



Maximizing vector distances using differential evolution—Relation to data redundancy

Martin Kolarik, Roman Jasek, and Zuzana Kominkova Oplatkova

Citation: [AIP Conference Proceedings](#) **1648**, 550019 (2015); doi: 10.1063/1.4912774

View online: <http://dx.doi.org/10.1063/1.4912774>

View Table of Contents: <http://scitation.aip.org/content/aip/proceeding/aipcp/1648?ver=pdfcov>

Published by the [AIP Publishing](#)

Articles you may be interested in

[Vector intensity reconstruction using the data completion method](#)

J. Acoust. Soc. Am. **133**, 2198 (2013); 10.1121/1.4792359

[Maximal dimension of invariant subspaces admitted by nonlinear vector differential operators](#)

J. Math. Phys. **52**, 043507 (2011); 10.1063/1.3574534

[Maximally symmetric vector propagator](#)

J. Math. Phys. **48**, 052306 (2007); 10.1063/1.2738361

[A phenomenological differential-relation-based vector hysteresis model](#)

J. Appl. Phys. **75**, 5484 (1994); 10.1063/1.355714

[On the use of a perceptual distance with speech vector quantization](#)

J. Acoust. Soc. Am. **75**, S59 (1984); 10.1121/1.2021511

Maximizing Vector Distances using Differential Evolution—Relation to Data Redundancy

Martin Kolarik, Roman Jasek and Zuzana Kominkova Oplatkova

Tomas Bata University in Zlin, Faculty of Applied Informatics, Nam. T. G. Masaryka 5555, 760 01 Zlin, Czech Republic

Abstract. This paper studies how redundant data affect maximizing of weighted distances of vectors in a set of vectors. To maximize distances differential evolution is used, because the problem does not have analytical solution and is complex. This paper at first describes suppressing of redundant data mathematically and then it checks this theoretical result in two experiments practically. As a result it was found that both experiments are in correspondence with theory.

Keywords: differential evolution, vector distances, optimization, data redundancy

INTRODUCTION

Maximizing of distances of vectors in a set of vectors with the help of weighting vectors' components is computationally complex problem. It has analytical solution only if vectors have two components and if distance metric is analytical (like arithmetic average is). Longer vectors or statistical metrics lead to near to infinite complexity.

We have successfully tested ([6]) differential evolution ([3]) as one of possible approaches of reducing the complexity of the problem. Differential evolution fits well, because in our case maximizing distances means looking for the optimal vector of component weights, what corresponds to DE principle—DE is a stochastic optimization algorithm working over vectors of real numbers.

Maximization assigns weights to components with the aim to move each vector as far from others as possible, thus components more contributing to vector distances obtain bigger weights. Weights intuitively express importance of components, what is likely somehow linked to informational content of components, so consequently components assigned bigger weights may carry bigger amount of information. If this consideration was true optimization problem would change into data-mining technique.

Particularly, authors analyze this approach as one of possibilities of determining importance of persons' features as a part of biometric identification system.

Emphasizing some components should also suppress other components containing similar data, because the information in similar components is nearly the same. We explored this idea in this paper. At first we described the idea using mathematics, at second we proposed and executed two experiments to validate these theoretical presumptions.

THEORY

Vectors, Set of Vectors, Weight Vector

Vector is an ordered row of numbers:

$$\vec{a} = \{a_1, a_2, \dots, a_n\} = \{a_i\}$$

where a_i are components of the vector \vec{a} and n is number of vector's components.

Many vectors of the same size form together a set \mathcal{S} of vectors. Particular vector component i occupies always the same position in all vectors:

$$\vec{a}_j = \{a_{j1}, a_{j2}, \dots, a_{jn}\} = \{a_{ji}\}, \quad j \in 1, 2, \dots, s \quad (1)$$

where j indexes vectors \vec{a}_j in the set \mathcal{S} , i indexes components a_{ji} of vectors and s denotes size of the set. We always used weighted vectors in experiments, where significance of particular component is expressed using weight.

The weight vector \vec{w}_i is single standalone vector applied to components (compare it to (1)):

$$\vec{a}_{wj} = \{w_i a_{ji}\} \quad (2)$$

$$\sum_i w_i = 1 \quad (3)$$

Distance Metric, Distance of Population

We used Euclidean metric to measure distances of two vectors. Note that distances were always computed using weighted vectors \vec{a}_{wj} , see (2), usage of weights allowed us to keep source vectors \vec{a}_j unchanged. Euclidean distance:

$$d_{jk}^2 = d^2(\vec{a}_{wj}, \vec{a}_{wk}) = \sum_i w_i^2 (a_{ji} - a_{ki})^2, \quad \text{see [4]} \quad (4)$$

Because vectors in the set are s and their mutual weighted distances are $\frac{1}{2}s(s-1)$, we needed to somehow compute single principal number $d_{\mathcal{S}}$ representing a weighted distance of the whole set \mathcal{S} . According to [6] we used arithmetic average of mutual distances:

$$\begin{aligned} \mathcal{D} &= \{d_{jk}\}, \quad j \neq k, \quad j, k \in \mathcal{S} \quad \text{is a set of all mutual distances} \\ d_{\mathcal{S}} &= \frac{2}{s(s-1)} \sum_{\mathcal{D}} d_{jk} \quad \text{is a distance representing whole set } \mathcal{S} \end{aligned} \quad (5)$$

Differential Evolution

Differential evolution (DE) is stochastic evolutionary algorithm, which operates over vectors of real numbers. During computation the algorithm combines (crosses) firstly randomly spread instances of vectors to minimize cost function.

Original algorithm of DE (as published in [3]) has been still being improved to obtain better convergence or speed or both. Among many recommended variants we used the variant tailored for small population sizes named *DE/best/1 with jitter*, see [5], which computes offspring candidate using the following formula (6):

$$\vec{v}_{g+1}^m = \vec{v}_{\text{best}} + F \left(\vec{b}_g - \vec{c}_g \right) + J_m \left(\vec{b}_g - \vec{c}_g \right) \quad (6)$$

where \vec{b}_g, \vec{c}_g are members randomly selected from DE's current generation for each generation member \vec{v}_g^m , m iterates over all DE's generation members, g is generation index, F is differential weight and J_m is random jitter in [0..0.0001] different for each \vec{v}_g^m .

Then the offspring replaces parent \vec{v}_g^m if $\text{rand}(0..1) \leq C_r$, where C_r is crossover probability.

We used the following DE parameters recommended by [5] as a good start in our experiments:

$$\begin{aligned} F &= 1.0 && \text{(differential weight)} \\ C_r &= 0.5 && \text{(crossover probability)} \\ P &= 100 && \text{(number of population members)} \\ G &= 150 && \text{(number of DE's generations)} \end{aligned} \quad (7)$$

Duplicating Vector Component

This section describes change of d_{jk} (see (4)) if one component in vectors is duplicated.

Firstly original d_{Ojk} (vectors j and k are represented with \vec{i} and \vec{u}) with no component duplicated is computed. Both vectors \vec{i}_O and \vec{u}_O are weighted (see (2)) and belong to \mathcal{S} :

$$\begin{aligned}
\vec{t}_O &= (w_1 t_1, w_2 t_2, \dots, w_i t_i) \\
\vec{u}_O &= (w_1 u_1, w_2 u_2, \dots, w_i u_i) \\
d_O^2(\vec{t}_O, \vec{u}_O) &= w_1^2 (u_1 - t_1)^2 + w_2^2 (u_2 - t_2)^2 + \dots + w_i^2 (u_i - t_i)^2 = \\
&= C + w_i^2 (u_i - t_i)^2
\end{aligned} \tag{8}$$

Secondly, component i in both vectors is duplicated, and resulting d_{Djk} is computed. Note that duplication requires splitting of w_i because (3) must still be valid. Splitting ratio $r \in [0..1]$ is arbitrary, so $w_i = r w_i + (1 - r) w_i$, where the first addend weights original component and the second addend weights duplicated component:

$$\begin{aligned}
\vec{t}_D &= (w_1 t_1, w_2 t_2, \dots, r w_i t_i, (1 - r) w_i t_i) \\
\vec{u}_D &= (w_1 u_1, w_2 u_2, \dots, r w_i u_i, (1 - r) w_i u_i) \\
d_D^2(\vec{t}_D, \vec{u}_D) &= w_1^2 (u_1 - t_1)^2 + w_2^2 (u_2 - t_2)^2 + \dots + r^2 w_i^2 (u_i - t_i)^2 + (1 - r)^2 w_i^2 (u_i - t_i)^2 = \\
&= C + w_i^2 (u_i - t_i)^2 (r^2 + (1 - r)^2)
\end{aligned} \tag{9}$$

(8) and (9) differs only in term $R = r^2 + (1 - r)^2$, which completely decides about resulting distance change. Straightforward application of calculus shows that R is minimal for $r = 0.5$ what consequently means that any other r gives bigger R . Maximal $R = 1$, giving $d_D = d_O$, is obtained if either $r = 1$ or $r = 0$.

The result is principal because it means that if DE is ordered to maximize distances it should suppress either original or duplicated component just by assigning full w_i to either of them.

EXPERIMENTS

In order to test theoretical result we prepared two experiments. In both experiments, we created a set of vectors with 20 components and populated these components with random variables of normal distribution $N(\mu = 0, \sigma = 3)$.

In both experiments we used $-d_{\mathcal{F}}$ (see (5)) as cost function for differential evolution.

Duplication of Components

Description. We designed this experiment to test simple component duplication. We duplicated each second component, so we took vectors with 30 components in total containing 10 pairs of duplicates. Then we ran differential evolution with parameters described in (7) 1000 times. Because we had 10 pairs of duplicated components in each run, we in total obtained 10000 pairs of duplicates to be analyzed.

Results. As theory described above presumes, we expected that in each pair one weight ends close to zero. Our experiment ended with 9929 pairs corresponding to the theory and with 71 pairs not corresponding to the theory, so experiment confirmed theory in 99.29% cases.

We expected complete concordance of theory and experiment to be unlikely, because DE is stochastic algorithm which may not always find the solution. That is also why we considered the theory to be confirmed.

Linear Combination of Components

Description. We designed this experiment to test if and how differential evolution resolves more complex component mixing. We took pairs $p \in [1..10]$ of original components a_i and we created new component as $a_{\text{new}} = 0.1 p a_{(2p-1)} + 0.1 (10 - p) a_{(2p)}$. We obtained vectors with 30 components in total containing 10 triples of combined components. Then we ran differential evolution with parameters described in (7) 1000 times. Because we had 10 triples of combined components in each run, we in total obtained 10000 of triples to be analyzed.

Results. Although we did not express the problem mathematically, we expected that after maximizing distances at most two components may keep their weights and that the third component should be suppressed. It also should be arbitrary which two components are kept and which one is eliminated.

Results showed that 42 triples had all three weights nonzero and that 9843 triples had at least one weight zeroed, what gives 99.58 % rate of correspondence to expectations.

We expected complete concordance of theory and experiment to be unlikely, because DE is stochastic algorithm which may not always find the solution.

CONCLUSION

Vectors with the same length composing a some set of vectors may be adjusted in a way that their distances increase. The increase helps, for example, with looking for the best match of some unknown vector to some vector of the set. Increasing cannot change vectors unlimitedly, because their distances would spread infinitely, and that is why we were changing only weights of vectors' components, leaving components themselves untouched.

Spreading of vectors must prefer vectors' components contributing to distance more than others. Intuitively, "more contribution" is somehow connected to informational content of components and that is the topic we were exploring in this paper. Particularly, we focused to special cases when some vectors' component is duplicated or made as linear combination of others. We assumed that duplicates should have been eliminated and we also proved it using mathematical deduction.

The problem of maximizing of weighted vectors is continuous, of almost infinite complexity and having no analytical solution. To decrease the complexity we choose differential evolution, which is by experience of authors capable of resolving the problem.

To confirm mathematical prove in real situation we proposed and executed two experiments, one with duplicated component and second one with linearly combined components. Both experiments ran over set of 30 vectors with 30 components with data populated using random variables. Duplicates and combinations were made artificially. In each experiment we ran 1000 runs of differential evolution.

Results of both experiments showed that duplicated or combined data were suppressed in a way that only single data representative survived.

ACKNOWLEDGEMENT

This work was supported by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089, by Internal Grant Agency at Tomas Bata University in Zlin, project nr. IGA/FAI/2014/054 and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co- financed by ESF and state budget of the Czech Republic.

REFERENCES

1. Jain, A. K., Ross A., 2008. *Handbook of Biometrics*. Springer, ISBN 978-0-387-71040-2.
2. Spall, J. C., 2003. *Introduction to Stochastic Search and Optimization*. Wiley, ISBN 0-471-33052-3.
3. Price, K., Storn, R. M. and Lampinen, J. A., 2005. Natural computing series. *Differential evolution: A Practical Approach To Global Optimization*. Iss. 1.
4. Deza M. M., Deza E., 2009. *Encyclopedia of Distances* Springer, ISBN 3-642-00233-5.
5. Storn R. *Differential Evolution (DE)*. Online, <http://www1.icisi.berkeley.edu/~storn/code.html>, cited 2014-03-02.
6. Kolarik, M., Jasek, R., Kominkova-Oplatkova, Z., 2014. Maximizing Vector Distances for Purpose of Searching—A Study of Differential Evolution Suitability in *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, pp. 419–428. Springer, ISBN 978-3-319-08155-7.