

Why ℓ_p -methods in Signal and Image Processing: A Fuzzy-Based Explanation

Fernando Cervantes¹, Bryan Usevitch¹,
and Vladik Kreinovich²

¹Department of Electrical and Computer Engineering

²Department of Computer Science

University of Texas at El Paso

500 W. University

El Paso, TX 79968, USA

fcervantes@miners.utep.edu, usevitch@utep.edu

vladik@utep.edu

Abstract—In signal and image processing, it is often beneficial to use semi-heuristic ℓ_p -methods, i.e., methods that minimize the sum of the p -th powers of the discrepancies. In this paper, we show that a fuzzy-based analysis of the corresponding intuitive idea leads exactly to the ℓ_p -methods.

I. FORMULATION OF THE PROBLEM

In general, signal and image reconstruction are ill-posed problems. While cameras and other image-capturing devices are getting better and better every day, none of them is perfect, there is always some blur. This blur comes from the fact that while we would like to capture the intensity $I(x, y)$ at each spatial location (x, y) , the signal captured by a real-life measuring device is influenced not only by the intensity $I(x, y)$ at the desired location (x, y) , but also by the intensities $I(x', y')$ at nearby locations (x', y') . As a result, instead of reflecting the intensity $I(x, y)$ at the desired point (x, y) , the signal $s(x, y)$ measured by the device is a combination of intensities at the point (x, y) and at the nearby points:

$$s(x, y) = \int w(x, y, x', y') \cdot I(x', y') dx' dy',$$

for appropriate weights $w(x, y, x', y')$.

When we take a photo of a friend with a modern sophisticated cell phone camera, this blur is barely visible – and does not constitute a serious problem. However, when a spaceship takes a photo of a distant planet, the blur is very visible – and needs to be eliminated. In such situations, we need to reconstruct the original image $I(x, y)$ from the blurred image $s(x, y)$.

From the purely mathematical viewpoint, this reconstruction problem is *ill-posed* in the sense that large changes in $I(x, y)$ can lead to very small changes in $s(x, y)$ – and, as a result, unless we impose additional constraints on the original image $I(x, y)$, we cannot reconstruct the image with any reasonable accuracy. This mathematical feature is easy to explain: blurring averages the image. Instead of the original intensity at the point (x, y) , we have an average intensity over all the neighbors (x', y') of the original point (x, y) . It is known that such averaging eliminates high-frequency components. Thus, if

instead of the original signal $I(x, y)$, we consider a different signal

$$I^*(x, y) = I(x, y) + c \cdot \sin(\omega_x \cdot x + \omega_y \cdot y),$$

with a high-frequency component added, the resulting signal $s^*(x, y)$ will be practically the same $s^*(x, y) \approx s(x, y)$ – but the original image, for large c , may be very different.

To be able to reconstruct the image reasonably uniquely, we cannot allow all possible dependencies $I(x, y)$, we need to impose some additional conditions on the original image. This imposition of additional conditions that helps reconstruct the original image is known as *regularization*; see, e.g., [9].

Similarly, the problem of reconstructing a 1-D signal $x(t)$ from observations is ill-posed.

Tikhonov's regularization: a brief reminder. If a signal or an image is smooth (differentiable), then a natural idea is to require that the corresponding derivatives are, on average, small, i.e., e.g., that the mean square value of the derivative does not exceed a certain constant C .

Let us describe this requirement in precise terms. If we have n values d_1, \dots, d_n , then the mean square value is

$$d \stackrel{\text{def}}{=} \sqrt{\frac{d_1^2 + \dots + d_n^2}{n}}.$$

The requirement that this mean square value is bounded by C , i.e., that

$$\sqrt{\frac{d_1^2 + \dots + d_n^2}{n}} \leq C$$

is equivalent to

$$\frac{d_1^2 + \dots + d_n^2}{n} \leq C^2$$

and, thus, to

$$d_1^2 + \dots + d_n^2 \leq c,$$

where $c \stackrel{\text{def}}{=} n \cdot C^2$.

When we go from the discrete data to the continuous signal or image, then the sum turns into an integral. So, for 1-D signals, we have a constraint

$$\int (\dot{x}(t))^2 dt \leq c,$$

and for 2-D images $I(x, y)$, we have a similar constraint

$$\int \left(\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right) dx dy \leq c.$$

Out of all signals or images that satisfy this constraint, we want to find a one which is the best fit with the observation, i.e., for which, e.g., the mean square error is the smallest:

$$J \stackrel{\text{def}}{=} \sum_i e_i^2 \rightarrow \min,$$

where e_i is the difference between the value measured in the i -th measurement and the value predicted based on the corresponding signal or image. Thus, we need to minimize J under the constraint

$$\int (\dot{x}(t))^2 dt \leq c$$

or

$$\int \left(\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right) dx dy \leq c.$$

In general, constraint optimization problems can be solved by using Lagrange multiplier method, which reduced the above constraint optimization problems to the following unconstrained ones:

$$J + \lambda \cdot \int (\dot{x}(t))^2 dt \rightarrow \min_{x(t)}$$

or

$$J + \lambda \cdot \int \left(\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right) dx dy \rightarrow \min_{I(x,y)}.$$

This idea is known as *Tikhonov regularization*.

From continuous to discrete signals and images. In practice, we can only observe a signal with a certain temporal resolution. As a result, in effect, we can only reconstruct the values $x_i = x(t_i)$ of the signal $x(t)$ at points $t_i = t_0 + i \cdot \Delta t$ from an appropriate grid.

Similarly, we only observe an image with a certain spatial resolution, so we can only reconstruct the values

$$I_{ij} = I(x_i, y_j)$$

on a certain grid $x_i = x_0 + i \cdot \Delta x$ and $y_j = y_0 + j \cdot \Delta y$.

In this discrete case, instead of the derivatives, we have differences:

$$J + \lambda \cdot \sum_i (\Delta x_i)^2 \rightarrow \min_{x_i}$$

or

$$J + \lambda \cdot \sum_i \sum_j ((\Delta_x I_{ij})^2 + (\Delta_y I_{ij})^2) \rightarrow \min_{I_{ij}}$$

where:

- $\Delta x_i \stackrel{\text{def}}{=} x_i - x_{i-1}$,
- $\Delta_x I_{ij} \stackrel{\text{def}}{=} I_{ij} - I_{i-1,j}$, and
- $\Delta_y I_{ij} \stackrel{\text{def}}{=} I_{ij} - I_{i,j-1}$.

Limitations of Tikhonov regularization. Tikhonov regularization is based on the assumption that the signal or the image is smooth. In real life, signals and images are, in general, not smooth; for example, many of them exhibit a fractal behavior; see, e.g., [5]. In such non-smooth situations, Tikhonov regularization does not work so well.

ℓ^p -methods as a heuristic idea to take non-smoothness into account. To take into account non-smoothness, researchers have proposed to modify the Tikhonov regularization formulas by using, instead of the squares of the derivatives, the p -th powers corresponding to some $p \neq 2$. In the resulting ℓ^p -approach, we solve the following minimization problems (see, e.g., [2], [4], [8]):

$$J + \lambda \cdot \sum_i |\Delta x_i|^p \rightarrow \min_{x_i}$$

or

$$J + \lambda \cdot \sum_i \sum_j (|\Delta_x I_{ij}|^p + |\Delta_y I_{ij}|^p) \rightarrow \min_{I_{ij}}.$$

These methods work much better than the original Tikhonov regularization [2], [4], [8].

Remaining problem. The remaining problem is that the ℓ^p -methods are heuristic, there is no convincing explanation of why necessarily we replace the square with a p -th power and not, for example, with some other function.

What we do in this paper. In this paper, we show that a natural formalization of the corresponding intuitive ideas indeed leads to ℓ^p -methods.

To formalize the intuitive ideas behind signal and image reconstruction, we use *fuzzy techniques*, a known way to transform imprecise intuitive ideas into exact formulas.

II. LET US APPLY FUZZY TECHNIQUES TO OUR PROBLEM Need to describe imprecise (“fuzzy”) expert knowledge.

In many areas of science and engineering, a large portion of expert knowledge is formulated by using imprecise words from natural language, such as “small”, “fast”, etc. Fuzzy logic is a technique designed to translate such knowledge into precise computer-understandable form; see, e.g., [3], [7], [10].

The main idea behind fuzzy logic is that to describe an imprecise notion like “small”, we assign, to each possible value x of the corresponding quantity, a degree $\mu(x)$ from the interval $[0, 1]$ to which this quantity satisfies the given property (e.g., to what extent x is small):

- the value $\mu(x) = 1$ means that the expert is absolutely sure that x is small;
- the value $\mu(x) = 0$ means that the expert is absolutely sure that x is not small, and
- values $\mu(x)$ strictly between 0 and 1 indicate that the expert is not fully confident that x is small.

Each value $\mu(x)$ can be obtained, e.g., by asking an expert to indicate, on a scale from 0 to 1, to what extent the value x is small.

“And”- and “or”-operations: a brief reminder. A large part of expert knowledge is formulated in terms of if-then rules. For example, we can have a rule like: “if the temperature is high and the humidity is low, then there is a high chance that the fertilizer may self-ignite”.

To adequately translate these rules into precise terms, we need to know the degree to which, for given temperature and given humidity, the condition of this rule is satisfied, i.e., to which extent it is true that the temperature is high *and* the humidity is low.

Ideally, we should be able to elicit these degrees from the expert, by asking the expert to what extent this condition is true for all possible combinations of temperature and humidity. However, this is often not practically possible, since there are many possible such combinations, and it is even less practical if the condition of a rule combines three or more statements.

To deal with such situations, it is necessary to be able, given the expert’s degrees of belief a and b in statements A and B , to estimate the expert’s degree of belief in a composite statement $A \& B$. The corresponding estimation algorithm $f_{\&}(a, b)$ is known as an “*and*”-operation, or *t-norm*.

The fact that this operation corresponds to “and” implies some of its natural properties. For example, since $A \& B$ means that same as $B \& A$, this operation should be commutative:

$$f_{\&}(a, b) = f_{\&}(b, a).$$

The fact that $A \& (B \& C)$ means the same as $(A \& B) \& C$ implies that the “and”-operation should be associative, i.e.,

$$f_{\&}(a, f_{\&}(b, c)) = f_{\&}(f_{\&}(a, b), c),$$

etc.

Similarly, to describe the expert’s degree of belief in statements of the type $A \vee B$ (“ A or B ”) based on his/her degrees of confidence a and b in individual statements A and B , we need to use “or”-operations (t-conorms) $f_{\vee}(a, b)$ which are also commutative and associative.

What we are trying to formalize. We are trying to formalize the statement that the signal or image is continuous, i.e., for signal, that all the differences $\Delta x_i = x_i - x_{i-1}$ are small. In other words, we want to say that the difference Δx_1 is small *and* that the difference Δx_2 is small *and* that the difference Δx_3 is small, etc.

Similarly, we want to say that the differences $\Delta_x I_{ij}$ and $\Delta_y I_{ij}$ between image intensities at nearby points are small.

Among all the signals (images) which are consistent with the observations, i.e., for which $J \leq c_0$ for some c_0 , it is then reasonable to select a signal for which the degree d with which the above “and”-statement is satisfied is the largest possible.

What we need to do to formalize this statement. According to the above-mentioned fuzzy logic techniques, to formalize this statement,

- we need to describe what is small, and
- we also need to select an appropriate “and”-operation.

Let $\mu(x)$ describe the degree to which x is small. Then the degree d to which the whole “and”-statement is satisfied is equal to

$$d = f_{\&}(\mu(\Delta x_1), \mu(\Delta x_2), \mu(\Delta x_3), \dots).$$

Selecting an “and”-operation. It is known (see, e.g., [6]) that each “and”-operation can be approximated, for any given accuracy $\varepsilon > 0$, by an *Archimedean* “and”-operation, i.e., by an “and”-operation of the type $f_{\&}(a, b) = f^{-1}(f(a)) \cdot f(b)$ for some increasing function $f(a)$ from $[0, 1]$ to $[0, 1]$, where f^{-1} denotes the inverse function.

Thus, without losing generality, we can safely assume that the actual “and”-operation has exactly this type.

The selection of an “and”-operation simplifies the corresponding optimization problem. For the above Archimedean “and”-operation, the above expression for d has the form

$$d = f^{-1}(f(\mu(\Delta x_1)) \cdot f(\mu(\Delta x_2)) \cdot f(\mu(\Delta x_3)) \cdot \dots).$$

Since the function $f(x)$ is increasing, maximizing d is equivalent to maximizing the value

$$f(d) = f(\mu(\Delta x_1)) \cdot f(\mu(\Delta x_2)) \cdot f(\mu(\Delta x_3)) \cdot \dots$$

Maximizing this product is equivalent to minimizing its negative logarithm

$$L \stackrel{\text{def}}{=} -\ln(d).$$

Since the logarithm of the product is equal to the sum of the logarithms, we get

$$L = -\sum_i \ln(f(\mu(\Delta x_i))),$$

i.e.,

$$L = \sum_i g(\Delta x_i),$$

where we denoted $g(x) \stackrel{\text{def}}{=} -\ln(f(\mu(x)))$.

In these terms, selecting a membership function is equivalent to selecting the related function $g(x)$.

Which function $g(x)$ should we select: idea. The value $\Delta x_i = 0$ is definitely small, so we should have $\mu(0) = 1$. Here, $f(1) = 1$, so $f(\mu(0)) = 1$ and thus, $g(0) = -\ln(1) = 0$.

The numerical value of a difference Δx_i depends on the choice of a measuring unit. If we choose a measuring unit which is a times smaller than the original one, then instead of the original numerical value Δx_i , we will have a different numerical value $a \cdot \Delta x_i$.

It is reasonable to request that the requirement

$$\sum_i g(\Delta x_i) \rightarrow \min$$

should not change if we simply change a measuring unit. For example, if for two pairs (z_1, z_2) and (z'_1, z'_2) , we have the

same value of the sum $\sum_i g(z_i)$, then this equality should remain true for all values $a > 0$. In precise terms, if we have

$$g(z_1) + g(z_2) = g(z'_1) + g(z'_2),$$

then we should have

$$g(a \cdot z_1) + g(a \cdot z_2) = g(a \cdot z'_1) + g(a \cdot z'_2).$$

Let us find the corresponding function $g(x)$. Let us consider the case when:

- z'_1 is close to z_1 , i.e., when $z'_1 = z_1 + \Delta z$ for a small value Δz , and
- z'_2 is close to z_2 , i.e.,

$$z'_2 = z_2 + k \cdot \Delta z + o(\Delta z)$$

for an appropriate k .

Substituting these values z'_1 and z'_2 into the above equality, we get

$$g(z_1) + g(z_2) = g(z_1 + \Delta z) + g(z_2 + k \cdot \Delta z).$$

Here,

$$g(z_1 + \Delta z) = g(z_1) + g'(z_1) \cdot \Delta z + o(\Delta z)$$

and

$$g(z_2 + k \cdot \Delta z) = g(z_2) + g'(z_2) \cdot k \cdot \Delta z + o(\Delta z),$$

so the above equality implies that

$$g'(z_1) \cdot \Delta z + g'(z_2) \cdot k \cdot \Delta z + o(\Delta z) = 0.$$

Diving both sides by Δz and taking $\Delta z \rightarrow 0$, we get

$$g'(z_1) + g'(z_2) \cdot k = 0,$$

hence

$$k = -\frac{g'(z_1)}{g'(z_2)}.$$

The condition

$$g(a \cdot z_1) + g(a \cdot z_2) = g(a \cdot z'_1) + g(a \cdot z'_2)$$

similarly takes the form

$$g'(a \cdot z_1) + g'(a \cdot z_2) \cdot k = 0,$$

i.e.,

$$g'(a \cdot z_1) - g'(a \cdot z_2) \cdot \frac{g'(z_1)}{g'(z_2)} = 0.$$

Thus,

$$g'(a \cdot z_1) = g'(a \cdot z_2) \cdot \frac{g'(z_1)}{g'(z_2)}.$$

By moving all the terms related to z_1 to the left-hand side and all other terms to the right-hand side, we get

$$\frac{g'(a \cdot z_1)}{g'(z_1)} = \frac{g'(a \cdot z_2)}{g'(z_2)}$$

for all a , z_1 , and z_2 .

This means that the ratio $\frac{g'(a \cdot z_1)}{g'(z_1)}$ does not depend on z_i , it only depends on a :

$$\frac{g'(a \cdot z_1)}{g'(z_1)} = F(a)$$

for some function $F(a)$.

For $a = a_1 \cdot a_2$, we have

$$F(a) = \frac{g'(a \cdot z_1)}{g'(z_1)} = \frac{g'(a_1 \cdot a_2 \cdot z_1)}{g'(z_1)} =$$

$$\frac{g'(a_1 \cdot (a_2 \cdot z_1))}{g'(a_2 \cdot z_1)} \cdot \frac{g'(a_2 \cdot z_1)}{g'(z_1)} = F(a_1) \cdot F(a_2),$$

i.e.,

$$F(a_1 \cdot a_2) = F(a_1) \cdot F(a_2).$$

It is known (see, e.g., [1]) that every continuous function satisfying this property has the form $F(a) = a^q$ for some real number q .

The condition $\frac{g'(a \cdot z_1)}{g'(z_1)} = F(a)$ now takes the form

$$g'(a \cdot z_1) = g'(z_1) \cdot F(a) = g'(z_1) \cdot a^q.$$

In particular, for $z_1 = 1$, we get

$$g'(a) = C \cdot a^q,$$

where $C \stackrel{\text{def}}{=} g'(1)$.

We have an expression for the derivative $g'(a)$ of the desired function $g(a)$. To get $g(a)$, we therefore need to integrate this derivative. For this integration, we have two different formulas: for $q = -1$ and for all other q .

Let us show that the value $q = -1$ is impossible. Indeed, if $q = -1$, we get $g(a) = C \cdot \ln(a) + \text{const}$, which contradicts to the above requirement that $g(0) = 0$.

Thus, we have $q \neq -1$. Therefore, integration leads to

$$g(a) = \frac{C}{q+1} \cdot a^{q+1} + \text{const}.$$

Conclusion: we have indeed justified the ℓ^p -method. For the above function $g(x)$, we have

$$\sum_i g(\Delta x_i) = \frac{C}{q+1} \cdot \sum_i |\Delta x_i|^{q+1} + \text{const}.$$

Minimizing this sum is equivalent to minimizing the sum

$$\sum_i |\Delta x_i|^{q+1}.$$

According to the Lagrange multiplier method, minimizing this sum under the constraint $J \leq c$ is equivalent to minimizing the expression

$$J + \lambda \cdot \sum_i |\Delta x_i|^p,$$

for $p = q + 1$. Thus, for signals, we have indeed justified the ℓ^p -method.

Similar arguments explain the ℓ^p -method for images.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award “UTEP and Prudential Actuarial Science Academy and Pipeline Initiative” from Prudential Foundation.

REFERENCES

- [1] J. Aczél and J. Dhombres, *Functional Equations in Several Variables*, Cambridge University Press, 2008.
- [2] B. Amizic, L. Spinoulas, R. Molina, and A. K. Katsaggelos, “Compressive blind image deconvolution”, *IEEE Transactions on Image Processing*, 2013, Vol. 22, No. 10, pp. 3994–4006.
- [3] G. Klir and B. Yuan, “Fuzzy Sets and Fuzzy Logic”, Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [4] D. Krishnan and R. Fergus, “Fast image deconvolution using hyper-Laplacian priors”, In: Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (eds.), *Proceedings of the 23th Annual Conference on Neural Information Processing Systems NIPS'2009*, Vancouver, Canada, December 7–12, 2009.
- [5] B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, San Francisco, California, 1983.
- [6] H. T. Nguyen, V. Kreinovich, and P. Wojciechowski, “Strict Archimedean t-Norms and t-Conorms as Universal Approximators”, *International Journal of Approximate Reasoning*, 1998, Vol. 18, Nos. 3–4, pp. 239–249.
- [7] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2006.
- [8] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Verlag, New York, 2006.
- [9] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, V. H. Winston & Sons, Washington, DC, 1977.
- [10] L. A. Zadeh, “Fuzzy sets”, *Information and Control*, 1965, Vol. 8, pp. 338–353.