$$

where Γ denotes the diagonal matrix consisting of the eigenvalues. Since equation (1) can be rewritten as

$$a_{ij} = \langle \phi_i, \phi_j \rangle$$

with the inner product $\langle ., . \rangle$ defined by

$$\langle (x_1, \dots, x_n), (y_1, \dots, y_n) \rangle = \sum_{k=1}^{n} \lambda_k x_k y_k$$

symmetric positive definite matrices turn out to be kernels, i.e., in terms of linear algebra, symmetric positive definite matrices are Gram matrices composed of inner products.

As an inner product is a geometric notion, Gram matrices and therefore kernel functions as their generalization on more general index sets (continuum instead of discrete finite set of indices) often emerge in the context of optimization prodecures motivated by geometric ideas.

Recently, learning methods based on kernels like support vector machines, kernel principal component analysis, kernel Gram-Schmidt or Bayes point machines have received considerable advertency (see for example [2, 3, 19, 31]). What makes kernel methods so attractive can be explained by two aspects: firstly, by virtue of the so-called kernel trick data are mapped implicitely into a higher dimensional feature space in a way that preserves the geometrical notion of the inital optimization procedure based on linear models while extending it to non-linear models; secondly, the representer theorem guarantees that the non-linear optimum can be represented as a superposition of kernel functions which allows to design tractable optimization algorithms (see for example [2, 10, 33, 39, 40]).

These methods find successful applications to classification, regression, density estimation and clustering problems in computer vision, data mining and machine learning.

While the historical roots of kernel methods can be traced back to the mid of the last century [1, 26], the study of positive definite functions as kernels of integrals date back to the beginning of the 19th century [24]. It was Mercer who in [24], 1909, characterized kernels in terms of a positive defniteness condition as a generalization of the classical result from linear algebra (1).

A positive inner product of normed vectors x and y can also be looked at as a similarity measure S for the vectors under consideration. The smaller the angle α between the vectors the higher the degree of similarity due to the basic formula

$$S(x, y) = \cos(\alpha) = \frac{\langle x, y \rangle}{\|x\| \, \|y\|} \tag{2}$$

While this heuristcally introduced notion of similarity neglects the characteristic law of transitivity of similarity, fuzzy equivvalence relations provide an axiomatic framework for similarity taking also transitivity into account. If x and y are similar, and y and z are similar then we expect a certain degree of similarity between x and z, otherwise the similarity assertions would run into inconsistencies. For (2) by geometric reasoning it becomes evident that

$$S(x, z) \geq \cos(\alpha + \beta)$$

where $S(x, y) = \cos(\alpha)$ and $S(y, z) = \cos(\beta)$, as the resulting angle between x and z is bounded by the sum of α and β provided that they do not differ too much, that means the sum of both angles keeps within 90 degrees otherwise the similarity vanishes. It is interesing that this basic example

$$T_{Cos}(\cos(\alpha), \cos(\beta)) = \max\{\cos(\alpha + \beta), 0\} \tag{3}$$

turns out to fulfill the axioms for a triangular norm which in fuzzy logic plays the role of a fuzzy and operator. It is actually equation (3) that motivates the nomenclature T_{Cos}. It is the main result of this paper to show that all kernels $k : \mathcal{X} \times \mathcal{X} \to [0, 1]$ have to satisfy the T_{Cos}-transitivity

$$T_{Cos}(k(x, y), k(y, z)) \leq k(x, z) \tag{4}$$

for all elements x, y and z. For a servey on triangular norms see, e.g., [20].

This result can also be expressed in terms of inner products which leads to the following inequality

$$\langle p, q\rangle\langle q, r\rangle - \sqrt{1 - (\langle p, q\rangle)^2}\sqrt{1 - (\langle q, r\rangle)^2} \leq \langle p, r\rangle,$$

or, equivalently, but more compactly transformed into the form of a triangle inequality,

$$\arccos(\langle p, r\rangle) \leq \arccos(\langle p, q\rangle) + \arccos(\langle q, r\rangle)$$

which holds true for any choice of normed elements p, q and r ($\|p\| = 1$, $\|q\| = 1$, $\|r\| = 1$, where $\|.\| = \langle ., .\rangle$) of an arbitrary Hilbert space $(H, \langle ., .\rangle)$. Note that in the Eucledean geometry (1) and (4) are related by the well known trigonometric formula for sums of angles

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$$

By this, the different concepts discussed above, namely kernels (Gram matrices) and positive definiteness on the one hand and fuzzy equality relations and t-nomrs on the other hand, are shown to be closely related.

This is a novel view which might be interesting for both concepts. As there is a well developped theory on kernels, above all characterizations due to Bochner [4], Aronszajn [1] and Yaglom [42], there opens up new construction methods for equality relations. Vice versa new kernels can be constucted by virtue of fuzzy set theoretically based concepts.

To start with, in the following sections the main concepts concerning kernels and fuzzy equivalence relations are outlined. As the main result it is demonstrated that all kernels which map into the unit interval have to be T_{Cos}-equivalences. Finally, by means of the minimum t-norm a sufficient criterion is offered and a new way to construct kernels by engaging fuzzy sets is outlined.

1.1 Kernels

The term *kernel* originates in integral operator theory which dates back to the beginning of the last century (see, e.g., [30]). In this context kernels are two-placed functions which define a linear integral operator, e.g., for a Fredholm equation of the second kind

$$\phi(s) - \int_a^b k(s,t)\phi(t) = f(s)$$

where $a \leq s \leq b$, k is continuous on the square $[a,b] \times [a,b]$ and f is continuous on $[a,b]$. Equation (1.1) reduces to a system of n linear algebraic equations in m unknowns if the kernel k has the special form

$$k(s,t) = \sum_{j=1}^{m} \tau_j(s)\rho_j(t) \tag{5}$$

This is the reason why kernels of the form (5) and, in particular,

$$k(s,t) = \sum_{j=1}^{m} \tau_j(s)\tau_j(t) \tag{6}$$

play such an important role in the framework of integral equations. Functions of the form (6) are closely related to positive definiteness.

Definition 1. Let $\mathcal{X}$ be a non-empty set. A real-valued function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a positive definite kernel (short kernel) iff it is symmetric, that is, $k(x,y) = k(y,x)$ for all $x, y \in \mathcal{X}$, and positive definite, that is, $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$ for any $n \in \mathbb{N}$ and choice of $x_1, \ldots, x_n \in \mathcal{X}$ and any choice of real numbers $c_1, \ldots, c_n \in \mathbb{R}$.

Remark 2. In contrary to linear algebra this definition of positive definiteness is common in the approximation and machine learning literature (compare [2, 41]).

Obviously functions of the form (6) are symmetric, they are also positive definite since

$$\begin{aligned} \sum_{i,j=1}^{n} c_i c_j k(s_i, s_j) &= \sum_{l=1}^{m} \sum_{i,j=1}^{n} c_i c_j \tau_l(s_i)\tau_l(s_j) \\ &= \sum_{l=1}^{m} \left(\sum_{i=1}^{n} c_i \tau_l(s_i) \right)^2 \\ &\geq 0 \end{aligned}$$

and, therefore, they are kernels in the sense of definition (1).

Actually, the property (6) and its formulation in terms of inner products is characteristic for kernels according to a classical result from functional analysis due to Aronszajn [1].

Theorem 3. *For any kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that*

$$k(x,y) = \langle \Phi(x), \Phi(y) \rangle, \tag{7}$$

for any $x, y \in \mathcal{X}$, where $\langle ., . \rangle$ denotes the inner product in the Hilbert space.

Because of its relevance for kernel methods the property (7) in literature is sometimes chosen to be the starting point for the definition of a kernel (compare, e.g., [10]).

Theorem (3) does not tell how to construct the Hilbert space $\mathcal{H}$ (feature space) and the mapping Φ. Actually, $\mathcal{H}$ is not even uniquely determined.

One way to obtain $\mathcal{H}$ is to start with $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}} := \{f : \mathcal{X} \to \mathbb{R}\}$, a set of real-valued functions on $\mathcal{X}$, and apply the Riesz representation theorem (cf, e.g., [32]), by which a bounded linear functional $\mathcal{F} : \mathcal{H} \to \mathbb{R}$, that is there is an upper bound $M > 0$ such that

$$\forall f \in \mathcal{H} : \mathcal{F}[f] \leq M\|f\|_{\mathcal{H}},$$

is uniquely represented by

$$\mathcal{F}[f] = \langle a, f\rangle_{\mathcal{H}} \tag{8}$$

for an element $a \in \mathcal{H}$ and where $\langle ., .\rangle_{\mathcal{H}}$ and $\|.\|_{\mathcal{H}}$ denotes the inner product of $\mathcal{H}$ and the norm induced by it, respectively.

If all evaluation functionals $\delta_x : \mathcal{H} \to \mathbb{R}$, $x \in \mathcal{X}$, given by

$$\delta_x[f] = f(x)$$

are postulated to be bounded, then due to Riesz representation theorem (8), to each element $x \in \mathcal{X}$ there is an element $K_x \in \mathcal{H}$ such that

$$f(x) = \delta_x[f] = \langle K_x, f\rangle_{\mathcal{H}}.$$

Particularly, for $f = K_y$ we obtain

$$K(x, y) := K_x(y) = \langle K_x, K_y\rangle_{\mathcal{H}} \tag{9}$$

This shows that each Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$, for which all evaluation functionals are bounded, induce a positive kernel K with feature map $\Phi(x) = K(x, .)$. Such Hilbert sapces are called *reproducing kernel Hilbert sapce* (*RKHS* for short) due to equation (9). Vice versa it can be shown that a positive kernel K induces uniquely a RKHS which is generated by K (see, e.g., [1, 2]).

While the feature space $RKHS$ is a function space, Mercer's theorem demonstrates the construction of a feature space made up of sequences, that is ℓ_2, the set of square summable sequences (see [24]).

Theorem 4. *Suppose $k \in L_\infty(\mathcal{X}^2)$ is a symmetric real-valued function such that for all $f \in L_2(\mathcal{X})$ and any finite measure μ on $\mathcal{X}$, we have*

$$\int_{\mathcal{X}^2} k(x, y)f(x)f(y)d\mu(x)d\mu(y) \geq 0. \tag{10}$$

Let $\Psi_j \in L_2(\mathcal{X})$ be the normalized orthogonal eigenfunctions of the integral operator $T_{k,\mu}$ given by

$$T_{k,\mu} : L_2(\mathcal{X}) \to L_2(\mathcal{X})(T_{k,\mu})(x) := \int_{\mathcal{X}} k(x, y)f(y)d\mu(y) \tag{11}$$

associated with the eigenvalues $\lambda_j > 0$, sorted in non-decreasing order. Then

- $(\lambda_j)_j \in \ell_2$,
- $k(x, y) = \sum_{j=1}^{N} \lambda_j \Psi_j(x)\Psi_j(y)$ *holds for allmost all* $(x, y) \in \mathcal{X}^2$. $N \in \mathbb{N}$, *or* $N = \infty$*; in the latter case, the series converges absolutely and uniformly for allmost all* $(x, y) \in \mathcal{X}^2$.

1.2 Properties and Classes of Kernels

In this section some basic properties of kernels are summarized which can be found in [8] or [2].

Proposition 5. ***(Cauchy Schwartz Inequality)*** *Any kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfies*

$$|k(x,y)|^2 \leq k(x,x)k(y,y) \tag{12}$$

for any choice of $x, y \in \mathcal{X}$

Proof. From the positive definiteness assumption it follows that the 2×2 matrix

$$\begin{pmatrix} k(x,x) & k(x,y) \\ k(y,x) & k(y,y) \end{pmatrix}$$

is a positive semi-deifinite matrix and implies a non-negative determinant. Therefore, $k(x,x)k(y,y) - k(x,y)k(x,y) \geq 0$ □

Kernels also have pleasant algebraic properties, so they form a cone and, even, the product of two kernels is again a kernel.

Proposition 6. ***(Cone of Kernels)*** *If k_1 and k_2 are kernels on the same domain, then*

$$k(x,y) = \lambda_1 k_1(x,y) + \lambda_2 k_2(x,y) \tag{13}$$

is again a kernel, where $\lambda_i \geq 0$, $i \in \{1,2\}$.

Proof. The proof follows immediately from the definition (1) □

The next proposition is also an immediate consequence of the definition (1).

Proposition 7. ***(Renaming Arguments)*** *Let $\sigma : \mathcal{X} \to \mathcal{X}$ be a bijecttion and let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel iff $\tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, given by $\tilde{k}(x,y) = k(\sigma(x), \sigma(y))$ is a kernel.*

Proposition 8. ***(Product of Kernels)*** *If k_1 and k_2 are kernels on the same domain, then*

$$k(x,y) = k_1(x,y)k_2(x,y) \tag{14}$$

is again a kernel

Proof. We have to show that for any $n \in \mathbb{N}$ and any choice of reals $c_1, \ldots, c_n$ there holds

$$\sum_{i,j=1}^{n} c_i c_j k_1(x_i,x_j) k_2(x_i,x_j) \geq 0. \tag{15}$$

Consider the matrices $K_1 = (k_1(x_i,x_j))_{i,j}$ and $K_2 = (k_2(x_i,x_j))_{i,j}$. Due to the basic characterization from linear algebra of positive semi definite matrices in terms of eigenvalues and eigenvectors it follws that there are matrices $S_1 = (s^1_{i,j})_{i,j}$, $S_2 = (s^2_{i,j})_{i,j}$ composed of eigenvectors of K_1 and K_2 and diagonal matrices Γ_1, Γ_2 made up by their non negative eigenvalues $\lambda^1_1, \ldots, \lambda^1_n$ and $\lambda^2_1, \ldots, \lambda^2_n$ of K_1 and K_2, respectively, such that

$$K_1 = S_1^T \Gamma_1 S_1, K_2 = S_2^T \Gamma_2 S_2.$$

This leads to

$$\begin{aligned}\sum_{i,j=1}^{n} c_i c_j k_1(x_i,x_j) k_2(x_i,x_j) &= \sum_{i,j} c_i c_j \sum_k s^1_{i,k}\lambda^1_k s^1_{k,j} \sum_l s^2_{i,l}\lambda^2_l s^2_{l,j} \\ &= \sum_{k,l} \lambda^1_k \lambda^2_l \left(\sum_i s^1_{i,k} s^2_{i,l} \right)^2 \\ &\geq 0.\end{aligned}$$

□

Generalizing (6) and (8) a theorem due to [8] yields

Theorem 9. ***(Closeness Properties of Kernels)*** *Let $f : \mathbb{R}^n \to \mathbb{R}$, $n \in \mathbb{N}$ then $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by*

$$k(x,y) := f(k_1(x,y), \ldots, k_n(x,y))$$

is a kernel for any choice of kernels $k_1, \ldots, k_n$ on $\mathcal{X} \times \mathcal{X}$ iff

$$f(x_1, \ldots, x_n) = \sum_{k_1 \geq 0, \ldots k_n \geq 0} c_{k_1,\ldots,k_n} x_1^{k_1} \cdots x_n^{k_n}$$

where $c_{k_1,\ldots,k_n} \geq 0$ for all nonnegative indeces $k_1, \ldots, k_n$.

Translation invariant kernels, i.e., kernels k with $k(x,y) = k(x-y)$ for all $x, y \in \mathcal{X}$, can be characterized by their spectral representation due to Bochner, [4].

Theorem 10. ***(Bochner's characterization)*** *Let $k : \mathbb{R}^n \to \mathbb{R}$. Then k is a translation invariant kernel iff it can be represented by*

$$k(x-y) = \int_{\mathbb{R}^n} \cos(\omega^T(x-y))\mu(d\omega) \tag{16}$$

where μ is a positive finite measure.

Note that Bochner's representation (16) is the Fourier transform of a real function. For

$$\mu(A) = (\frac{\sigma}{2})^{(d/2)} \int_A \exp\left(-\omega^T \omega \ \left(\frac{\sigma}{2}\right)^2 \right) d\omega$$

equation (16) yields the widely used *Gaussian* kernel

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{\sigma^2} \right)$$

For further details and further classes see [2, 9, 12].

1.3 Triangular norms and T-equalities

Triangular norms have been originally studied within the framework of probabilistic metric spaces (cf. [34, 35]. In this context t-norms proved to be an appropriate concept when dealing with triangle inequalities. Later on, t-norms and their dual version t-conorms have been used to model conjunction and disjunction for many-valued logic (cf. ([11, 13, 14, 20]).

Definition 11. A function $T : [0,1]^2 \to [0,1]$ is called *t-norm* (triangular norm), if it satisfies the following conditions:

$$\begin{array}{llll} (i) & \forall x,y \in [0,1]: & T(x,y) = T(y,x) & \text{(commutativity)} \\ (ii) & \forall x,y,z \in [0,1]: & T(x,T(y,z)) = T(T(x,y),z) & \text{(associativity)} \\ (iii) & \forall x,y,z \in [0,1]: & y \leq z \Longrightarrow T(x,y) \leq T(x,z) & \text{(monotonicity)} \\ (iv) & \forall x,y \in [0,1]: & T(x,1) = x \wedge T(1,y) = y & \text{(boundary condition)} \end{array}$$

A t-norm is called Archemedian if it is continuous and satisfies

$$x \in (0,1) \implies T(x,x) < x.$$

Archmedian t-norms are characterized by the following representation theorem due to [23]

Theorem 12. *Let $T : [0,1] \times [0,1] \to [0,1]$ be a t-norm. Then, T is Archemedian iff there is a continuous, strictly decreasing function $f : [0,1] \to [0,\infty]$ with $f(1) = 0$ such that for $x,y \in [0,1]$*

$$T(x,y) = f^{-1}(\min(f(x)+f(y), f(0)))$$

By setting $g(x) = \exp(-f(x))$ Ling's characterization yields an alternative representation with a multiplicative generator function

$$T(x,y) = g^{-1}(\max(g(x)\,g(y), g(0)))$$

For $g(x) = x$ we get the product, $T_P(x,y) = x\,y$. $f(x) = 1-x$ yields the so-called Łukasiewcz t-norm $T_L(x,y) = \min(x+y-1, 0)$. Archemedian t-norms are either isomorphic to T_P or T_L. In the former case the Archemedian t-norm is called *strict* in the latter *non-strict*.

1.3.1 Φ-Operators

t-norms are also employed to construct extensions of the Boolean implication by means of the concept of a Φ-operator, which was introduced by Pedrycz [27], see also [13].

Given a t-norm T the function $\Phi : [0,1]^2 \to [0,1]$ is called a Φ-operator with respect to T if it satisfies

[Φ1] Φ is monotone increasing in the secong component
[Φ2] $T(a,\Phi(a,b)) \leq b$
[Φ3] $b \leq \Phi(a,T(a,b))$

While [Φ2] can be interpreted as a many-valued version of the law of *modus ponens* from Aristotelean logics, [Φ3] can be motivated by the classical tautology $b \to (a \to a \wedge b)$, which provides an interchanging rule for a and b and [Φ1] by the tautology $(b \to c) \to ((a \to b) \to (a \to c))$. It can be shown that a continuous t-norm T uniquely determines a Φ-operator Φ_T by

$$\Phi_T(a,b) = \sup\{c \in [0,1] | T(a,c) \leq b\} \tag{17}$$

It is interesting that among continuous t-norms non-strict Archemedian t-norms characterize continuous Φ-operators (see [25]).

Theorem 13. *For a continuous t-norm T the Φ-operator Φ_T is continuous iff T is non-strict Archemedian.*

If f is an additive generator of the non-strict Archemedian t-norm, that is $f(0) < \infty$, then

$$\Phi_T(a,b) = f^{-1}(\max(f(b) - f(a), 0)). \tag{18}$$

Table (1) lists examples of t-norms with their induced Φ-operators. For further examples see, e.g., [20].

definition of t-norm	induced Φ-*operator*
$T_D(a,b) = \begin{cases} \min(a,b) & \text{if } \max(a,b) = 1, \\ 0 & \text{else} \end{cases}$	$\Phi_D(a,b) = \begin{cases} b & \text{if } a = 1, b < 1, \\ 1 & \text{else} \end{cases}$
$T_P(a,b) = a\,b$	$\Phi_P(a,b) = \begin{cases} \frac{b}{a} & \text{if } a > b, \\ 1 & \text{else} \end{cases}$
$T_L(a,b) = \min(a+b-1, 1)$	$\Phi_L(a,b) = \min(b-a+1, 1)$
$T_M(a,b) = \min(a,b)$	$\Phi_M(a,b) = \begin{cases} b & \text{if } a > b, \\ 1 & \text{else} \end{cases}$

Table 1: Examples of t-norms and induced Φ-operators

Utilizing the concept of a t-norm T-equalities extend the two-valued concept of an equivalence relation to a many-valued version.

1.3.2 T-equivalences

If we want to classify based on a notion of similarity or indistinguishability we face the problem of transitivity. For instance, let us consider two real numbers to be indistinguishable if and only if they differ at most a certain bound $\varepsilon > 0$, that is modeled by the relation $\sim_\varepsilon$ given by $x \sim_\varepsilon y$ $:\Leftrightarrow |x - y| < \varepsilon$, $\varepsilon > 0$, $x, y \in \mathbb{R}$. Note that the relation $\sim_\varepsilon$ is not transitive and, therefore, not an equivalence relation. The transitivity requirement turns out to be to strong for this example. The problem of identification and transitivity in the context of similarity of physical objects was early pointed out and discussed philosophically by Poincaré ([28], [29]). In the framework of fuzzy logic the way to overcome this problem is to model similarity by fuzzy relations based on a many-valued concept of transitivity (see also [5], [6], [15], [16], [20], [43]).

Definition 14. A function $E : X^2 \longrightarrow [0,1]$ is called an *indistinguishability relation*, or synonymously, T-*equivalence* with respect to the t-norm T if it satisfies the following conditions:

$$\begin{array}{llll} (i) & \forall x \in X: & E(x,x) = 1 & \text{(reflexivity)} \\ (ii) & \forall x,y \in X: & E(x,y) = E(y,x) & \text{(symmetry)} \\ (iii) & \forall x,y,z \in X: & T(E(x,y),E(y,z)) \leq E(x,z) & \text{(T-transitivity)} \end{array}$$

The value $E(x, y)$ can be also looked at as the (quasi) truth value of the statement "*x is equal to y*". Following this semantics the *T-transitivity* can be seen as a many-valued model of the proposition "*If x is equal to y and y is equal to z, then x is equal to z*". T-equivalences for Archemedian t-norms are closely related to metrics and pseudo metrics as shown by

Theorem 15. *Let T be an Archimedian t-norm given by*

$$\forall a, b \in [0,1] : T(a,b) = f^{-1}(min(f(a) + f(b), f(0))),$$

where $f : [0,1] \to [0,\infty]$ is a strictly decreasing, continous function with $f(1) = 0$.
(i) If $d : X^2 \to [0,\infty[$ is a pseudo metric, then the function $E_d : X^2 \to [0,1]$ defined by

$$E_d(x,y) = f^{-1}(min(d(x,y), f(0)))$$

is an equaltiy relation with respect to T.
(ii) If $E : X^2 \to [0,1]$ is an equality relation with respect to T, then the function $d_E : X^2 \to [0,\infty]$ defined by

$$d_E(x,y) = f(E(x,y))$$

is a pseudo metric.

Proof. For a proof of this theorem, see [20, 25]. □

Another way to construct T-equivalences is to employ Φ-operators.

Theorem 16. *Let T be a continuous t-norm, Φ_T its induced Φ-operator, $\mu_i : \mathcal{X} \to [0,1]$, $i \in I$, I non-empty, then $E : \mathcal{X} \times \mathcal{X} \to [0,1]$ given by*

$$E(x,y) = \inf_{i \in I} \left(\min\left(\Phi_T(\mu_i(x), \mu_i(y)), \Phi_T(\mu_i(y), \mu_i(x))\right)\right) \tag{19}$$

is a T-equivalence relation.

The proof can be found in [21, 22, 37]. For further details on indistinguishability realtions see also [7, 17, 18, 36, 38].

2 Kernels are T-equivalencies

Let us start with the analysis of 3-dimensional matrices.

Lemma 17. *Let $M = (m_{ij})_{ij} \in [0,1]^{3\times 3}$ be a 3×3 symmetric matrix with $m_{ii} = 1$, $i = 1,2,3$, then M is positive semi-definite iff for all $i, j, k \in \{1,2,3\}$ there holds*

$$m_{ij}m_{jk} - \sqrt{1 - m_{ij}^2}\sqrt{1 - m_{jk}^2} \le m_{ik} \tag{20}$$

Proof. For simplicity, let $a = m_{1,2}$, $b = m_{1,3}$ and $c = m_{2,3}$. Then the determinant of M, $Det(M)$, is a function of the varibles a, b, c given by

$$D(a,b,c) = 1 + 2abc - a^2 - b^2 - c^2. \tag{21}$$

For any choice of a, b the quadratic equation $D(a, b, c) = 0$ can be solved for c yielding two solutions $c_1 = c_1(a, b)$ and $c_2 = c_2(a, b)$ as functions of a and b,

$$\begin{aligned} c_1(a,b) &= ab - \sqrt{1-a^2}\sqrt{1-b^2} \\ c_2(a,b) &= ab + \sqrt{1-a^2}\sqrt{1-b^2}. \end{aligned}$$

Obviously, for all $|a| \leq 1$ and $|b| \leq 1$ the values $c_1(a, b)$ and $c_2(a, b)$ are real. By substituting $a = \cos\alpha$ and $b = \cos(\beta)$ with $\alpha, \beta \in [0, \frac{\pi}{2}]$ it becomes readily clear that

$$\begin{aligned} c_1(a,b) &= c_1(\cos(\alpha), \cos(\beta)) \\ &= \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ &= \cos(\alpha+\beta) \in [-1, 1] \end{aligned}$$

and, analogously,

$$\begin{aligned} c_2(a,b) &= c_2(\cos(\alpha), \cos(\beta)) \\ &= \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta) \\ &= \cos(\alpha-\beta) \in [-1, 1] \end{aligned}$$

As for all $a, b \in [-1, 1]$ the determinant function $D_{a,b}(c) := D(a, b, c)$ is quadratic in c with negative coefficient for c^2 there is a uniquely determined maximum at $c_0(a, b) = ab$. Note that for all $a, b \in [-1, 1]$ we have

$$c_1(a,b) \leq c_0(a,b) \leq c_2(a,b) \tag{22}$$

and

$$D(a,b,c_0(a,b)) = 1 + 2ab(ab) - a^2 - b^2 - (ab)^2 = (1-a^2)(1-b^2) \geq 0. \tag{23}$$

Therefore, $D(a, b, c) \geq 0$ if and only if $c \in [c_1(a, b), c_2(a, b)]$. Recall that by renaming the indeces, the determinant does not change (a simple elementary interchanging operation $\sigma(k) = l$ and $\sigma(l) = k$, $k \neq l$, can be performed by a matrix multiplication $M[\sigma(k,l)] = T^T_{\sigma(k,l)} M T_{\sigma_{k,l}}$ where $T = (t_{i,j})_{i,j}$ is identical to the identity $t_{i,j} = 1$ for $i = j$ and $t_{i,j} = 0$ else except for the indeces k, l where $t_{k,k} = t(l, l) = 0$ and $t(k, l) = t(l, k) = 1$. As $Det(T_{\sigma(k,l)}) \in \{-1, 1\}$ it follows that $Det(\tilde{M}) = Det(T^T_{\sigma(k,l)})Det(M)Det(T_{\sigma_{k,l}}) = Det(M)$. An arbitrary permutation σ can be splitted in a sequence of elementary interchaning operations proving the assertion.) Therefore, without loss of generality we may assume that

$$a \geq b \geq c. \tag{24}$$

For convenience, let $Q = \{(x, y, z) \in [0, 1]^3 | x \geq y \geq z\}$. Then, obviously, for any choice of $a, b \in [0, 1]$ there holds $(a, b, c_1(a, b)) \in Q$. Elementary algebra shows that $(a, b, c_2(a, b)) \in Q$ is only the case for $a = b = 1$. As for $a = b = 1$ the two solutions c_1, c_2 coincide, $c_1(1, 1) = c_2(1, 1) = 1$ it follows that for any choice of $(a, b, c) \in Q$ there holds $Det(M) \geq 0$ if and only if

$$c_1(a,b) = ab - \sqrt{1-a^2}\sqrt{1-b^2} \leq c. \tag{25}$$

If $(a, b, c) \notin Q$ then the inequality (25) is trivially satisfied which together with (25) proves the lemma □

Proposition 18. *$T(a,b) = \max(ab - \sqrt{1-a^2}\sqrt{1-b^2}, 0)$ is a non-strict Archemedian t-norm with additive generator $f(x) = \arccos(x)$.*

Proof. Let $a = \cos(\alpha)$ and $b = \cos(\beta)$ with $\alpha, \beta \in [0, \pi/2]$, then

$$\begin{aligned} f^{-1}(\min(f(a)+f(b), f(0))) &= \\ \cos(\min(\arccos(\cos(\alpha)) + \arccos(\cos(\beta)), \arccos(0))) &= \\ \cos(\min(\alpha+\beta, \pi/2)) &= \\ \max(\cos(\alpha+\beta), 0) &= \\ \max(\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)) &= \\ \max(ab - \sqrt{1-a^2}\sqrt{1-b^2}) & \end{aligned}$$

which proves that $arccos$ is the additive generator □

The graph of T_{Cos} is depicted in figure (1).

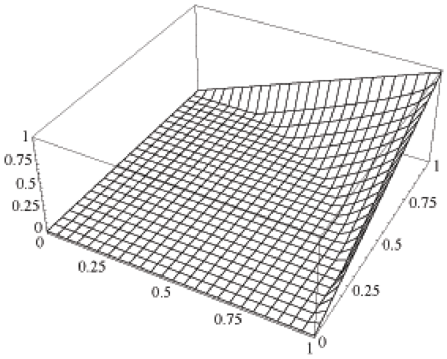

Figure 1: Graph of T_{Cos}

Corollary 19. *Let T_{Cos} be the t-norm defined in (18) and let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel, then for any choice of elements $x, y, z \in \mathcal{X}$ there holds*

$$T_{Cos}(k(x,y), k(y,z)) \leq k(x,z). \tag{26}$$

Moreover, T_{Cos} is the greatest t-norm with this property.

Proof. This, immediately follows from the definition (1) and the lemma (17). □

Remark. For dimensions ≥ 4 the inequalities (26) are no longer sufficient tp guarantee positive definiteness. Consider, for example,

$$A = \begin{pmatrix} 1 & 0 & 1/2 & 3/5 \\ 0 & 1 & 3/5 & 1/2 \\ 1/2 & 3/5 & 1 & 0 \\ 3/5 & 1/2 & 0 & 1 \end{pmatrix} \tag{27}$$

Though for all indeces $i, j, k \in 1, \ldots, 4$ the coefficients of $A = (a_{ij})_{ij}$ of example (27) satisfy the conditions (26), the matrix A is not positive semi-definite, as it has a negative eigenvalue.

As a kernel can be represented by an inner product of some Hilbert space, corollary (20) yields an interesting inequality for inner products. Note, that for dimension 2 in (12) by analysing the determinant we got the well-known and fundamental Cauchy-Schartz inequality. Now, the analysis of the determinant of 3-dimensional matrices leads to the inequalities (28).

Corollary 20. *Let T the t-norm defined in (18), let $\mathcal{H}$ be a Hilbert space endowed with the inner product $\langle ., . \rangle$, then for any choice of elements $x, y, z \in \mathcal{H}$ with $\|x\| \leq 1$, $\|y\| \leq 1$, $\|z\| \leq 1$ there holds*

$$\langle x, y\rangle\langle y, z\rangle - \sqrt{1 - \langle x, y\rangle^2}\sqrt{1 - \langle y, z\rangle^2} \leq \langle x, z\rangle, \tag{28}$$

or, equivalently,

$$\arccos(\langle x, z\rangle) \leq \arccos(\langle x, y\rangle) + \arccos(\langle y, z\rangle) \tag{29}$$

Proof. The proof directly follows from (22), (22), (22) and (23) and the trigonometric identity $\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) = \cos(\alpha + \beta)$. □

Inequality (29) can be looked at as a triangle inequality for inner products.

2.1 Constructing kernels by T_M

The following lemma and proposition can also be found as an exercise in [2].

Lemma 21. *Let $\sim$ be an equivalence relation on $\mathcal{X}$ and let $k : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ induced by $\sim$ via $k(x, y) = 1$ if and only if $x \sim y$. Then k is a kernel.*

Proof. By definition of positive definiteness, let us consider an arbritrary sequence of elements $x_1, \ldots, x_n$. Then there are at most n equivalence classes $Q_1, \ldots, Q_m$ on the set of indeces $\{1, \ldots, n\}$, $m \leq n$, where $\bigcup_{i=1,\ldots,m} Q_i = \{1, \ldots, n\}$ and $Q_i \cap Q_j = \emptyset$ for $i \neq j$. Note that $k(x_i, x_j) = 0$ if the indeces i, j belong to different equivalence classes. Then, for any choice of reals $c_1, \ldots, c_n$, we obtain

$$\begin{aligned}
\sum_{i,j} c_i c_j k(x_i, x_j) &= \sum_{p=1}^{m} \sum_{i,j \in Q_p} c_i c_j k(x_i, x_j) \\
&= \sum_{p=1}^{m} \sum_{i,j \in Q_p} c_i c_j \cdot 1 \\
&= \sum_{p=1}^{m} \left(\sum_{i \in Q_p} c_i \right)^2 \\
&\geq 0
\end{aligned}$$

□

Proposition 22. *$k : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ is a kernel iff it is induced by an equivalence relation.*

Proof. It only remains to be shown that if k is a kernel then it is the indicator function of an equivalence relation, that is, it is induced by an equivalence relation. If k is a kernel, according to (26) for all $x, y, z \in \mathcal{X}$ it has to satisfy $T_{Cos}(k(x,y), k(y,z)) \leq k(x,z)$ which implies,

$$k(x,y) = 1, \quad k(y,z) = 1 \Longrightarrow k(x,z) = 1. \tag{30}$$

Obviously, $k(x,x) = 1$ and $k(x,y) = k(y,x)$ due to the reflexitivity and symmetry of an equivalence relation, respectively. □

Lemma (21) can be extended to the $[0,1]$ intervall by employing the T_M $t$norm.

Corollary 23. *T_M-equivalences are kernels.*

Proof. Let $k : \mathcal{X} \times \mathcal{X} \to [0,1]$ with $k(x,x) = 1$, $k(x,y) = k(y,x)$ and $\min(k(x,y), k(y,z)) \leq k(x,z)$ for any $x, y, z \in \mathcal{X}$. We have to show that for any choice of finite subset $Q = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ it follows that for all $c_1, \ldots, c_n$ there holds

$$\sum_{i,j : x_i, x_j \in Q} c_i c_j k(x_i, x_j) \geq 0.$$

Consider the finte set $\{k(x_i, x_j) \in [0,1] | x_i, x_j \in Q\} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ with $m \leq n$. Without loss of generality we may assume $\alpha_p < \alpha_q$ for $p < q$. Further we set $\alpha_0 = 0$. Then, for any choice elements $x, y \in Q$ we obtain the representation

$$k(x,y) = \sum_{p=1}^{m} (\alpha_p - \alpha_{p-1}) 1_{\{(a,b) | k(a,b) \geq \alpha_{p-1}\}}(x,y)$$

Because, if $k(x,y) = \alpha_q$, $x, y \in Q$, then

$$\begin{aligned} k(x,y) &= \sum_{p=1}^{m} (\alpha_p - \alpha_{p-1}) 1_{\{(a,b) | k(a,b) \geq \alpha_{p-1}\}}(x,y) \\ &= \sum_{p=1}^{q} (\alpha_p - \alpha_{p-1}) \\ &= \alpha_q. \end{aligned}$$

Observe that $1_{\{(a,b) | k(a,b) \geq \alpha\}}(.,.)$ is the indicator function of an equivalence relation because of

$$\alpha \leq \min(k(x,y), k(y,z)) \leq k(x,z)$$

which demonstrates the transitivity. By this, k turns out to be the superposition of kernels with positive coefficients. From this, together with lemma (21) and the cone property of kernels, see (6), we conclude that k has to be a kernel. □

By means of (19) T_M-equivalences can be constructed starting from an arbitrary set of fuzzy sets. Let $\mu_i : \mathcal{X} -> [0,1]$, $i \in I$, be a family of fuzzy sets, then

$$E(x,y) = \inf_{i \in I} \left(\min \left(\Phi_M(\max(\mu_i(x), \mu_i(y)), \min(\Phi_M(\mu_i(y), \mu_i(x))) \right) \right)$$

generates a T_M-equivalence and, therefore, an kernel. If μ_i are indicator functions with $\mu_i(x) = \mu_j(x) = 1$ only if $i = j$, then the resulting T_M-equivalence is actually the indicator function of an equivalence relation induced by the fuzzy sets μ_i. For applications in machine learning by means of the fuzzy sets μ_i a priori knowledge or knowledge obtained by other methods, as for instance statistically derived informations about the samples, can be incorporated to construct the kernel. The kernel constructed in this way somehow represents the c̈lusterïnformation of the μs in a compact manner.

2.2 Conclusion

This paper is intended to be a starting point for further research exploring the interrelations between kernel based learning methods and the theory of T-equivalences. So, the close interrelationship between the concepts of kernels from machine learning and T-equivalences from fuzzy set theory was pointed out. This was mainly substantiated by two results. Firstly, kernels $k : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ are T_{Cos}-equivalences, where T_{Cos} is the non-strict Archemedian t-norm with additive generator $f(x) = \arccos(x)$. Secondly, a sufficient criterion is provided by the result that T_M-equivalences are kernels, where T_M denotes the minimum t-norm. Further, it was outlined how fuzzy sets can be incorporated to construct kernels.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[2] A. J. Smola B. Schölkopf. *Learning with Kernels*. MIT Press, Cambridge, 2002.

[3] A. J. Smola B. Schölkopf and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[4] S. Bochner. *Harmonic Analysis and the Theory of Probability*. University of California Press, Los Angeles, California, 1955.

[5] U. Bodenhofer. A note on approximate equality versus the Poincaré paradox. *Fuzzy Sets and Systems*. (to appear).

[6] U. Bodenhofer. A note on approximate equality versus the Poincaré paradox. *Fuzzy Sets and Systems*, 133(2):155–160, 2003.

[7] D. Boixader and J. Jacas. T-indistinguishability operators and approximate reasoning via CRI. In D. Dubois, E. P. Klement, and H. Prade, editors, *Fuzzy Sets, Logics and Reasoning about Knowledge*, volume 15 of *Applied Logic Series*, pages 255–268. Kluwer Academic Publishers, Dordrecht, 1999.

[8] A. Pinkus C. H. FitzGerald, C.A. Micchelli. Functions that preserve families of positive semidefinite matrices. *Linear Alg. and Appl.*, 221:83–102, 1995.

[9] N. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, New York, 1993.

[10] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[11] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Inform. Sci.*, 36:85–121, 1985.

[12] M. G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

[13] S. Gottwald. Fuzzy set theory with t-norms and Φ-operators. In A. Di Nola and A. G. S. Ventre, editors, *The Mathematics of Fuzzy Systems*, volume 88 of *Interdisciplinary Systems Research*, pages 143–195. Verlag TÜV Rheinland, Köln, 1986.

[14] S. Gottwald. *Fuzzy Sets and Fuzzy Logic*. Vieweg, Braunschweig, 1993.

[15] U. Höhle. Fuzzy equalities and indistinguishability. In *Proc. 1st European Congress on Fuzzy and Intelligent Technologies*, volume 1, pages 358–363, Aachen, 1993.

[16] U. Höhle. The Poincaré paradox and non-classical logics. In D. Dubois, E. P. Klement, and H. Prade, editors, *Fuzzy Sets, Logics and Reasoning about Knowledge*, volume 15 of *Applied Logic Series*, pages 7–16. Kluwer Academic Publishers, Dordrecht, 1999.

[17] F. Höppner, F. Klawonn, and P. Eklund. Learning indistinguishability from data. *Soft Computing*, 6(1):6–13, 2002.

[18] J. Jacas. On the generators of T-indistinguishability operators. *Stochastica*, 12:49–63, 1988.

[19] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 1986.

[20] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000.

[21] R. Kruse, J. Gebhardt, and F. Klawonn. *Fuzzy-Systeme*. B. G. Teubner, Stuttgart, 1993.

[22] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. John Wiley & Sons, New York, 1994.

[23] C. H. Ling. Representation of associative functions. *Publ. Math. Debrecen*, 12:189–212, 1965.

[24] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 209:415–446, 1909.

[25] B. Moser. *A New Approach for Representing Control Surfaces by Fuzzy Rule Bases*. PhD thesis, Johannes Kepler Universität Linz, October 1995.

[26] E. Parzen. Extraction and detection problems and reproducing kernel hilbert spaces. *Journal of the Society for Industrial and Applied Mathematics. Series A, On control*, 1:35–62, 1962.

[27] W. Pedrycz. Fuzzy control and fuzzy systems. Technical Report 82 14, Dept. of Math., Delft Univ. of Technology, 1982.

[28] H. Poincaré. *La Science et l'Hypothése*. Flammarion, Paris, 1902.

[29] H. Poincaré. *La Valeur de la Science*. Flammarion, Paris, 1904.

[30] A. D. Polyanin and A. V. Manzhirov. *Handbook of Integral Equations*. CRC Press, Boca Raton, 1998.

[31] T. Graepel R. Herbrich and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.

[32] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, 1966.

[33] B. Schölkopf. *Support Vector Learning*. Oldenbourg Verlag, Munich, 1997.

[34] B. Schweizer and A. Sklar. Associative functions and statistical triangle inequalities. *Publ. Math. Debrecen*, 8:169–186, 1961.

[35] B. Schweizer and A. Sklar. *Probabilistic Metric Spaces*. North-Holland, Amsterdam, 1983.

[36] E. Trillas, S. Cubillo, and E. Castiñeira. Menger and Ovchinnikov on indistinguishabilities revisited. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 7(3):213–218, 1999.

[37] E. Trillas and L. Valverde. An inquiry into indistinguishability operators. In H. J. Skala, S. Termini, and E. Trillas, editors, *Aspects of Vagueness*, pages 231–256. Reidel, Dordrecht, 1984.

[38] L. Valverde. On the structure of F-indistinguishability operators. *Fuzzy Sets and Systems*, 17(3):313–328, 1985.

[39] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[40] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[41] G. Wahba. Soft and hard classification by reproducing kernel hilbert space methods. In *Proc. of the National Academy of Sciences of the United States of America*, volume 99(26), pages 16524–16530, 2002.

[42] A. M. Yaglom. Some classes of random fields in n-dimensional space, related to stationary random processes. *Theory of Probability and its Applications*, 2:273–320, 1957.

[43] L. A. Zadeh. Similarity relations and fuzzy orderings. *Inform. Sci.*, 3:177–200, 1971.

A Nonlinear Approximation Formula Generator for Very High Dimensional Data Based on Variable Selection and Genetic Programming

Werner Groißböck

Department of Knowledge-Based Mathematical Systems
Fuzzy Logic Laboratorium Linz-Hagenberg
Johannes Kepler University Linz, A-4040 Linz, Austria
werner.groissboeck@jku.at

1 Introduction

A new approach for finding nonlinear approximation formulas for very high-dimensional data is presented. The method is based on linear regression, but instead of the original variables we use nonlinear terms with these variables. Such a formula is still linear in the parameters, so least squares can be applied to find the globally optimal parameters. We use an accelerated version of genetic programming to find the optimal nonlinear terms, and we use variable selection methods to select those terms leading to an approximation formula which shows an optimal balance of accuracy and simplicity. In general, evolutionary methods like genetic programming tend to produce many individuals with low fitness. To save computation time, an early stopping strategy in case of low fitness is used.

2 The new algorithm

2.1 The core of the new algorithm

In the following, the original independent is called y. At the beginning, the actual independent is the original independent $y_{actual} = y$. Later y_{actual} will be modified. The constant term $c = (1, \ \dots \ , 1)^T$ is always the first variable that is chosen. But this variable is not counted as real variable. The algorithm performs the following steps:

1. An accelerated version of genetic programming (including a population of individuals and a crossover operator) is used to generate millions of very

simple formulas. We select that formula x_A which is best correlated with the actual independent y_{actual}. We look only at the absolute value of the correlation coefficient.

2. Then we modify y_{actual} such that all the parts of y that can be approximated with the regressors already chosen are subtracted, setting y_{actual} to $y - \hat{y}(c, x_A)$. Here $\hat{y}(c, x_A)$ is the linear best approximation of y with the use of the regressors c and x_A. We can say, y_{actual} is y made orthogonal to the regressors already chosen.
3. Once again the accelerated version of genetic programming is used to generate millions of very simple formulas. And now we select that formula x_B which is correlated strongest with the actual independent y_{actual}. We look only at absolute values again.
4. Then once again, y_{actual} is made orthogonal to the regressors already chosen, so we set y_{actual} to $y - \hat{y}(c, x_A, x_B)$.
5. Continue in this manner, until a given number of regressor terms is selected or some other termination criterion is fulfilled.

2.2 The accelerated version of genetic programming - an overview

Interrupting the calculation of the fitness as early as possible, when it can be seen that the checked individual is not worth spending additional time, accelerates the algorithm. So we need an algorithm which calculates the fitness of only the better individuals exactly and estimates the fitness of all the other individuals.

In the following lines the major steps of the accelerated genetic programming algorithm are described.

1. Generate an initial population with n_{large} individuals.
2. Evaluate each individual for $n1$ points of the training data set and estimate the correlation coefficient with the actual dependent by using only these $n1$ points.
3. Determine the n_{small} best correlated individuals out of n_{large}, based on the estimated correlation coefficient. We look only at the absolute value of the correlation coefficient.
4. For these n_{small} chosen individuals the $e$xact value of the fitness function (i.e. the absolute value of the correlation coefficient) using $a$ll the points of the training data set has to be calculated.
5. Produce a new generation of n_{large} out of the n_{small} chosen individuals:
   - Repeat the following, until we have enough new individuals. Choose randomly two of the n_{small} individuals and compare their fitness. The better one is called the winner, and the other one is called the loser. Let the winner produce two offsprings, one is an exact copy of the winner, and the other offspring is made via crossover (as crossover partner, one of the n_{small} individuals is chosen, which is neither the winner nor the loser).

- The individual which is the best so far is always copied into the next generation ('elitism').
- A small part of the new generation is produced in the same way as the initial population. This is one way of avoiding the problem with local optima. A mutation is not needed any more.

6. Go to step 2, until a termination criterion is fulfilled.

The runpar-list: The definition of a few important parameters

Now a parameter list is introduced, which can be used to control the global behavior of the algorithm.

AlgoVariant: This is the central parameter.
If $AlgoVariant$ is 1, then no acceleration is used and no genetic programming. The terms are generated randomly, and the term with the best fitness is chosen. The algorithm stops if $GPTimeMax$ seconds are elapsed. If $AlgoVariant$ is 2, then the accelerated version of AlgoVariant 1 is used. So terms are generated randomly, and the fitness is estimated by using only $n1$ points and if the individuum is near the best so far, then the fitness is calculated exactly. The accelerated variant performs better than the not accelerated variant. Only if a data set is very short, for example 50 or less data points, then the not accelerated variant may be useful.
If $AlgoVariant$ is 3, then the new variant which is described here is used, which uses the accelerated version of genetic programming (see 2.2) including a population and a crossover operator.

GPTimeMax: This parameter is only used if $AlgoVariant$ is 1 or 2. In these variants, we try to find the best correlated term, and we stop this procedure, when GPTimeMax seconds are elapsed. Then the next term is searched, and again until GPTimeMax seconds are elapsed. And so on. The approximation formula that we finally want is a combination of these terms.

n1: The parameter $n1$ tells the algorithm, how many points are used to get a quick estimation of the correlation coefficient. For $AlgoVariant = 2$ Monte Carlo experiments have been performed with $n1 = 20$, $n1 = 30$ and $n1 = 50$ and we have seen, that $n1 = 50$ leads to the best results.

popsize: Determines, how many individuals are in one generation of the genetic programming algorithm. Only used, if $AlgoVariant$ is 3. In section 2.2 the expression n_{large} can be found. This parameter n_{large} is exactly corresponding to the parameter *popsize* here. One of our standard settings is a *popsize* of 5000. The larger *popsize*, the more computation time is needed.

popsizeDivisor: Only used, if $AlgoVariant$ is 3. In section 2.2 the expression n_{small} can be found. The parameter $popsizeDivisor$ is used to define the parameter n_{small}.

$$n_{small} = popsize/popsizeDivisor$$

It is assumed, that $popsizeDivisor$ is a divisor of $popsize$.
Example: If $popsize$ is 5000 and $popsizeDivisor$ is 10, then $n_{small} = 500$. This is one of our standard settings. A $popsizeDivisor$ of 5 has also been used quite often.

nGenerations: In section 2.2 the phrase 'until a termination criterion is fulfilled' can be found. As termination criterion we simply use that the actual number of generations reaches the parameter $nGenerations$. Quite often $nGenerations = 24$ or $nGenerations = 18$ is used. If a more exact formula shall be reached, then the parameter $nGenerations$ can be increased, if possible while $popsize$ is also increased. To make further improvements possible, when $nGenerations$ and $popsize$ are already quite high, the parameter $nRuns$ has been introduced.

nRuns: The whole algorithm, as described in section 2.2 is repeated $nRuns$ times and the best overall individual is stored. So if you want to spend a lot of computation time for finding one specific nonlinear term, then you can increase $popsize$, $nGenerations$ and $nRuns$. If $popsize$ and $nGenerations$ are already quite high, for example $popsize = 32000$ and $nGenerations = 100$, then increasing $popsize$ would lead to storage problems. If $nGenerations$ is very high, then the optimization process is likely to converge to a local optimum. But the parameter $nRuns$ can be increased without any limits, as long as enough computation power is available.

NewIndPercentage: In our genetic programming algorithm, when the individuals for the next generation are constructed, then most of them are made via copying and crossovering of the parent individuals. But a certain percentage of the next generation is made with the same method that is used to generate the initial population. This percentage is given by the parameter $NewIndPercentage$. An example: If $popsize$ is 5000 and $NewIndPercentage$ is 0.25, then 1250 individuals of the 5000 are generated with the same method that has been used to generate the initial population. The reason for doing this is that we want to maintain genetic diversity.
This parameter can also be used to make experiments with $NewIndPercentage = 1$. Then no crossover at all is performed, and

all the individuals are only generated like the initial population. If this would lead to good results, then the crossover operator would be useless. Our experiments have shown that the crossover operator is not useless.

numofelems: The parameter $numofelems$ is needed to generate a new individuum in the initial population. $numofelems$ is used as an *upper bound* for the number of elements in the corresponding genetic programming tree. If $numofelems$ is 4, then formulas up to degree 2 will appear. If $numofelems$ is 6, then formulas up to degree 3 will appear. If $numofelems$ is 8, then formulas up to degree 4 will appear.

NewIndElemsVaryingFlag: If this flag is one, then the individuals in the initial population have widely varying sizes. If this flag is zero, then the size of all the individuals in the initial population is roughly constant. An example: If $numofelems$ is 16 and $NewIndElemsVaryingFlag$ is 0, then approximately 70% of the generated individuals have a size of 15, and approximately 30% have a size of 14. This is the case, because the individual generator repeatedly makes a small tree larger by replacing a terminal symbol with a function and the corresponding terminals. If this is done for a tree with 14 elements, and we try to replace a terminal by the function + and two random arguments, then we would get a tree with 16 elements, and this is too large, by the definition of $numofelems$. So in this case the individual generation algorithm stops and takes that individual that has a size of 14.

If $numofelems$ is 16, and $NewIndElemsVaryingFlag$ is 1, then $numofelems_{actual}$ is set to a random integer number in the interval $[1, 16]$, and $numofelems_{actual}$ is used as an upper bound for the size of the tree instead of $numofelems$. So the size of the individuals that we get is varying. If $numofelems_{actual}$ is for example 9, then we expect to get an individual of size 8, but with a probability of approximately 0.3 we get an individual with size 7.

Crossnumofelems: If a crossover operator is used, then the size of the offspring individual can be up to twice the size of the parent individuals. For this reason a delimiter is needed. We use the parameter $Crossnumofelems$ to do so. If the size of an individual is too large, then the crossover is repeated. If the individual is still to large, then the crossover operator is used again to get a new individual. If the new individual is again too large, then we stop the crossover process and instead we copy one of the parent individuals. To calculate the size of an individual we use the number of elements in the corresponding genetic programming tree. So this is done exactly in the same way as for the parameter *numofelems*.

2.3 Systematic Experiments to Find Good Parameter Settings

Totally eight data sets have been used to perform our experiments, we call them 'D1', ... ,'D8'.

- D1: 8748 rows and 14 columns; AVL, simulated engine test bench data;
- D2: 475 rows and 149 columns; DaimlerChrysler, measured engine test bench data.
- D3: 161 rows and 70 columns; GUASCOR, measured engine test bench data
- D4: 506 rows and 14 columns; UCI-repository: the HOUSING data set
- D5: 1000 rows and 8 columns, simulated, no noise.
- D6: 1000 rows and 8 columns, simulated, with noise.
- D7: 1000 rows and 50 columns, simulated, no noise.
- D8: 1000 rows and 50 columns, simulated, with noise.

Each of the data sets has been split into two parts. Usually the first eighty percent are used as training data and the other twenty percent are used as test data[1]. An approximation formula is determined with using only the training data, and then the quality of this formula is calculated, as well on the training data as on the test data. As a quality measure for the training data we take R^2 as usual. As a quality measure for the test data we take the square of the correlation coefficient of the measured data and the approximated data. In our experiments, several nested loops have to be run (so they need a lot of computation time usually about ten hours):

1. First, eight files are used, and the experiments have to be performed for every file.
2. Then, for each file an approximation for the first eight variables is made.
3. Furthermore to get significant results each experiment has usually been repeated ten times.
4. In our algorithm at first the most important term is determined and a regression formula is made with this term, and then the next term is added to the term collection, and once again a regression formula is made, now for two terms. This process is repeated until $DimMax$ terms are selected. So the GP-based term finder algorithm has been called $DimMax$ times. And for each of the resulting formulas, the quality measure is determined, as well for the training data as for the test data. The more terms are used, the higher the quality measure, at least for the training data. For very complicated approximation formulas, over-fitting problems may arise. So simpler formulas can have better performance on the test data set.

[1] For the housing data set, only fifty percent are used as training data and the other fifty percent are used as test data.

The result of one large experiment with all the loops mentioned above is a huge file. Typically it contains $8 \cdot 8 \cdot 10 = 640$ rows, because we have eight files, eight variables and ten repetitions. In the following, such an experiment producing these 640 rows is called *Standard Experiment*. And in each row of such an experiment we find the following information:

1. a) The approximation formula based on *one* nonlinear term
   b) The quality of this approximation formula, evaluated on the training data set
   c) The quality of this approximation formula, evaluated on the test data set
2. a) The approximation formula based on *two* nonlinear terms
   b) The quality of this approximation formula, evaluated on the training data set
   c) The quality of this approximation formula, evaluated on the test data set
3. And so on, until we have an approximation formula based on $DimMax$ terms

Usually the value of $DimMax$ is five. Making $DimMax$ larger does not only increase the needed computation time, but also allows more complicated formulas and thus the risk of overfitting.

To compare two different parameter settings, the following has to be done:

1. At first two large experiments have to be started with the two parameter settings of interest.
2. Then the two result files have to be compared, especially the quality measures.

To make two result files comparable, usually for each result file the *average row* is determined, as far as the qualities are concerned. If $DimMax$ is for example five, then we have ten averaged qualities for each file:

1. a) The quality of the *one*-term-approximation-formula on the *training* data.
   b) The quality of the *one*-term-approximation-formula on the *test* data.
2. a) The quality of the *two*-term-approximation-formula on the *training* data.
   b) The quality of the *two*-term-approximation-formula on the *test* data.
3. a) The quality of the *three*-term-approximation-formula on the *training* data.
   b) The quality of the *three*-term-approximation-formula on the *test* data.
4. a) The quality of the *four*-term-approximation-formula on the *training* data.
   b) The quality of the *four*-term-approximation-formula on the *test* data.

5. a) The quality of the *five*-term-approximation-formula on the *training* data.
   b) The quality of the *five*-term-approximation-formula on the *test* data.

In this section, these qualities are put into one *table*, a first example can be seen in table 1.

Number of terms	**1**	**2**	**3**	**4**	**5**
Quality on training data	0.5558	0.7176	0.7899	0.8011	0.8122
Quality on test data	0.5333	0.6809	0.7501	0.7612	0.7631

Table 1. This is an example quality table. Such tables are used to compare the results of two experiments

So if the qualities in the quality table of experiment A are better than the qualities of experiment B, then the parameters of experiment A are supposed to be better than the parameters of experiment B. In the result files, usually the average row has been determined for the first experiment and for the second experiment.

References

1. Hastie, T., Tibshirani, R., and Friedman J. , "The Elements of Statistical Learning: Data Mining, Inference, and Prediction.", Springer Berlin, 2001
2. F. E. Harrell jr., "Regression modeling strategies: With applications to linear models, logistic regression and survival analysis", Springer Series in Statistics, 2001
3. J. R. Koza, "Genetic Programming", The MIT Press, Cambridge, Massachusetts, 1992
4. J. R. Koza, "Genetic Programming II", The MIT Press, Cambridge, Massachusetts, 1994
5. W. M. Lee, C. P. Lim, K. K. Yuen, S. M. Lo, "A Hybrid Neural Network Model for Noisy Data Regression", IEEE Transactions on Systems, Man and Cybernetics, Part B 34:2, Pages 951-960, 2004
6. L. Ljung, "System Identification: Theory for the User", ISBN 0-13-656695-2, Prentice Hall PTR, New Jersey, 1999
7. Merz, C. J., Pazzani, M. J., "Combining Neural Network Regression Estimates with Regularized Linear Weights", Advances in Neural Information Processing Systems 9, edited by M.C. Mozer, M.I. Jordan, and T. Petsche, 1997
8. A. Miller, "Subset Selection in Regression - Second Edition", ISBN 1-58488-171-2 Chapman & Hall/CRC Boca Raton London New York Washington, D.C. 2002
9. O. Nelles, "Nonlinear System Identification - From Classical Approaches to Neural Networks and Fuzzy Models", ISBN 3-540-67369-5 Springer-Verlag Berlin Heidelberg New York, 2001

10. W. H. Press, S. A. Teukolsky, W. T. Vetterling and P.B. Flannery, "Numerical Recipes in C: The Art of Scientific Computing", Cambridge University Press, Cambridge, U.K., second ed., 1992
11. J. R. Quinlan, "Combining Instance-Based and Model-Based Learning", Proceedings on the Tenth International Conference of Machine Learning, Pages 236-243, University of Massachusetts, Morgan Kaufmann, 1993

Technical Report
SCCH–TR–0517

A Bacterial Evolutionary Algorithm for Feature Selection

Mario Drobics
Software Competence Center Hagenberg
e-mail *mario.drobics@scch.at*

János Botzheim
Guest at the Fuzzy Logic Laboratory Linz-Hagenberg
e-mail *botzheim@alpha.tmit.bme.hu*

Abstract — When creating regression models from data the problem arises that the complexity of the models rapidly increases with the number of features involved. Especially in real world application where a large number of potential features are available, feature selection becomes a crucial task. A novel approach for feature selection is presented which uses a bacterial evolutionary algorithm to identify the optimal set and the optimal number of features with respect to a given learning problem *and* a given learning algorithm. This method ensures high accuracy and significantly increases interpretability of the resulting models.

Key words — *feature selection, bacterial evolutionary algorithm*

Kplus
Kompetenzzentren-Programm

1 Introduction

To create regression models from data several methods like statistical regression, neural networks, or regression tree methods [BFSO84] exist. The problem, however, arises that the complexity of these models rapidly increases with the number of features involved. This causes two major problems: On the one hand, some features may have a very low bias but a high variance and mislead the regression methods. On the other hand a large number of predictors decreases interpretability, as the major influences are likely to be shadowed by other unimportant features. Furthermore, taking measurements is often a time consuming and costly task. Reducing the number of measurements (features) used is therefore an important design goal.

Dimension reduction or feature selection can be used to reduce the dimensionality of the original state space (i.e. to reduce the number of features under investigation). While dimension reduction methods like Principal Component Analysis (PCA) [Bis95] use projection methods which often cumber interpretation of the resulting models, feature selection methods aim to identify the most relevant features out of the original set of features [GE03, LKM01].

In this paper we will present a novel approach to feature selection using bacterial evolutionary algorithm [BDK04]. In machine learning the goal is to find a function which models a relation between the input and the output space. An increase in the dimensionality of the input space increases the complexity of this learning problem. When features with only minor or no relation at all to the output space are involved, the resulting function f might tend to overfit the training data. Although various methods exist to overcome these problems, it is often more efficient to reduce the number of features beforehand.

Feature selection can be described as the task of identifying an optimal subset of m out of the available n features. The resulting subset of m features is then used to compute a function $\bar{f}$, which maps from the m-dimensional input space $\bar{X}$ to the output space Y. We would like to identify not only the optimal subset of a given size (m) but to find the optimal size of this subset, too.

2 Bacterial evolutionary algorithm

There are several optimization algorithms which were inspired by the evolutionary processes of biological organisms. One of the recent evolutionary approaches is referred to as bacterial evolutionary algorithm. This method iteratively combines two operations inspired by the microbial evolution phenomenon. The bacterial mutation operation optimizes the features of a single bacterium, while the gene transfer operation provides the transfer of information between the bacteria in the population. These processes can be easily applied in optimization problems where one individual corresponds to one solution of the problem.

2.1 The encoding method

In the algorithm, one bacterium $\xi_i, i \in I$ corresponds to one solution of a given problem. First, we have to define how a solution is encoded in such a bacterium (chromosome). For the task of selecting m features from a set of n features ($m \leq n$), the bacterium consists of a vector of integers $\xi_i = \{\xi_i^1, \ldots, \xi_i^m\}, 1 \leq \xi_i^k \leq n$, where $\xi_i^k \neq \xi_i^l$ for $k \neq l$. Because we want to find the optimal m value too, thus this value is not predefined for the bacteria.

2.2 The evaluation function

Similar to genetic algorithms the fitness of a bacterium ξ_i is evaluated using an *evaluation function* $\phi(\xi_i)$. The choice of this evaluation function is problem dependent. For the task of feature selection we use the features encoded in bacterium ξ_i and the training data set $\mathcal{S}$ to compute a regression model f_i according to:

$$f_i(\mathbf{x}) : X_{\xi_i^1} \times \ldots \times X_{\xi_i^m} \mapsto Y.$$

The evaluation function is then computed as the average squared error of the input-recall behavior of this model on test data set $\mathcal{T} \subset X \times Y$ supplemented by the length of the bacterium:

$$\phi(\xi_i) = \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{x},y) \in \mathcal{T}} \big(f_i(\mathbf{x}) - y\big)^2 + \beta \frac{l(\xi_i)}{MAXLEN},$$

where $l(\xi_i)$ means the length of the bacterium ξ_i, *MAXLEN* is a predefined value for the maximal allowed bacterium length and β is a trade-off parameter between accuracy and complexity.

2.3 The evolutionary process

The basic algorithm consists of three steps [BHK$^+$02, NF99]. First, an initial population has to be created randomly. Then, bacterial mutation and gene transfer are applied, until a stopping criteria is fulfilled. The evolution cycle is summarized below:

Bacterial Evolutionary Algorithm

```
create initial population
do {
    apply bacterial mutation
    apply gene transfer
} while stopping condition not fulfilled
return best bacterium
```

2.4 Generating the initial population

First an initial bacterium population of N_{ind} bacteria $\{\xi_i, i \in I\}$ is created randomly $(I = \{1, \ldots, N_{\text{ind}}\})$. Figure 1 shows a bacterium ξ_i with $n = 50$. The length of the bacterium is initialized also randomly between 1 and *MAXLEN*. On figure 1 the length of the bacterium is 5.

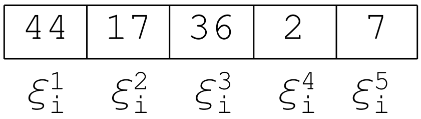

Figure 1: A single bacterium

2.5 Bacterial mutation

Bacterial mutation is applied to all bacteria $\xi_i, i \in I$. First, N_{clones} copies (clones) of the bacterium are created.

$$\xi_{i,j} = \xi_i, \qquad \forall j : 1 \leq j \leq N_{\text{clones}}$$

Then, in each clone $\xi_{i,j}$ a random segment of the chromosome is replaced by random numbers not greater than n ($\xi_{i,j}^k = \text{Random}[n]$). When we change a segment of a bacterium, we must take care that the new segment is unique within the selected bacterium $\xi_i^k \neq \xi_i^l$ for $k \neq l$. Next, all the clones and the original bacterium are evaluated using the evaluation function $\phi(\xi)$. The bacterium with the best evaluation result is used to transfer the mutated segment to the other individuals. This cycle is repeated for the remaining segments, until all segments of the chromosome have been mutated and tested. At the end, the best bacterium is kept and the remaining N_{clones} are discharged. The length of the segment is also a parameter of the bacterial mutation (*MutationLength*). When we change a segment in some clone, then this segment can be shorter or longer (or remain the same length) than before. Thus, new numbers can be added or some numbers can be removed from the bacterium. We also have a parameter on this (*ModifiedMutationLength*). Figure 2 shows an example mutation for $N_{\text{clones}} = 3$, $MutationLength = 1$, $ModifiedMutationLength = 0$.

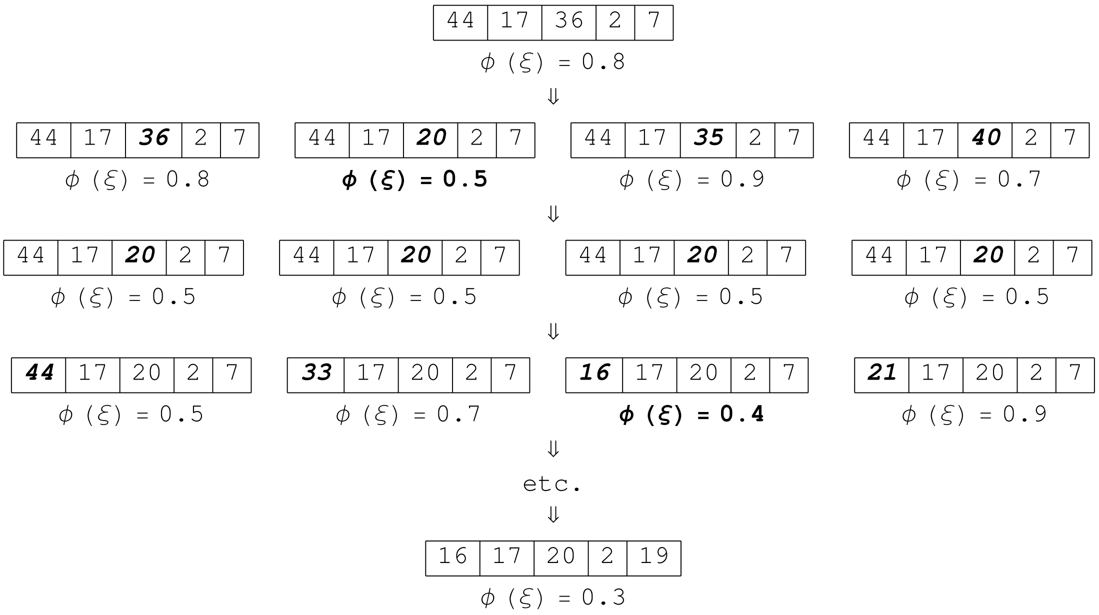

Figure 2: Bacterial mutation

2.6 Gene transfer

The bacterial mutation operator optimizes the bacteria in the population. Often, however, this is not enough as we need to provide a possibility for some information flow within the population. Using the gene transfer operator, the recombination of genetic information between two bacteria is possible.

1. First, the population must be sorted and divided into two halves according to their evaluation results. The bacteria with a better score are called superior half, the bacteria with a worse score inferior half.
2. Then one bacterium is randomly chosen from the superior half and another from the inferior half. These two bacteria are called the source bacterium, and the destination bacterium, respectively.
3. A segment from the source bacterium is randomly chosen and this segment is used to overwrite a random segment of the destination bacterium if the source segment is not already in the destination bacterium, or the source segment can be added to the destination bacterium without any overwriting.

Gene transfer is repeated N_{inf} times, where N_{inf} is the number of “infections” per generation. As in the bacterial mutation, here we have also two other parameters, the length of the source segment (*GeneTransferLength*), and the length of the change in the destination bacterium (*ModifiedGeneTransferLength*). Figure 3 shows an example for the gene transfer operations ($N_{\text{ind}} = 4, N_{\text{inf}} = 3$, $GeneTransferLength = 1$, $ModifiedGeneTransferLength = 0$).

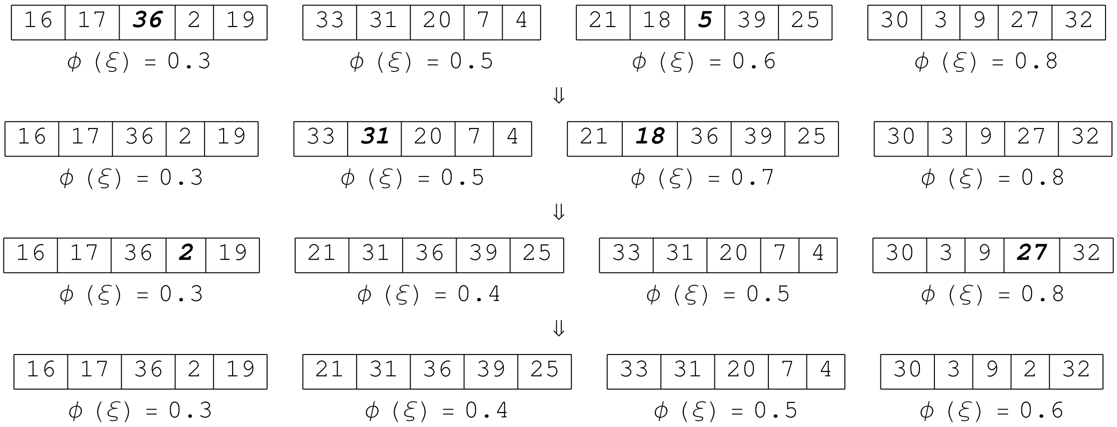

Figure 3: Gene transfer

2.7 Stopping condition

If a minimum error value is reached by the best bacterium in the population or the maximum number of generations N_{gen} is reached then the algorithm ends, otherwise it returns to the bacterial mutation step.

3 Simulation results

The algorithm used for many simulations was realized in Mathematica. We applied a high-dimensional problem to test the power of the algorithm. The test function was defined over a 20-dimensional data space $[0,1]^{20}$ according to:

$$f_{20}(\mathbf{x}) = x_1 x_2^2 x_{13}^3 - x_{20} + 5\sin(x_{16}) - 25\cos(x_5 x_{18}) + \exp(x_3 x_5) + x_4 x_{19} + x_{10}^2 + x_{11}^5.$$

We created 500 training samples by assigning each input dimension a random number with equal distribution. The function has random behavior in the remaining dimensions generated by a random generator. To find an optimal approximation function we used linear, as well as exponential, quadratic and cubic transformations of the input dimensions. Then we created a prediction model by computing a least-squares fit to the data as a linear combination of the input features. This gives us a set of 80 possible input features.

The learning curves for a sample simulation are shown in Figure 4. The parameter setting of this simulation is the following:

N_{gen} = 40	*MutationLength* = 1
N_{ind} = 4	*ModifiedMutationLength* = 1
N_{clones} = 6	*GeneTransferLength* = 1
N_{inf} = 3	*ModifiedGeneTransferLength* = 1
MAXLEN = 10	β = 0.2

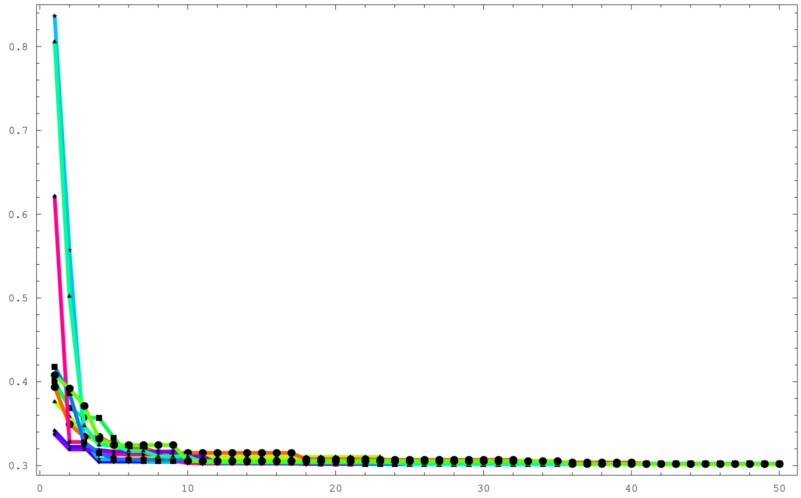

Figure 4: Simulation result using linear approximation

We can see, that the algorithm converged in all test runs within at most 40 iterations to an optimal solution. Even within 10 iterations, a good solution was found in all cases. When using forward selection, only a suboptimal solution with a 20% higher average squared error was found.

4 Conclusion and future work

Bacterial evolutionary algorithm for feature selection was discussed in this paper. We have extended our previous method to identify not only the optimal subset of a given size, but to find the optimal size of this subset, too. Additionally, we improved the bacterial operators which can change not only one element, but a longer segment of the chromosome to avoid local optima.

Future work will concentrate on finding optimal values for the parameters of the algorithm. Additionally, more in-depth comparisons with other methods using different data sets have to be carried out.

Acknowledgements

This research was supported by the National Research and Development Project Grant 2/0015/2002, a National Scientific Research Fund Grant OTKA T048832, and CEEPUS SK-42 Grant.

This work has been done in the framework of the K*plus* Competence Center Program which gratefully acknowledges support by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the K plus Competence Center Program.

References

[BDK04] J. Botzheim, M. Drobics, and L. T. Kóczy. Feature selection using bacterial optimization. In *Proc. 10th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 797–804, Perugia, July 2004.

[BFSO84] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, editors. *Classification and Regression Trees*. CRC Press, 1984.

[BHK$^+$02] J. Botzheim, B. Hámori, L. T. Kóczy, , and A. E. Ruano. Bacterial algorithm applied for fuzzy rule extraction. In *Proc. Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1021–1026, Annecy, France, 2002.

[Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 4:1157–1182, 2003.

[LKM01] M. Last, A. Kandel, and O. Maimon. Information-theoretic algorithm for feature selection. *Pattern Recognition Letters*, 22:799–811, 2001.

[NF99] N. E. Nawa and T. Furuhashi. Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Trans. Fuzzy Syst.*, 7:608–616, 1999.

ANALYSIS OF SPOT COUNTING ALGORITHMS IN FLUORESCENCE MICROSCOPY IMAGES

LEILA MURESAN, BETTINA HEISE

Introduction

A frequent task in medical image processing is to identify spots in fluorescence microscopy images. In the case of micro-arrays, the spots are due to DNA sequences labelled with fluorophores (Cy3 or Cy5), one or more per sequence. For classical microscopes, the high density of fluorophores cannot be well resolved, so only mean intensities of circular regions are computed. With $NanoScout^{®}$, single peaks (spots) are counted for each region of interest, and the relative abundance of the sequence (of each specific gene) can be determined even from very small samples.

From the image processing point of view peaks can be defined as bright, small, circular features, with little detail at the given resolution. We shall approximate them with a 2D Gaussian profile. We compare the results of two peak detection algorithms, *à trous* wavelets and robust background statistics. Since the true number of peaks is not known, several approaches to validate the result are discussed. Also the cases when one of the algorithms outperforms the other are identified.

1. Simple pre-processing of micro-arrays

The first step in micro-array analysis is dividing the 2GB image in smaller regions of interest, containing each one single spot. The division algorithm is performed on a binned version of the original micro-array image, so that one pixel is the average of a 10 × 10 pixels region of the original image. The binned input is thresholded (via the Otsu method) and cleaned of small blobs, resulting in a few clear spots.

Projecting the cleaned image on the two axis, an approximated grid can be reconstructed. Parallely, an artificial grid is built. In the binary cleaned image the best (brightest) spot's radius is selected as standard spot radius R and the median of the inter-spot distances D is computed. The two grids are fitted to each other, and the D is adjusted accordingly (figure 1)

Each spot is delimited by a rectangle described as a quadruple (left, upper, right, lower) points, and it offers the advantage of automatically determining background regions needed for signal-to-noise ratio analysis.

2. Spot detection algorithms

The spot detection algorithms that are analyzed in our study are:

(1) Mathematical morphology
(2) *À trous* wavelets
(3) Feature based statistics

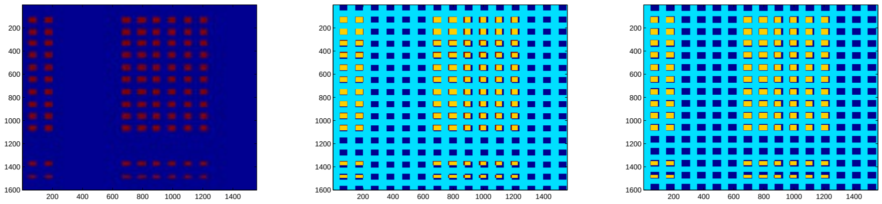

Figure 1. Approximated grid and overlap of the two griddings before and after fitting (dark blue - reconstructed grid, yellow - generated grid, red - difference)

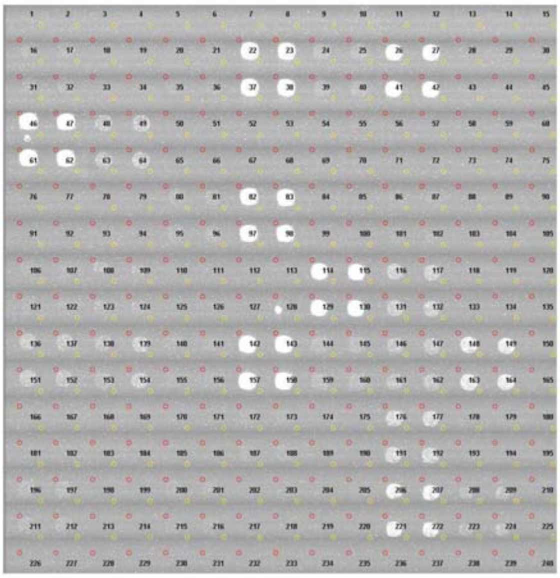

Figure 2. Result of the pre-processing step (red - left upper corner, yellow - right lower corner of the region of interest)

2.1. **Mathematical morphology.** We shall use the mathematical morphology methods in order to have a first approximate for the signal in the spot. This is the traditional approach to analyze micro-array data [1, 2], but in our case we shall use it only as a test or as to identify the density of signal peaks (high value - ensemble -regime, low value - single peak counting is possible). If the spot density is too high, the task of counting single spots becomes impossible (classical mean intensity methods have to be applied).

2.2. *À trous* **wavelets.** The *à trous* wavelet method, described in [4] consists of successive B-spline kernel convolutions. Initially the original image is convolved with the kernel K_0, a B-spline of order 3, $A_1 = Original * K_0$, where :

$$K_0 = \begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}$$

The smoothed image is than convolved with a kernel obtained from the kernel of the previous step, by inserting between each line and each column of the old kernel a line and a column of zeros, respectively.

The wavelet coefficients at step i are: $W_i(x,y) = A_i(x,y) - A_{i-1}(x,y)$. After J steps:

$$Original(x,y) = A_J(x,y) + \sum_{i=1}^{J} W_i(x,y) \tag{2.2.1}$$

The advantage of this image decomposition is the fact that real features tend to be persistent over the scales. So, if we set:

$$W_i(x,y) = 0, \frac{W_i(x,y) - \mu}{\sigma} < 3 \tag{2.2.2}$$

and compute

$$Spot_J(x,y) = \prod_{i=1}^{n} W_i(x,y) \tag{2.2.3}$$

the bigger the value of $Spot_J(x,y)$ the bigger the likelihood of a spot at location (x,y).

2.3. **Robust background statistics.** The robust background statistics method was thoroughly described in [3]. We only mention that is based on the modified z-score method for outlier detection. The outliers in several features are forming the set of spot candidates. This set is clustered in three subsets according to their acceptability level: the class of best, sharpest spots, acceptable spots and the uncertain / out-of-focus spots. The best results were obtained for the Gustafson-Kessel clustering algorithm. The features considered in this paper are: the mean intensity value over a window of size three and the intensity value of the Laplace -filtered image.

3. Performance criteria

In order to analyze the two algorithms first the results obtained for synthetic test images are compared. A summary of the results is given in table 1. The results of the robust background estimation method are better (even though the density and the noise level in some of the test images are worse than many in the real case).

For real data, in a totally analyzed microarray the correlation coefficient between *à trous* wavelet method and robust background can give a measure of the reliability of these methods. For the micro-array in figure 2 the correltaion coefficient is 0.865. The robust background method is affected if single spots cannot be recognized (approximately 20 spots cannot be directly analyzed for the micro-arry in figure 2).

Spots	Number of spots	*À trous* method	Robust background
Gaussian, N(0,0.08)	1000	506	967
Gaussian, N(0,0.08)	10000	467	1824
Poisson	1000	939	955
Poisson	10000	1198	6725

Table 1. Comparison of the results

However, if we elminate these worst cases, when single peaks cannot be detected, the correlation coefficient improves to 0.935.

The sensitivity of the new micro-array data can be tested also by making use of the empty spots. Some of the spots of the micro-array do not contain oligos (so should not contain peaks either). The position of these spots is known (in our case 34 such spots). The two algorithms are performed on these spots, and the mean m_{Empty} and standard deviation σ_{Empty} of the number of detected (false) peaks are computed.

Spots containing less then $m_{Empty} + 3 \cdot \sigma_{Empty}$ peaks are considered empty. The number of empty spots in the analyzed micro-array for the wavelet methods is 72, while for the robust background is 76.

The reliability of the result can be measured also by comparing the signal mean intensity with the variance of the background. The latter is approximated from the

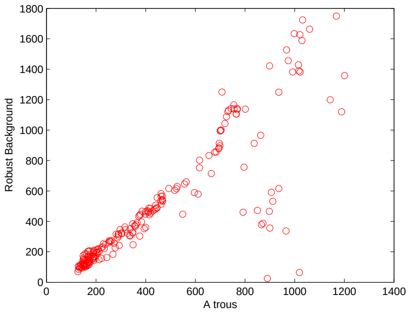

Figure 3. Result for *à trous* wavelets plotted against the robust background statistics method

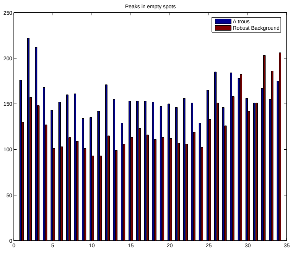

Figure 4. Peaks detected in empty spots. Blue - *à trous* wavelet method, red - robust background method

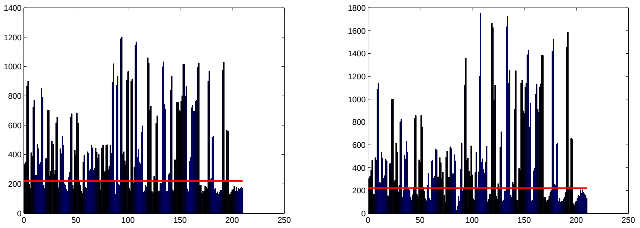

Figure 5. Peaks detected by *à trous* wavelet method and robust background method (for each spot). Red line - threshold under which the spot is considered empty

neighbouring background areas (as resulting from the detection of region of interest in 2). The signal intensity is measured for each of the resulting class as described above.

4. Conclusions

In this paper we presented some of the challenges of spot detection in fluorescence microscopy images, and we compared the performance of two distinct algorithms. The *à trous* wavelets method can handle a wider range of oligo densities, but seems somehow less sensitive then the robust background method. The latter has to be complemented by a simple algorithm, that determines the bi-modality of the intensity histogram (in order to detect the single-peak regime, in which this approach is powerful).

References

1. J. Angulo and J. Serra, *Automatic analysis of dna microarray images using mathematical morphology*, Bioinformatics **19** (2003), no. 5, 553–562.
2. Serra J., *Image analysis and mathematical morphology*, vol. 1, Academic Press, 1988.
3. L. Muresan and B. Heise, *Microarray image analysis*, Tech. report, Dept. of Knowledge-based Mathematical Systems, 2004.
4. J. C. Olivo-Marin, *Extraction of spots in biological images using multiscale products*, Pattern Recognition **35** (2002), 1989–1996.

Image Segmentation for DIC images of cells

Bettina Heise, Leila Muresan
Department of Knowledge-Based Mathematical Methods

Introduction

Well-performed image segmentation is the crucial point for further feature extraction and object classification. In the field of medical image processing the biological structures often are not clearly separable from background or are touching and overlapping each other. Especially if a classification by only slightly varying shape criteria is requested a careful segmentation is necessary.

Segmentation methods can be divided into three main groups: threshold based methods, region growing methods and edge based methods. Segmentation based on threshold mostly uses the intensity as criteria, but furthermore also texture features, scale or colors (for rgb-images) can be applied. Although global threshold methods are fast and can be automated (e.g. Otsu algorithm), they can fail if features are overlapping, e.g. if background intensity is not uniform, in case of high noise or in our special case of DIC images, (Fig.1). An improvement is achieved by applying local adaptive thresholds, (Fig.2). Nevertheless, the results are sensitive to noise and an additional offset depending on SNR must be used.

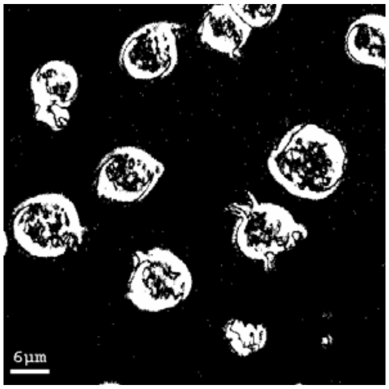

Fig.1: Bi-level threshold of Jurkat cell DIC image

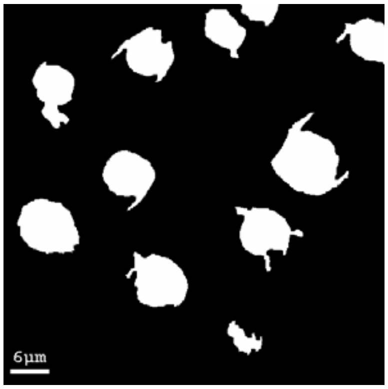

Fig2: Local adaptive threshold of the iterative Hilbert transformed DIC image

Region growing methods are performed either, starting with randomly distributed seeds or as regular image pyramid (Gaussian or Laplacian pyramid) merging and splitting neighbored regions in a strictly hierarchical way. The realization also can be done in a fuzzy version. The influence of noise can be reduced by an appropriate choice of merge and split parameters, but on the other side the whole iterative computation of the hierarchical structure is computational expensive.

Edge based methods suffer from the discontinuity of the edges and need additional edge linking methods.

PDE based image segmentation

As a further approach PDE-based methods for image segmentation are considered. Classical PDE- methods for image reconstruction can be written as

$$\begin{aligned}
&\frac{\partial u}{\partial t} + F(x,y,u,\nabla u,\nabla u^2) = 0 \quad in\, \Omega\, x(0,T) \\
&\partial_N u = 0 \qquad on\, \partial\Omega\, x(0,T) \qquad\qquad (1) \\
&u(x,y,0) = u_0(x,y)
\end{aligned}$$

where *$u_0(x,y)$* denominates the original image and *$u(x,y,t)$* the reconstructed image at time *t*. The choice of *F* should be balanced between two different targets, the edges have to be preserved and the noise should be smoothed. Non-linear diffusion by Perona-Malik can be an approach for these aims.

The general segmentation problem can be described by the Mumford-Shah Functional

$$J_{MS}(u,C) = \lambda \int_\Omega (u_0(x,y) - u(x,y))^2\, dxdy + \nu \int_{\Omega\backslash C} |\nabla u(x,y)|^2 \; dx\, dy + \mu\, Length(C) \qquad (2)$$

which has to be minimized. The first term describes the image similarity, the second term controls the smoothness of the area and the third term controls the smoothness of the contour *C*. λ, μ, ν are positive parameters. The problem can be further simplified by the restriction of *J_{MS}* to only piecewise constant functions *u*, in our case taking only two intensity values- c_1 equal to the average value of *u_0* inside the contour C and c_0 equal to the average value of *u_0* outside the contour C.

The evolution of the curve *C* can also be expressed by level set formulation:

$$\begin{aligned}
&C = \partial\omega = \{(x,y) \in \Omega : \phi(x,y) = 0\} \\
&inside(C) = \omega = \{(x,y) \in \Omega : \phi(x,y) > 0\} \quad . \qquad (3) \\
&outside(C) = \varpi = \{(x,y) \in \Omega : \phi(x,y) < 0\}
\end{aligned}$$

In many applications the intensity gradient on the boundary is low or vanishes nearly completely. By the introduction of an Heaviside function *H(Φ)* we can cope this problem and the formulation of the minimization problem by means of this function gives the Chan-Vese Functional [1]

$$\begin{aligned}
J_{CV}(c_1,c_2,\phi) &= \lambda_1 \int_\Omega (u_0 - c_1)^2 H(\phi) dxdy + \lambda_2 \int_\Omega (u_0 - c_2)^2 (1 - H(\phi)) dxdy \\
&+ \nu \int_\Omega H(\phi) dxdy + \mu \int_\Omega |\nabla H(\phi)|\, dxdy \quad . \qquad (4)
\end{aligned}$$

The minimization of this functional is equivalent to the solution of the PDE

$$\frac{\partial \varphi}{\partial t} = \delta_{\varepsilon}(\varphi)\left\{ \mu\, div(\frac{\nabla \varphi}{|\nabla \varphi|}) - \left[(u_0 - c_1)^2 - (u_0 - c_0)^2 \right] \right\} \tag{5}$$

with the initial value condition

$\varphi(i, j, t = 0) = \varphi_0(i, j)$.

It can be shown that the constants c_1 and c_0 are minimizing J_{CV} if

$$c_1(\phi) = \frac{\int_{\Omega} u_0\ H(\phi) dxdy}{\int_{\Omega} H(\phi) dxdy}, \quad c_0(\phi) = \frac{\int_{\Omega} u_0\ (1 - H(\phi)) dxdy}{\int_{\Omega} H(\phi) dxdy}. \tag{6}$$

Implementation and results

The method was numerically implemented as described in [2]. The Heaviside function *H(Φ)* and its derivative δ= *H'(Φ)* are realized in a smoothed version by

$$H_{2,\varepsilon}(x) = \tfrac{1}{2}[1 + \tfrac{2}{\pi}\arctan(\tfrac{x}{\varepsilon})] \qquad \delta_{2,\varepsilon}(x) = H'(x) = \tfrac{1}{\pi}\tfrac{\varepsilon}{\varepsilon^2 + x^2} \tag{7}$$

Tests were performed for different parameter sets (time step, ε, µ, number of iterations) and for different initial functions. The results are displayed in Fig.3a-c. The DIC images are transformed by iterative Hilbert Transform before applying the level set method.

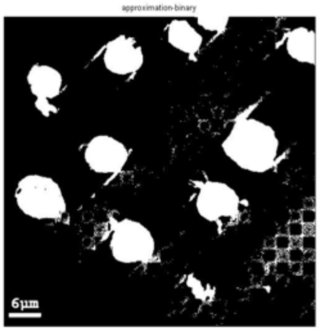

Fig.3a: Image segmentation into foreground–background by level set method,
δt=0.01, ε=0.5, 20 iterations

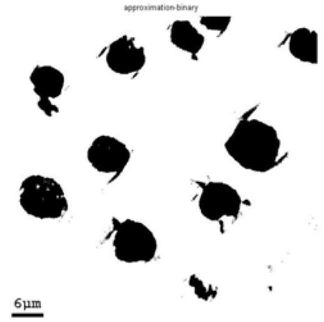

Fig.3b: Image segmentation into foreground–background by level set method,
δt=0.01, ε =1, 100 iterations

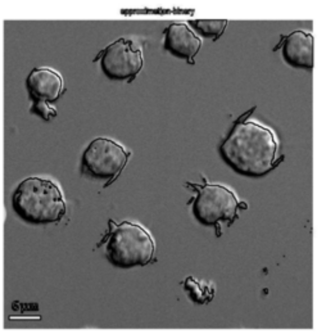

Fig.3c: Overlay of the finally extracted cell boundary with parameter setting as in Fig.3b

Also level set methods are sensitive to noise and results can converge to a noisy solution (Fig.3a). But by an appropriate choice of the parameter, especially of the Heaviside function slope parameter ε, a sufficient segmentation can be performed (Fig.3b and c). A further improvement -especially in our case of images suffering from a streaky noisy background structure-can be achieved by an anisotropic regularization formulation.

References:

1: Chan T., Vese L., *Active Contours without edges, IEEE Transaction on image processing*, 10(2), 266-277, 2001
2: Vese L., Chan T., *A multiphase level set framework for image segmentation using the Mumford and Shah model*, International Journal of Comp. Vision, 50(3), 271-293, 2002