

6-2011

Why Fuzzy Transform Is Efficient in Large-Scale Prediction Problems: A Theoretical Explanation

Irina Perfilieva
University of Ostrava

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Follow this and additional works at: https://scholarworks.utep.edu/cs_techrep



Part of the [Computer Engineering Commons](#)

Comments:

Technical Report: UTEP-CS-11-21a

Published in *Advances in Fuzzy Systems*, Vol. 2011, Article ID 985839, doi:10.1155/2011/985839.

Recommended Citation

Perfilieva, Irina and Kreinovich, Vladik, "Why Fuzzy Transform Is Efficient in Large-Scale Prediction Problems: A Theoretical Explanation" (2011). *Departmental Technical Reports (CS)*. 607.
https://scholarworks.utep.edu/cs_techrep/607

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.

Why Fuzzy Transform Is Efficient in Large-Scale Prediction Problems: A Theoretical Explanation

Irina Perfilieva¹ and Vladik Kreinovich²

¹University of Ostrava

Inst. for Research and Applications of Fuzzy Modeling
Ostrava, Czech Republic, Irina.Perfilieva@osu.cz

²Department of Computer Science

University of Texas at El Paso
El Paso, TX 79968, USA, vladik@utep.edu

Abstract

In many practical situations like weather prediction, we are interested in large-scale (averaged) value of the predicted quantities. For example, it is impossible to predict the exact future temperature at different spatial locations, but we can reasonably well predict average temperature over a region. Traditionally, to obtain such large-scale predictions, we first perform a detailed integration of the corresponding differential equation, and then average the resulting detailed solution. This procedure is often very time-consuming, since we need to process all the details of the original data.

In our previous papers, we have shown that similar quality large-scale prediction results can be obtained if instead, we apply a much faster procedure – first average the inputs (by applying an appropriate fuzzy transform), and then use these averaged inputs to solve the corresponding (discretization of the) differential equation.

In this paper, we provide a general theoretical explanation of why our semi-heuristic method works, i.e., why fuzzy transforms are efficient in large-scale predictions.

1 Formulation of the Problem

Predictions are needed. One of the main objectives of science is to predict the future values of the physical quantities. For example, it is desirable to predict tomorrow's weather, the weather for several days ahead, etc. For a spreading flu epidemics, it is desirable to predict how this epidemics will spread if we do not introduce any restrictions on travel – and how this spread will change if such restrictions are introduced.

Detailed predictions are often impossible. Of course, ideally, it is desirable to have predictions which are as detailed as possible. For example, ideally, we would like to know the exact value of tomorrow's temperature and wind speed at all possible spatial locations within a given region – or to predict exactly where the epidemics will spread and exactly how many people will fall ill if we do not introduce any travel restrictions.

However, in many practical situations, such a detailed prediction is impossible. In some of these situations, prediction is potentially possible, but it requires such a large amount of computations that even on the fastest modern computers, the computations finish long after the future event (that we are trying to predict) has already occurred.

Large-scale predictions are usually sufficient. In many practical situations in which we cannot predict the *exact* values of the future quantities, it is often sufficient to predict the *average* values of the future quantities, averaged over certain areas.

For example, from the practical viewpoint, even though we cannot predict the exact value of tomorrow's temperature at all possible spatial locations, it would be beneficial to predict the *average* temperature over a given small geographic region. Similarly, for an epidemic, even though we are unable to predict where exactly it will spread, and how many people will fall ill in different small towns, it is very beneficial to be able to predict how many people *on average* will get ill in the region.

For predicting time series – e.g., financial time series formed by the prices of different stocks at different moments of time, though it is impossible to predict the exact values of the future prices, it is desirable to at least be able to predict the *trends*, i.e., the prices averaged over a certain time period.

Comment. For clarity and simplicity, in the following text, we will describe the case when both the input $x(t)$ and the output $y(t)$ depend only on time t . The exact same formulas can also be applied if we have a spatial dependence; in this case, t and s are the corresponding spatial points.

Towards a precise mathematical description of quantities predicted by large-scale prediction. Instead of predicting the values $y(t)$ for different moments of time t , we predict the weighted averages $\bar{y}(t)$, i.e., the average of the values $y(s)$ for the values s which are close to t .

It is reasonable to assume that for different moments t we use the same averaging, i.e., that the weight with which the value $y(s)$ contributes to $\bar{y}(t)$ depends only on the difference $t - s$ and not on the absolute values of t or s . Under this assumption, the general formula for the weighted average takes the form

$$\bar{y}(t) = \int w(t - s) \cdot y(s) ds, \quad (1)$$

where all the weights are non-negative and for each t , the total weight of all the values $y(s)$ is equal to 1:

$$\int w(s) ds = 1. \quad (2)$$

An example and a useful equivalent reformulation of averaging. A natural example of such averaging is a *Gaussian averaging*, where we use Gaussian weights:

$$w(s) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(-\frac{s^2}{2\sigma^2}\right). \quad (3)$$

It is often convenient to represent this Gaussian weight function as

$$w(s) = \text{const} \cdot W(s), \quad (4)$$

where the new weight function $W(s)$ is described by a simpler formula

$$W(s) = \exp\left(-\frac{s^2}{2\sigma^2}\right). \quad (5)$$

This new weight function satisfies the property ($W(0) = 1$ and)

$$\max_s W(s) = 1. \quad (6)$$

Large-scale quantities and fuzzy transform. A similar representation is often useful for other weight functions as well. In general, once we know this new weight function $W(s)$, we can use the normalized condition (2) to find that

$$w(s) = \frac{W(s)}{\int W(t) dt}. \quad (7)$$

Thus, in terms of the new weight function $W(s)$, the weighted average (1) takes the form

$$\bar{y}(t) = \frac{\int W(t-s) \cdot y(s) ds}{\int W(s) ds}. \quad (8)$$

Expression (8) is a particular case of the expression of a *fuzzy transform* [4, 5, 6] which is, in general, defined as

$$Y = \frac{\int A(s) \cdot y(s) ds}{\int A(s) ds} \quad (9)$$

for some function $A(s) \geq 0$ for which $\max_s A(s) = 1$. For a special uniform case [5, 6], we have several functions $A(s)$ of the form $A_n(s) = W(t_n - s)$, where $W(s)$ is a given function. The corresponding values Y_n of the fuzzy transforms are then equal to

$$Y_n = \frac{\int A_n(s) \cdot y(s) ds}{\int A_n(s) ds} = \frac{\int W(t_n - s) \cdot y(s) ds}{\int W(s) ds}, \quad (10)$$

i.e., coincide with the values $\bar{y}(t_n)$ corresponding to different points t_n .

Thus, from the mathematical viewpoint, the weighted averages are simply the values of the fuzzy transform.

Typical prediction procedure: solving a differential equation. Most relations in physics are described by differential equations. In particular, the relation between the observed signals $x(t)$ and the predicted values $y(t)$ can also be described by a differential equation.

Traditional procedure for large-scale predictions. Since prediction usually means solving a known differential equation, a usual procedure for large-scale predictions is as follows:

- first, we use the known values $x(t)$ to solve the differential equations and get the values $y(t)$;
- then, we apply the weighted average procedure (8) to the resulting values $y(t)$, and get the desired large-scale predictions $\bar{y}(t)$.

Drawbacks of the traditional procedure. The main drawback of the traditional procedure is that we spend a lot of computation time to get a detailed solution $y(t)$ – but at the end, we only return a few values corresponding to large-scale predictions.

For example, in weather prediction, we spend hours of computer time on high-performance supercomputers to solve a complex system of differential equations with thousand of variables – and then only use the large-scale weighted average of this solution.

Natural idea. We are only interested in *large-scale* predictions, i.e., only in the weighted *averages* of the result $y(t)$ of solving the differential equation, averages that ignore the fine structure of the solution $y(t)$. So why not start with the averaged values of the input $x(t)$, i.e., why not ignore the fine structure of $x(t)$ from the very beginning – and thus, save computation time.

In other words:

- traditionally, we first *integrate* the differential equation, and then *average* the solution;
- what we propose is that we first *average*, and only then *integrate*; in this manner, we will need fewer values to integrate and thus, less computation time.

Empirically, this idea seems to work. For several differential equations, we implemented the above idea of how to speed up computations. Specifically,

- instead of the original input $x(t)$, we use the fuzzy transform values X_1, \dots, X_n ,
- then we use the values X_i in the discretized version of the original differential equation, and

- we use the results Y_1, \dots, Y_n of this solution as an estimate for the desired large-scale averages (= fuzzy transform of $y(t)$).

Surprisingly, we got a very good approximation to the values Y_i computed based on the detailed $y(t)$ [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12].

What we do in this paper. In this paper, we provide a theoretical explanation for the empirical success of the fuzzy-transform-based methods of speeding up computations.

This explanation makes us confident that this fuzzy transform technique can be successfully used in other large-scale prediction problems as well.

2 Theoretical Explanation

Linearization. Usually, the effect of each input value $x(t)$ on the prediction results is small. In this sense, we can say that the inputs are relatively small. Thus, we can use the standard technique of dealing with dependence on small value:

- extend the dependence of $y(t)$ on $x(s)$ in Taylor series,
- ignore quadratic and higher order terms, and thus
- keep only linear terms in this dependence.

In this case, we get the following dependence:

$$y(t) = y_0(t) + \int y_1(t, s) \cdot x(s) ds, \quad (11)$$

for some functions $y_0(t)$ and $y_1(t, s)$.

Shift-invariance. We are interesting in systematic predictions, predictions that need to be repeated again and again. In these predictions, there is no fixed moment of time: if we start with the same input repeated later (i.e., shifted in time, from $x(t)$ to $x_{\text{new}}(t) = x(t - t_0)$), we get the same result (similarly shifted) $y_{\text{new}}(t) = y(t - t_0)$.

For the formula (11), this shift-invariance means that

- first, we must have $y_0(t) = y_0(t - t_0)$ for all t and t_0 ; in particular, for $t_0 = t$, we conclude that $y_0(t) = y_0(0)$, i.e., that y_0 should not depend on time at all: $y_0(t) = y_0$;
- second, we must have $y_1(t, s) = y_1(t - t_0, s - t_0)$ for all t , s , and t_0 ; in particular, for $t_0 = s$, we conclude that $y_1(t, s) = y_1(t - s, 0)$ and that the function $y_1(t, s)$ should only depend on the difference $t - s$.

Thus, we arrive at the following dependence:

$$y(t) = y_0 + \int y_1(t - s) \cdot x(s) ds. \quad (12)$$

Main result: formulation. In the traditional approach, we first find the detailed output (12) and then average it by applying the averaging

$$\bar{y}(t) = \frac{\int W(t-s) \cdot y(s) ds}{\int W(s) ds}. \quad (13)$$

An alternative approach is to first apply the same averaging to the original signal $x(t)$, resulting in

$$\bar{x}(t) = \frac{\int W(t-s) \cdot x(s) ds}{\int W(s) ds}, \quad (14)$$

and try use this averaged signal $\bar{x}(t)$ as the input to the corresponding dynamical systems (i.e., in effect, to the transformation (12)):

$$\bar{y}_f(t) = y_0 + \int y_1(t-s) \cdot \bar{x}(s) ds. \quad (15)$$

Our claim is that these two approaches always lead to the same result, i.e.,

$$\bar{y}_f(t) = \bar{y}(t) \quad (16)$$

for all moments of time t .

Proof. In terms of the normalized weight function (7), the original signal has the form

$$\bar{y}(t) = \int w(t-s) \cdot y(s) ds, \quad (17)$$

where $y(s)$ is determined by the formula (12). Substituting the expression

$$y(s) = y_0 + \int y_1(s-u) \cdot x(u) du \quad (18)$$

into the formula (17), we conclude that

$$\bar{y}(t) = y_0 + \int w(t-s) \cdot y_1(s-u) \cdot x(u) ds du, \quad (19)$$

i.e., that

$$\bar{y}(t) = y_0 + \int \bar{w}(t, u) \cdot x(u) du, \quad (20)$$

where

$$\bar{w}(t, u) \stackrel{\text{def}}{=} \int w(t-s) \cdot y_1(s-u) ds. \quad (21)$$

Similarly, in terms of the normalized weight function $w(t)$, we have

$$\bar{x}(t) = \int w(t-s) \cdot x(s) ds. \quad (22)$$

Substituting the corresponding formula

$$\bar{x}(s) = \int w(s-u) \cdot x(u) du \quad (23)$$

into the expression (15) for $\bar{y}_f(t)$, we conclude that

$$\bar{y}_f(t) = y_0 + \int y_1(t-s) \cdot w(s-u) \cdot x(u) ds du, \quad (24)$$

i.e., that

$$\bar{y}_f(t) = y_0 + \int \bar{w}_f(t, u) \cdot x(u) du, \quad (25)$$

where

$$\bar{w}_f(t, u) \stackrel{\text{def}}{=} \int y_1(t-s) \cdot w(s-u) ds. \quad (26)$$

In view of the formulas (20) and (25), to prove that the values $\bar{y}(t)$ and $\bar{y}_f(t)$ always coincide, it is sufficient to prove that the corresponding functions $\bar{w}(t, u)$ and $\bar{w}_f(t, u)$ coincide for all t and u . These functions are defined by expressions (21) and (26).

To prove that these expressions coincide, let us try to transform them into each other. In the expression (26), we take the value of the normalized weight function $w(t)$ at the point $s-u$. In contrast, in the expression (21), we use the value $w(t-s)$ for the corresponding auxiliary variable s . To transform the expression (26) into the form (21), let us introduce a new auxiliary variable v for which $s-u = t-v$. From this formula, we conclude that $s = t+u-v$, hence $t-s$ takes the form $t-(t+u-v) = v-u$. Thus, in terms of the new variable v , the integrated expression in (26) takes the form

$$y_1(t-s) \cdot w(s-u) = y_1(v-u) \cdot w(t-v) = w(t-v) \cdot y_1(v-u). \quad (27)$$

Hence, the integrals of these two expressions must also coincide:

$$\int y_1(t-s) \cdot w(s-u) ds = \int w(t-v) \cdot y_1(v-u) dv. \quad (28)$$

The right-hand side of this equality is exactly the expression (21) – the only difference is that we use a different name for the integration variable (v instead of s). Thus, the functions $\bar{w}(t, u)$ and $\bar{w}_f(t, u)$ indeed coincide – and hence, $\bar{y}_f(t) = \bar{y}(t)$.

The equality is proven.

Comment. In the ideal case, when quadratic terms can be completely ignored and there is no dependence on absolute time, the new method leads to *exact* same large-scale predictions as the traditional one. In practice, if we take into account that

- the quadratic terms are small but non-zero, and that

- there may be an underlying trend-like dependence on absolute time (like global warming in weather prediction)

we end up with *approximate* equality between the traditional and fuzzy-transform based predictions – and this approximate equality is what we observed in our experiments [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12].

Since large-scale predictions are approximate anyway, this approximate equality means that in terms of accuracy, the new predictions are, in effect, as good as the traditional ones. Since the new predictions are much faster to compute, they have a clear practical advantage.

Acknowledgments.

This work was supported in part by the National Science Foundation grant HRD-0734825, by Grant 1 T36 GM078000-01 from the National Institutes of Health, and by Grant MSM 6198898701 from MŠMT of Czech Republic. The authors are thankful to the anonymous referees for valuable suggestions.

References

- [1] F. Di Martino, V. Loia, and S. Sessa, “Fuzzy transforms method and attribute dependency in data analysis”, *Information Sciences*, 2010, Vol. 180, pp. 493–505.
- [2] F. Di Martino, V. Loia, and S. Sessa, “Fuzzy transforms method in prediction data analysis”, *Fuzzy Sets and Systems*, to appear.
- [3] V. Novák, M. Štěpnička, I. Perfilieva, and V. Pavliska, “Analysis of periodical time series using soft computing methods”, *Proceedings of FLINS’2008*, World Scientific, Singapore, 2008, pp. 55–60.
- [4] I. Perfilieva, “Fuzzy transforms”, In: J. F. Peters et al. (Eds.), *Transactions on Rough Sets II*, Springer Lecture Notes in Computer Science, 2004, Vol. 3135, pp. 63–81.
- [5] I. Perfilieva, “Fuzzy transforms: theory and applications”, *Fuzzy Sets and Systems*, 2006, Vol. 157, pp. 993–1023.
- [6] I. Perfilieva, “Fuzzy transforms: a challenge to conventional transforms”, *Advances in Imaging and Electron Physics*, 2007, Vol. 47, pp. 137–196.
- [7] I. Perfilieva, V. Novák, and A. Dvořák, “Fuzzy transform in the analysis of data”, *International Journal of Approximate Reasoning*, 2008, Vol. 48, No. 1, pp. 36–46.
- [8] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, “Analysis and Prediction of Time Series Using Fuzzy Transform,” *Proc. IEEE World Congress on Computational Intelligence*, Hong Kong, 2008, pp. 3875–3879.

- [9] M. Štěpnička, A. Dvořák, V. Paviska, and L. Vavříčková, “Linguistic Approach to Time Series Analysis and Forecasts”, In: *Proceedings of the 2010 IEEE World Congress on Computational Intelligence WCCI'2010*.
- [10] M. Štěpnička, A. Dvořák, V. Paviska, and L. Vavříčková, “Fuzzy approach to time series analysis: partII – Problems of linguistic descriptions”, In: *Abstracts of the FSTA'10*, 2010, p.127.
- [11] M. Štěpnička, A. Dvořák, V. Paviska, and L. Vavříčková, “A linguistic approach to time series modeling with help of the F-transform”, *Fuzzy Sets and Systems*, to appear.
- [12] M. Štěpnička, V. Pavliska, V. Novák, and I. Perfilieva, “Time series analysis and prediction based on fuzzy rules and the fuzzy transform”, *Proceedings of IFSA World Congress/EUSFLAT Conference*, Lisbon, Universidade Técnica de Lisbon, 2009, pp. 483–488.