

# Calibrating Parameters of Cost Functionals

Laurent Younes

CMLA (CNRS, URA 1611)  
Ecole Normale Supérieure de Cachan  
61, avenue du Président Wilson  
F-94 235 Cachan CEDEX  
email: younes@cmla.ens-cachan.fr  
Tel: 33.1.47.40.59.18, Fax: 33.1.47.40.59.01

**Abstract.** We propose a new framework for calibrating parameters of energy functionals, as used in image analysis. The method learns parameters from a family of correct examples, and given a probabilistic construct for generating wrong examples from correct ones. We introduce a measure of frustration to penalize cases in which wrong responses are preferred to correct ones, and we design a stochastic gradient algorithm which converges to parameters which minimize this measure of frustration. We also present a first set of experiments in this context, and introduce extensions to deal with data-dependent energies.

**keywords:** Learning, variational method, parameter estimation, image reconstruction, Bayesian image models

## 1 Description of the Method

Many problems in computer vision are addressed through the minimization of a cost functional  $U$ . This function is typically defined on a large, finite, set  $\Omega$  (for example the set of pictures with fixed dimensions), and the minimizer of  $x \mapsto U(x)$  is supposed to conciliate several properties which are generally antithetic.

Indeed, the energy is usually designed as a combination of several terms, each of them corresponding to a precise property which must be satisfied by the optimal solution. As an example among many others, let us quote probably the most studied cost functional in computer vision, namely the Mumford/Shah energy (cf. [5]), which is used to segment and smooth an observed picture. Expressed in a continuous setting, it is the combination of three terms, one which ensures that the smoothed picture  $x$ , defined on a set  $D \subset \mathbb{R}^2$  is not too different from the observed one  $\xi$ , another which states that the derivative of the smoothed picture is small, except, possibly, on a discontinuity set  $\Delta$ , and a last one which ensures that the discontinuity set has small length. These terms are weighted by parameters, yielding an energy function of the kind

$$U(x) = \int_D (\xi(s) - x(s))^2 ds + \alpha \int_{D \setminus \Delta} |\nabla_s x| * 2 ds + \beta \mathcal{H}(\Delta) \quad (1)$$

where  $\mathcal{H}(\Delta)$  is Hausdorff measure of the discontinuity set.

In this paper, we consider cost functionals of the kind

$$U(x) = U_0(x) + \sum_{i=1}^d \theta_i U_i(x)$$

where the  $\theta_i$  are positive parameters. Whatever vision task this functional is dedicated to (restoration, segmentation, edge detection, matching, pattern recognition, ...), it is acknowledged that variations in the values of the parameters have significant effects on the qualitative properties of the minimizer. Very often, these parameters are fixed by trial and error, while experimenting the optimization algorithm. We here propose a systematic way for tuning them, based on a learning procedure.

The method is reminiscent to the qualitative box estimation procedure which has been introduced by Azencott in [1]. It relies on some *a priori* knowledge which is available to the designer. The basic information can be expressed under the statement: *For some configurations  $x$  and  $y$  in  $\Omega$ , one should have  $U(x) \geq U(y)$ .* In other terms,  $y$  is a “better” solution than  $x$ .

When this is known for a number of pairs of configurations,  $\{(x_k, y_k), k = 1, \dots, N\}$ , we get a system of constraints which take the form, for  $k = 1, \dots, N$ :

$$U_0(y_k) - U_0(x_k) + \sum_{i=1}^d \theta_i (U_i(y_k) - U_i(x_k)) \leq 0$$

If we let  $\theta = (\theta_1, \dots, \theta_d)$ ,  $\Delta_{ki} = U_i(y_k) - U_i(x_k)$ , and  $\Delta_k = (\Delta_{k1}, \dots, \Delta_{kd})$ , this can be written

$$\Delta_{0k} + \langle \theta, \Delta_k \rangle \leq 0, k = 1, \dots, N,$$

$\langle \cdot, \cdot \rangle$  being the usual inner product on  $\mathbb{R}^d$ .

Solving such a system of linear inequalities can be performed by a standard simplex algorithm. However, when the system has no solution (which is likely to occur if there are many inequalities, and/or if they are deduced for the observation of noisy real data), it is difficult to infer from the simplex method which parameter should be selected. We thus define a new cost functional in the parameters, or *measure of frustration*, which is large when the inequalities are not satisfied: denote by  $\alpha^+$  the positive part of a real number  $\alpha$ , and set

$$F_0(\theta) = \sum_{k=1}^N [\Delta_{0k} + \langle \theta, \Delta_k \rangle]^+$$

It is practically more convenient to use a smooth approximation of this function, so that we let, for  $\lambda > 0$

$$F_\lambda(\theta) = \frac{1}{2} \sum_{k=1}^N q_\lambda[\Delta_{0k} + \langle \theta, \Delta_k \rangle]$$

with  $q_\lambda(\alpha) = \lambda \log(e^{\frac{\alpha}{\lambda}} + e^{-\frac{\alpha}{\lambda}}) + \alpha$ . Given properly selected examples, the minimization of  $F_\lambda$  is the core of our estimation procedure. We therefore study some related properties.

## 2 Properties of the Function $F_\lambda$

**Proposition 1.** *For all  $\lambda \geq 0$ ,  $F_\lambda$  is a convex function of  $\theta$ . Moreover,  $\lim_{\lambda \rightarrow 0^+} F_\lambda(\theta) = F_0(\theta)$ .*

This is more or less obvious and left to the reader. Let us, however, write down the derivatives of  $F_\lambda$ , for  $\lambda > 0$ , since they will be used in the sequel (recall that the first derivative is a vector and the second derivative a  $d \times d$  symmetric matrix). One has:

$$F'_\lambda(\theta) = \sum_{k=1}^N \left( 1 + \tanh \left[ \frac{1}{\lambda} (\Delta_{0k} + \langle \theta, \Delta_k \rangle) \right] \right) \Delta_k \tag{2}$$

$$F''_\lambda(\theta) = \frac{1}{\lambda} \sum_{k=1}^N \left( 1 - \tanh^2 \left[ \frac{1}{\lambda} (\Delta_{0k} + \langle \theta, \Delta_k \rangle) \right] \right) \Delta_k \cdot {}^t \Delta_k \tag{3}$$

Denote by  $\Sigma_\Delta$  the covariance matrix of the  $\Delta_k$ , namely  $\Sigma_\Delta = \sum_{k=1}^N \Delta_k \cdot {}^t \Delta_k$ .

**Proposition 2.** *The matrix  $\Sigma_\Delta$  is positive definite if and only if, for all  $\lambda > 0$ , the function  $F_\lambda$  is strictly convex, and if and only if, for some  $\lambda > 0$ , the function  $F_\lambda$  is strictly convex*

*Proof.* If, for some  $\lambda > 0$ , and for some  $\theta$ ,  $F''_\lambda(\theta)$  is not definite positive, there exists a vector  $u \in \mathbb{R}^d$  such that  ${}^t u \cdot F''_\lambda(\theta) \cdot u = 0$ . But one has

$${}^t u \cdot F''_\lambda(\theta) \cdot u = \frac{1}{\lambda} \sum_{k=1}^N \left( 1 - \tanh^2 \left[ \frac{1}{\lambda} (\Delta_{0k} + \langle \theta, \Delta_k \rangle) \right] \right) \langle u, \Delta_k \rangle^2$$

and this expression can vanish only if, for all  $k$ ,  $\langle u, \Delta_k \rangle = 0$ , but this implies that  ${}^t u \Sigma_\Delta u = 0$  so that  $\Sigma_\Delta$  cannot be definite.

Conversely, if  $\Sigma_\Delta$  is not positive, one shows similarly that there exists  $u$  such that  $\langle u, \Delta_k \rangle = 0$  for all  $k$ , but this implies that, for any  $\lambda > 0$ , for any  $\theta$  and any  $t \in \mathbb{R}$ ,  $F_\lambda(\theta + tu) = F_\lambda(\theta)$  so that  $F_\lambda$  cannot be strictly convex.

Thus, non convexity is equivalent to the existence of a fixed linear relation among  $\Delta_{k1}, \dots, \Delta_{kd}$ .

We now address the question of the existence of a minimum of  $F_\lambda$ . We assume  $\lambda > 0$  and strict convexity, ie  $\Sigma_\Delta > 0$ . The convex function  $F_\lambda$  has no minimum if and only if it has a direction of recession, ie. if and only if there exists a vector  $u \in \mathbb{R}^d$  such that, for all  $\theta$ ,  $t \mapsto F_\lambda(\theta + tu)$  is decreasing. By studying the derivative of this function, we can show that, in order to have a direction of recession, there must exist some  $u$  such that  $\langle \Delta_k, u \rangle \leq 0$  for all  $k$ , with a strict inequality for some  $k$  in order to have strict convexity. If  $u$  provides a direction of recession, then  $t \cdot u$  will be a solution of the original set of inequalities as soon as  $t$  is large enough. This is a very inconvenient feature, since, in particular, it

will completely cancel out the role of  $U_0$ . Such a situation is in fact caused a lack of information in the original set of examples  $(x_k, y_k), k = 1, \dots, N$ , in the sense that this set fails to provide situations in which the role of  $U_0$  has some impact.

### 3 Learning from Examples

#### 3.1 Objective Function from Small Variations

We now provide a framework in which this simple technique can be applied when some examples of “correct configurations” are available. They may come, either from simulated, synthetic data, or from real data which have been processed by an expert. The idea is to generate random perturbations of the correct configurations and to estimate the parameters so that the perturbed configurations have a higher energy than the correct ones.

Let us first assume, that a single configuration  $y_0$  is provided. Our goal is thus to design the parameters so that  $y_0$  will be, in some local sense, a minimizer of the energy. The key of the learning process is to define a process which generates *random perturbations* of a given configuration. This process of course depends on the application, and should provide a sufficiently large range of new configurations from the initial one. Formally, it will be associated to a transition probability  $P(y_0, \cdot)$  on  $\Omega$ , which will produce variations of the correct configuration  $y_0$ . Assume this is done  $K$  times independently, and that a sample  $x_1, \dots, x_K$  has been drawn from this probability. From the fact that  $y_0$  is a good configuration, we assume that, for all  $k, U(y_0) - U(x_k) \leq 0$ . Slightly changing the notation, define  $\Delta(y_0, x)$  to be the vector composed with the  $U_i(y_0) - U_i(x)$  for  $i = 1, \dots, d$  and  $h(y_0, x) = U_0(y_0) - U_0(x)$ . The previous method leads to minimize

$$F_\lambda^K(\theta) = \frac{1}{2} \sum_{k=1}^K q_\lambda[h(y_0, x_k) + \langle \theta, \Delta(y_0, x_k) \rangle]$$

Now, when  $K$  tends to infinity, the limit of  $F_\lambda^K/K$  is almost surely given by (since the samples are drawn independently)

$$F_\lambda(\theta) = \frac{1}{2} \mathbf{E}_{y_0} \{q_\lambda[h(y_0, x_k) + \langle \theta, \Delta(y_0, x_k) \rangle]\}$$

where  $\mathbf{E}_{y_0}$  is the expectation with respect to the probability  $P(y_0, \cdot)$ . This functional becomes our measure of frustration, which should be minimized in order to calibrate  $\theta$ .

Assume now that several examples are provided, under the form of a learning set  $y_1, \dots, y_N$ : the new objective function is

$$F_\lambda(\theta) = \frac{1}{2} \sum_{j=1}^N \mathbf{E}_{y_j} \{q_\lambda[h(y_j, x_k) + \langle \theta, \Delta(y_j, x_k) \rangle]\}$$

### 3.2 Minimizing $F_\lambda$

To simplify the notation, we restrict again to the case of a single example  $y_0$ . We still have the fact that, for any  $\lambda$ , the function  $F_\lambda$  is convex, with first derivative

$$F'_\lambda(\theta) = \mathbf{E}_{y_0} \left\{ \left( 1 + \tanh \left[ \frac{1}{\lambda} (h(y_0, \cdot) + \langle \theta, \Delta(y_0, \cdot) \rangle) \right] \right) \Delta(y_0, \cdot) \right\} \quad (4)$$

According to the discussion of section 2, the transition probability  $P$  should, to avoid directions of recession, explore a sufficiently large neighborhood of  $y_0$ , to provide enough information on the variations of  $U$ . Because of this, it is likely that the gradient in (4) cannot be efficiently computed, neither analytically nor numerically. To minimize  $F_\lambda$  in such a case, we use a stochastic gradient learning procedure, which we describe now:

### Learning Procedure

0. Start with some initial value  $\theta_0$

1. At time  $n$ ,  $\theta_n$  being the current parameter, draw at random a sample  $X^n$  from the transition probability  $P(y_0, \cdot)$ , and set

$$\theta_{n+1} = \theta_n - \gamma_{n+1} \left( 1 + \tanh \left[ \frac{1}{\lambda} (h(y_0, X^n) + \langle \theta, \Delta(y_0, X^n) \rangle) \right] \right) \Delta(y_0, X^n) \quad (5)$$

where  $(\gamma_n, n \geq 1)$  is a decreasing sequence of positive gains satisfying  $\sum_n \gamma_n = +\infty$  and  $\sum_n \gamma_n^2 < +\infty$ .

Standard results in stochastic approximation (see [2], for example), show that, in the absence of direction of recession, the sequence  $(\theta_n)$  generated by this algorithm almost surely converges to the minimizer of  $F_\lambda$ .

If there are more than one example  $y_1, \dots, y_M$ , the previous algorithm simply has to be modified by taking, at each step,  $y_0$  at random in the set  $\{y_1, \dots, y_M\}$ .

### 3.3 Remark

Notice that, under its most general form, and when the perturbations explore a large set of configurations, there is very little chance that there exists a parameter set for which all the constraints are truly satisfied, that is for which the energy of the correct configurations  $y_j$  are smaller than the energies of all the perturbations which might be generated by  $P(y_j, \cdot)$ . This could be made possible by designing an energy with a very large number of terms, which will then essentially work as an associative memory (like an Hopfield neural net [4]), in which the correct configurations are *stored*, but this certainly is not a desirable feature of an energy function in image processing. A more efficient goal is to learn some common important trends of the correct configurations, and not all their peculiarities, in which case having some residual frustration is not a problem.

## 4 Illustration

### 4.1 Description

We illustrate this methodology with binary example. Let  $\Omega$  be the set of configurations  $x = (x_s, s \in S := \{1, \dots, M\}^2)$  with  $x_s = 0$  or 1 for all  $s$ . We define an energy  $U(x)$  on  $\Omega$  as follows.

Let  $U_0(x) = \sum_s x_s$ . For a radius  $r > 0$  and a direction  $\alpha \in [0, 2\pi[$ , we define an energy term  $U_{r,\alpha}$  which operates as an edge analyzer in the direction  $\alpha$ , with scale  $r$ .

For  $s = (i, j) \in S$ , let  $\mathcal{B}_s(r)$  be the discrete ball of center  $s$  and radius  $r$ , ie. set of all  $s' = (i', j') \in S$  such that  $(i - i')^2 + (j - j')^2 \leq r^2$ . For each direction  $\alpha$ , divide this ball in two parts  $\mathcal{B}_s^+(r, \alpha)$  and  $\mathcal{B}_s^-(r, \alpha)$  according to the sign of  $(i - i') \cos \alpha + (j - j') \sin \alpha$ , then define

$$U_{r,\alpha}(x) = \sum_{s \in S} \left| \sum_{s' \in \mathcal{B}_s^+(r, \alpha)} x_{s'} - \sum_{s' \in \mathcal{B}_s^-(r, \alpha)} x_{s'} \right|$$

Finally, select a series of pairs  $(r_i, \alpha_i)$  for  $i = 1, \dots, d$ , and set

$$U(x) = \theta_0 U_0(x) + \sum_{i=1}^d \theta_i U_{r_i, \alpha_i}(x)$$

Our experiment will consist in learning the parameters  $\theta_0, \dots, \theta_d$  on the basis of a single image  $y_0$ , and then try to analyze which features of the image have emerged in the final model. Notice that we have added a parameter,  $\theta_0$ , for the first term  $U_0$ , which is also estimated. If there exist parameters such that  $U(x) > U(y_0)$  for all configurations  $x$  which can be generated by  $P(y_0, \cdot)$ , the extraneous parameter is redundant (only its sign matters), and this creates a direction of recession for the minimized functional. But such a case did not seem to happen in the present set of experiments, so that, even with one additional parameter, the measure of frustration did remain strictly convex.

For learning, the perturbations  $P(y_0, \cdot)$  consist in adding or deleting balls of random centers and radii to the configuration  $y_0$ . To validate the estimated parameters, we run an energy minimization algorithm (simulated annealing with exponentially fast decay of temperature) with different starting configurations (including the learned image  $y_0$  itself) to see whether  $y_0$  is close to the minimizing solutions.

### 4.2 Experiments

We have used three pictures (disc, square and triangle, see fig. 1), and estimated parameters independently for each picture. The results were quite different for each image.

The disc-picture seems to have been perfectly stored, in the sense of an associative memory, by the learned parameters: starting with any initial picture, the

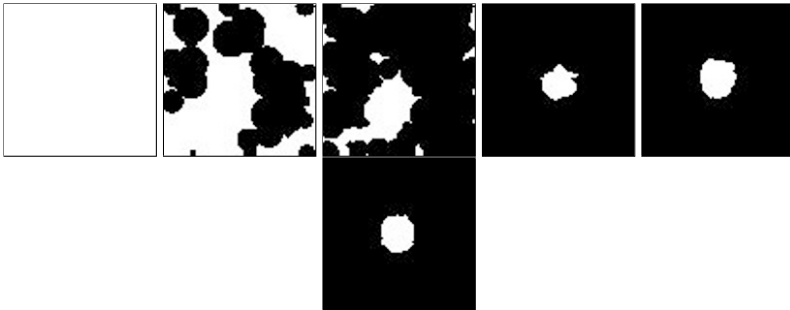
final restored picture is a disc, with only minor variations. This is not surprising, in fact, since the energy function is itself based on disc-shaped analyzers.

The square picture is stabilized by the restoration algorithm, again with minor variations, so that the estimation has succeeded in making this picture (almost) a local minimum of the energy. However, starting from other configurations does not always result in a white square on a dark background, and a phenomenon reminiscent of phase transition can be observed (see fig. 5). This is due to the fact that, in the square picture, the number of white pixels is almost equal to the number of black pixels.

Finally, the triangle picture is not even stabilized by the restoration algorithm. It is in fact significantly modified, as shown in fig. 6. As stated before, it would not be difficult to design an energy with additional terms in order to perfectly store the triangle. It is however more interesting to stay with a given energy, and analyse which features for the triangle picture have been learned. This can be seen in fig. 7, where the restored picture from a uniformly white input clearly has nothing to do with a triangle, but shares essential local features, in particular regarding the orientations of the boundaries.



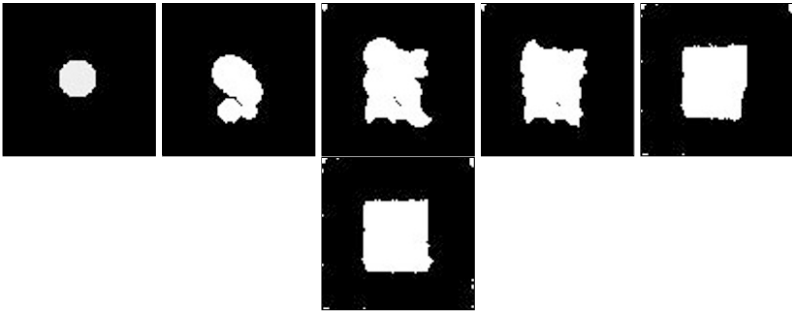
**Fig. 1.** Pictures of disc, square, triangle



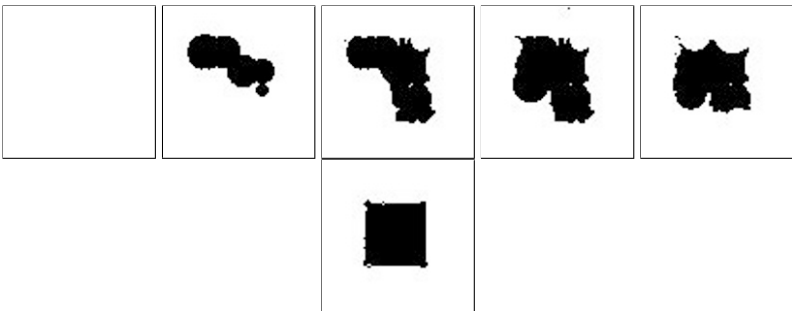
**Fig. 2.** Starting with a white picture with parameters estimated from the disc



**Fig. 3.** Starting with a black picture with parameters estimated from the disc



**Fig. 4.** Starting from the disc with parameters estimated from the square

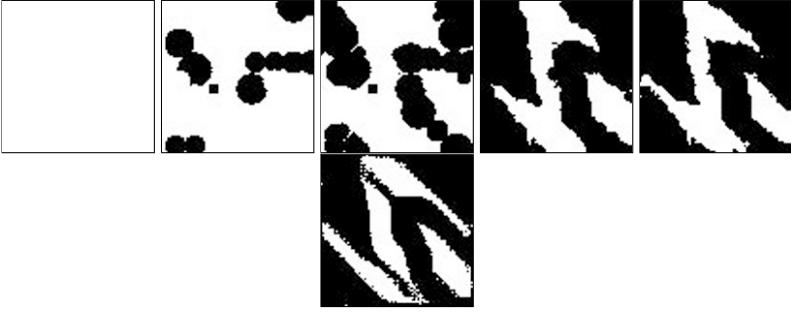


**Fig. 5.** Starting with a white picture with parameters estimated from the square, exhibiting a phase-transition-like phenomenon



**Fig. 6.** Output of the restoration algorithm, initialized with the triangle, and using parameters estimated from the triangle





**Fig. 7.** Starting with the white picture with parameters estimated from the triangle

## 5 Extension to Data-Dependent Cost Functions

### 5.1 Generalities

In a typical use of energy minimization methods for image analysis, one (or several terms) in the energy depends on an extraneous configuration of observed data  $\xi$ , like the first term in equation (1). Such situations directly arise from the Bayesian framework which has been introduced in [3], and applied many times since then.

In this case, the calibrated parameters should be able to adapt to variations of the data, and  $\theta_1, \dots, \theta_d$  should be functions of  $\xi$ . One simple way to address this is to model each  $\theta_i$  as a linear combination of some fixed functions of  $\xi$ , as in regression analysis:

$$\theta_i = \sum_{j=1}^K \beta_{ij} \Phi_j(\xi)$$

The functions  $\Phi_j$  are fixed in the learning procedure. They should be relevant statistics of the data, for the given application. From a formal point of view, we are back to the framework of section 3.2, with the new energy terms

$$\tilde{U}_{ij}(\xi, y) = \Phi_j(\xi) U_i(y)$$

and parameters  $\beta_{ij}$ . However, in this case, it is clear that learning can only be performed on the basis of sufficiently large number of correct analyses, of the kind  $(\xi_1, y_1), \dots, (\xi_N, y_N)$ , since we are going to estimate functions of the variable  $\xi$ .

An alternative to choosing fixed functions  $\Phi_j$  is to set  $\Phi_j = \Phi(h_j + \langle W_j, \xi \rangle)$  where  $h_j \in \mathbb{R}$  and  $W_j$  is a vector of same dimension as  $\xi$ , which also have to be estimated. Here  $\Phi$  is a fixed function, typically sigmoidal. It is not hard to adapt the stochastic gradient descent algorithm to deal with this model, which will have more learning power than the initial linear combinations. The counterpart of this is that the measure of frustration is not convex anymore.

We now illustrate this approach by considering a simple unidimensional framework.

## 5.2 A 1D Example

We consider the issue of smoothing a function  $\xi : [0, 1] \mapsto \mathbb{R}$ . Fixing a discretization step  $\delta = 1/M$ , we let  $\xi_k = \xi(k\delta)$  and consider the cost function

$$U(\xi, x) = \sum_{k=1}^N (\xi_k - x_k)^2 + \lambda \sum_{i=2}^N (x_k - x_{k-1})^2$$

where  $x$  is the unknown smooth signal.

To calibrate the parameters, we let  $(\Phi_1(\xi), \dots, \Phi_p(\xi))$  be regularly spaced quantiles of the distribution of  $(\xi_k - \xi_{k-1})$  and look for  $\lambda$  in the form

$$\lambda = \sum_{i=1}^p \lambda_i \Phi_i(\gamma)$$

The learning dataset is generated by first simulating the smooth signal  $x$  by random linear combinations of cosine functions on  $[0, 1]$ :

$$x(t) = \sum_{p=1}^K \alpha_p \cos(\omega_p t + \phi_p)$$

where the  $\alpha_p$ ,  $\omega_p$  and  $\phi_p$  are random;  $\xi$  is obtained from  $x$  by adding a gaussian white noise of random variance  $\sigma^2$ . The random perturbations in the learning procedure consisted in adding a small variation to one or several  $x_k$ 's.

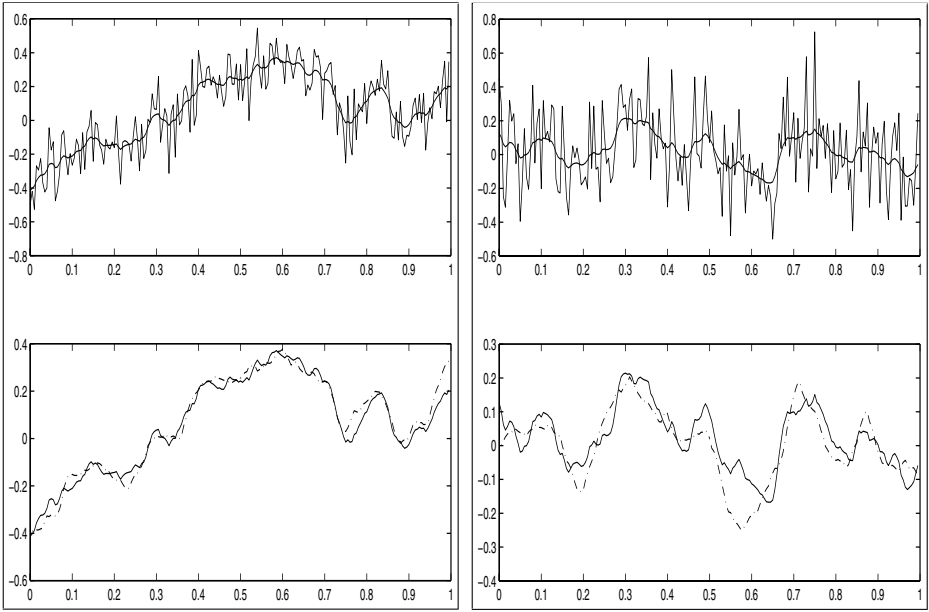
The learning procedure achieved the estimation of  $\lambda$  as a linear function of the distribution of the  $\xi_i - \xi_{i-1}$ . It is an odd function of the quantiles, which implies that it is not affected if a constant value is added to  $\xi_i - \xi_{i-1}$  (ie. a linear term added to  $\xi_i$ ). It can be very tightly approximated by the polynomial  $250q^7 + 3.1 * q$ , which means that  $\sum_q \lambda(q)$  is a linear combination of the 8th centered moment and the variance of the  $\xi_i - \xi_{i-1}$ .

The cost function  $U$  has been minimized on test data generated independently, and some results are shown in fig. 8.

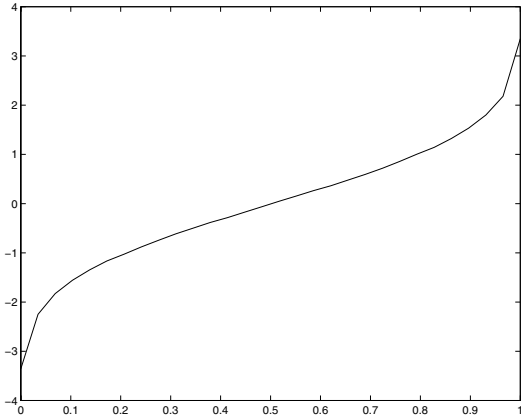
## 6 Conclusion

In this paper, we have developed a new learning framework for calibrating parameters of energy functionals, as used in image analysis. Given a probabilistic way for building wrong examples from correct ones, we have introduced a stochastic gradient algorithm which consistently estimates parameters, in order to minimize a measure of frustration designed to wrong examples to have a larger energy than correct ones. An extension of the method in the case of data-dependent energies have been proposed, resulting in an adaptive set of parameters reacting to the statistical distribution of the data. The approach has been illustrated by a preliminar series of experiments.

We are now aiming at developping this approach to deal with realistic imaging problems. We are, in particular, studying image segmentation energies, and developing 2D perturbations to learn parameters.



**Fig. 8.** Smoothing 1D data. Left and right: two distinct examples; up: observed and estimated signals; down: true and estimated signals.



**Fig. 9.** Plot of the  $\lambda_i$  vs. the quantiles

## References

1. R. AZENCOTT, *Image analysis and markov fields*, in Proc. of the Int. Conf. on Ind. and Appl. Math, SIAM, Paris, 1987.
2. A. BENVENISTE, M. MÉTIVIER, AND P. PRIOURET, *Algorithmes Adaptatifs et Approximations Stochastiques, Théorie et Application*, Masson, 1987.
3. S. GEMAN AND D. GEMAN, *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images*, IEEE Trans. PAMI, 6 (1984), pp. 721–741.
4. J. J. HOPFIELD, *Neural networks and physical systems with emergent collective computational abilities*, Proc. Nat. Acad. Sci. USA, 79 (1982), pp. 2554–2558. Biophysics.
5. D. MUMFORD AND SHAH, *Optimal approximation by piecewise smooth functions and variational problems*, Comm. Pure and Appl. Math., XLII (1988).