ECSE-626

Statistical Computer Vision

Regularization in Vision

Regularization in Vision

- We will now consider several inverse problems in vision and assess their ill-posedness.
- For several ill-posed vision problems, we will examine the application of the regularization framework outlined for their solution.

In order to solve an inverse problem using regularization we need to know the following information:

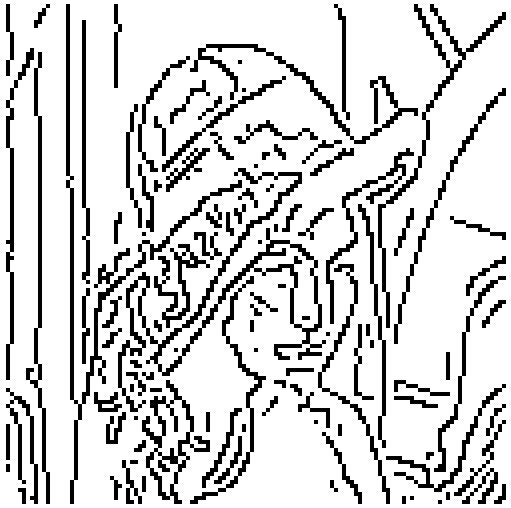
- The data space $\boldsymbol{U}$ and the solution space $\boldsymbol{F}$
- The metrics for the data and solution spaces
- The operator $\boldsymbol{A}$
- The form of the stabilizer $\boldsymbol{\Omega}$
- The value of the regularization parameter $\boldsymbol{\alpha}$
- The data value, $\boldsymbol{u}$

Edge Detection

Let us take as an example that of regularizing the process of ***edge detection***.

Edge Detection

Find edges in the scene based on intensity changes in the image.



Edge Detection

There is no *natural* definition of an edge - it is an abstract quantity - but it is often defined as the location in an image where the magnitude of the gradient is maximum.

So, an important part of an edge detection algorithm is the computation of the gradient.

Edge Detection

For simplicity, let us look at the 1-D case (e.g. look for edges along horizontal lines in an image).

The computation of the derivative can be thought of as solving the inverse problem associated with the following forward operator:

$$u(x) = \int_{-\infty}^{x} z(s)ds$$

Edge Detection

Is this problem well-posed?

Let's first examine whether the solution depends continuously on the data…

Edge Detection

Suppose the data u is perturbed slightly by measurement noise to produce

$$u(x) = u_T(x) + \delta \sin(kx),$$

where $u_T(x) = \int_{-\infty}^{x} z_T(s)ds$

Edge Detection

It can be seen that the solution to the inverse problem of finding $z(s)$ where $u(x) = \int_{-\infty}^{x} z(s)ds$ is given by

$$z(s) = z_T(s) + k\delta\cos(ks)$$

Edge Detection

The perturbation in the data is bounded,

$$\rho_U(u(x) - u_T(x)) = \delta$$

But the perturbation in the solution is

$$\rho_F(z(s) - z_T(s)) = k\delta$$

No matter how small δ is (i.e. how small the perturbation in the data is) we can always find a finite value of k that makes the perturbation in the solution arbitrarily large.

Edge Detection

Differentiation "amplifies high frequency noise.

Thus, the solution is unstable, and the inverse problem is ill-posed.

Regularization

Let us apply the Tikhonov regularization theory to the handling of the ill-posed problem of differentiation.

Let us assume that the solution space and the data space are both $L_2(\mathbf{R})$ (the space of square-integrable functions).

A suitable metric for these spaces is therefore:

$$\rho(f_1, f_2) = \int_{-\infty}^{\infty} (f_1(x) - f_2(x))^2 \, dx$$

Regularization

As note earlier the forward operator for the differentiation problem is:

$$u(x) = Az = \int_{-\infty}^{x} z(s)ds$$

Now, we need to specify a suitable stabilizing functional.

Regularization

Tikhonov proposed a specific stabilizer which is commonly used in computer vision applications, and which has come to be known as the Tikhonov stabilizer.

Regularization

The Tikhonov stabilizer is defined over the space F_1 which is the Sobolev space $W_2^p(\mathbf{R})$, the space of functions that have square integrable derivatives up to p^{th} order.

$$\Omega[z(x)] = \int_{-\infty}^{\infty} \sum_{r=0}^{p} q_r \left(\frac{d^r z(x)}{dx^r} \right)^2 dx$$

where $q_r \geq 0$ for $0 \leq r < p$ and $q_p > 0$.

Regularization

For example, we could take $p = 2,\ q_0 = q_1 = 0,\ q_2 = 1$:

$$\Omega[z(x)] = \int_{-\infty}^{\infty} \left(\frac{d^2 z(x)}{dx^2} \right)^2 dx$$

Regularization

The functional to minimize to get the solution is then:

$$M^{\alpha}[z] = \alpha \rho_U (Az, u) + \Omega[z]$$

Regularization

The functional to minimize to get the solution is then:

$$M^{\alpha}[z] = \alpha \rho_U (Az, u) + \Omega[z]$$

$$= \alpha \int_{-\infty}^{\infty} \left(u(x) - \int_{-\infty}^{x} z(s) ds \right)^2 dx + \int_{-\infty}^{\infty} \left(\frac{d^2 z(x)}{dx^2} \right)^2 dx$$

Regularization

For historical reasons, in most computer vision applications of regularization, the regularization parameter is placed in front of the stabilizing functional:

$$M^{\alpha}[z] = \rho_U(Az, u) + \frac{1}{\alpha}\Omega[z]$$

or, more usually:

$$M^{\lambda}[z] = \rho_U(Az, u) + \lambda\Omega[z]$$

Regularization

The integral might be difficult or time-consuming to compute, so in Computer Vision applications, a slightly different approach is often used:

The data are interpolated or approximated by an analytical function.
Subsequently, one can compute the analytical derivative of this function.

Regularization

Thus we define $v(x) = \int_{-\infty}^{x} z(s)ds.$

This is then differentiated to obtain $z(x) = dv/dx.$

The functional to minimize to get the solution is then:

$$M^{\lambda}[z] = \rho_U(v,u) + \lambda\Omega[v]$$

$$= \int_{-\infty}^{\infty} \left(u(x) - v(x)\right)^2 dx + \lambda \int_{-\infty}^{\infty} \left(\frac{d^2 v(x)}{dx^2} \right)^2 dx$$

Regularization

This may seem like a silly thing to do, since it appears that all we are doing is replacing our solution process with two other processes:

- that of regularizing a problem that doesn't need it (that of solving for $v(x)$)

and

- solving an ill-posed problem (that of differentiating $v(x)$ to get $z(x)$).

Regularization

The point to keep in mind is that the application of regularization to produce $v(x)$ effectively limits the space U of data (i.e. $v(x)$) to one for which differentiation IS well-posed.

Remember, whether a problem is well-posed or not depends on the spaces involved.

Regularization

The effect of the regularization on the value of $v(x)$ is to make $v(x)$ a *smoothed* version of $u(x)$.

This smoothing eliminates the instability issues that caused the differentiation problem to be ill-posed. So, it is safe to differentiate $v(x)$ to get $z(x)$.

Regularization

Why is the effect of regularization in this case one of smoothing of the data?

To see why, we need to look at the two terms of the regularization functional:

$$M^{\lambda}[z] = \rho_U(v,u) + \lambda\Omega[v]$$

$$= \int_{-\infty}^{\infty} \left(u(x) - v(x)\right)^2 dx + \lambda \int_{-\infty}^{\infty} \left(\frac{d^2 v(x)}{dx^2}\right)^2 dx$$

Remember, we are trying to find the $v(x)$ which minimizes the functional.

Minimizing the first term $\int_{-\infty}^{\infty} \left(u(x)-v(x)\right)^2 dx$ forces the solution $v(x)$ to be similar to, or consistent with the data $u(x)$.

Minimizing the second term $\lambda \int_{-\infty}^{\infty} \left(\frac{\partial^2 v(x)}{\partial x^2} \right)^2 dx$ forces the second derivative of $v(x)$ to be as small as possible. This is what creates the smoothing action.

Regularization

In computer vision applications the term that includes a Tikhonov stabilizer is often referred to as the ***Smoothness Term***, in recognition of its smoothing influence on the solution.

The other term is often referred to as the ***Data Consistency*** term.

Regularization

Most good edge detection algorithms incorporate a lowpass filtering or smoothing step followed by a differentiation step.

Regularization

Solving the minimization problem required by regularization is usually very computationally intensive, so regularization is usually not used for edge detection - it is simpler to just filter the image and then compute gradients.

Things are not so simple for other vision tasks...

Regularization

In computer vision the data and solution spaces are typically spaces of integer-valued vectors $\mathbb{Z}^N$ rather than Hilbert spaces such as $L^2(\mathbb{R}^N)$.

Regularization

The second term in the functional includes a finite-difference approximation to the continuous second derivative:

$$\frac{d^2 v}{dx^2} \approx \frac{1}{4\Delta}\left(v(x_{i-1}) - 2v(x_i) + v(x_{i+1})\right)$$

Regularization

The functional becomes discretized.

The integrals replaced by discrete summations.

The functional is then something like:

$$M^{\lambda}[v] = \rho_U(v,u) + \lambda\Omega[v]$$

$$= \sum_{i=1}^{N}\left(u(x_i) - v(x_i)\right)^2 + \lambda\sum_{i=2}^{N-1}\left(v(x_{i-1}) - 2v(x_i) + v(x_{i+1})\right)^2$$

Determining λ

How to determine the value of λ (α)?

One of four methods are typically used:

(1) Solve the minimization with λ a constant.

The solution z will then depend on λ.

Then determine the value of λ for which $\rho_{\mathrm{U}}(Az, u) = \delta$.

Determining λ

(2) Similar except determine λ such that $\Omega[z]=1/\delta$.

(3) Generalized Cross-Validation.

The data points themselves are used to choose λ

so as to provide a good prediction of missing data points.

(see paper by Bertero, Poggio, and Torre,

"Ill-Posed Problems in Early Vision")

Determining λ

(4) Ad-hoc.

The regularization problem is set by hand to acheive a suitable level of smoothing, or is loosely based on an estimate of the noise level (e.g. data variance).

How to do the Minimization?

In computer vision algorithms the minimization is usually performed using one of the following methods:

Calculus of Variations:

Derive a (set of) partial differential equation(s) (Euler equations) whose solution is the quantity that minimizes the functional. This equation is usually discretized and solved numerically. In most cases, the Euler equations cannot be put in closed form.

How to do the Minimization?

Gradient Descent:

If the space, F, of solutions admits a gradient, then one can select a point in the solution space and move in the direction of the negative of the gradient of the functional.

That is:

$$\frac{dz}{dt} = -\nabla_F M^{\lambda}(z,u)$$

One follows the negative of the gradient until the gradient becomes zero, corresponding to a minimum of the functional. This does not guarantee reaching the global minimum, however.

How to do the Minimization?

Exhaustive Search:

Discretize the solution space and examine the value of the functional at every point. Usually impractical due to the large (perhaps infinite) volume of the search space.

Stochastic Search:

Randomly sample the solution space, evaluating the functional at each of these samples.