Girosi, Jones, and Poggio

Regularization theory and neural network architectures

presented by

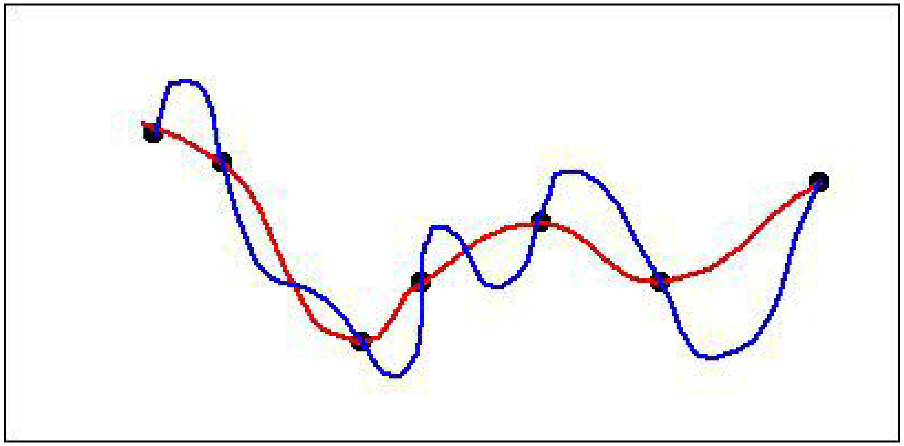
Hsin-Hao Yu

Department of Cognitive Science

October 4, 2001

Learning as function approximation

Goal: Given sparse, noisy samples of a function f, how do we recover f as accurately as possible?



Why is it hard? Infinitely many curves pass through the samples. This problem is *ill-posed*. Prior knowledge about the function must be introduced to make the solution unique. Regularization is a theoretical framework to do this.

Constraining the solution with "stablizers"

Let $(x_1, y_1) \ldots (x_N, y_N)$ be the input data. In order to recover the underlying function, we regularize the ill-posed problem by choosing the function f that minimizes the *functional* H:

$$H[f] = E[f] + \lambda\phi[f]$$

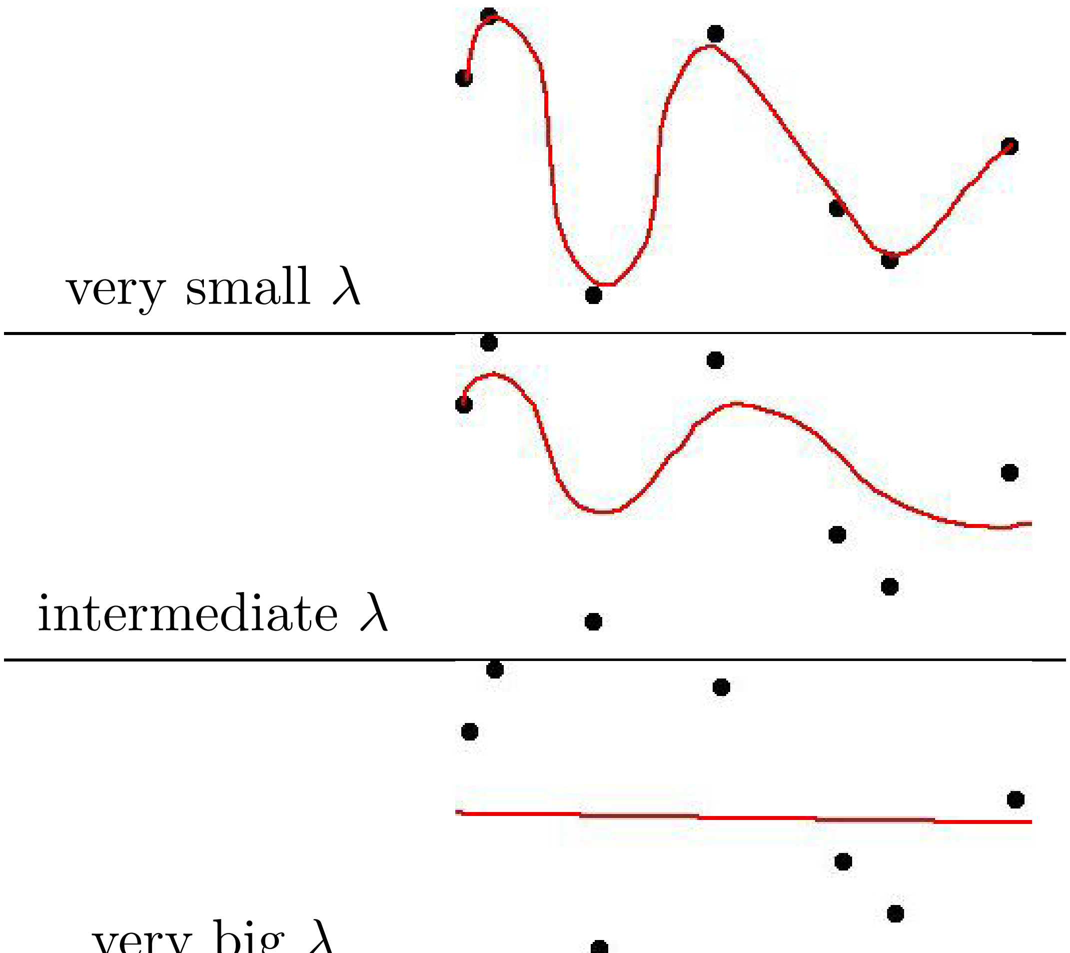
where $\lambda \in \mathcal{R}$ is a user chosen constant,

$E[f]$ represents the "fidelity" of the approximation,

$$E[f] = \frac{1}{2}\sum_{i=1}^{N}(f(x_i) - y_i)^2$$

and $\phi[f]$ represents a constraint on the "smoothness" of f. ϕ is called the *stablizer*.

The fidelity vs. smoothness trade-off



Math review: Calculus of variations

Calculus In order to find a number $\bar{x}$ such that the function $f(x)$ is an extremum at $\bar{x}$, we first calculate the derivative of f, then solve for $\frac{df}{dx} = 0$

Calculus of variations In order to find a function $\bar{f}$ such that the *functional* $H[f]$ is an extremum at $\bar{f}$, we first calculate the *functional derivative* of H, then solve for $\frac{\delta H}{\delta f} = 0$

	Calculus	Calculus of variations
Object for optimization	function	functional
Solution	number	function
Solve for	$\frac{df}{dx} = 0$	$\frac{\delta H}{\delta f} = 0$

An example of regularization

Consider a one-dimensional case. Given input data $(x_1, y_1) \ldots (x_N, y_N)$, we want to minimize the functional

$$H[f] = E[f] + \lambda \phi[f]$$

$$E[f] = \sum_{i=1}^{N} (f(x_i) - y_i)^2$$

$$\phi[f] = \int \left(\frac{d^2 f}{d^2 x} \right)^2 dx$$

To proceed,

$$\frac{\delta H}{\delta f} = \frac{\delta E}{\delta f} + \lambda \frac{\delta \phi}{\delta f}$$

Regularization continued

$$
\begin{aligned}
\frac{\delta E}{\delta f} &= \frac{1}{2}\frac{\delta}{\delta f}\sum_{i=1}^{N}(f(x_i)-y_i)^2 \\
&= \frac{1}{2}\frac{\delta}{\delta f}\int\sum_{i=1}^{N}(f(x)-y_i)^2\delta(x-x_i)dx \\
&= \frac{1}{2}\int\frac{\delta}{\delta f}\sum_{i=1}^{N}(f(x)-y_i)^2\delta(x-x_i)dx \\
&= \int\sum_{i=1}^{N}(f(x)-y_i)\delta(x-x_i)dx
\end{aligned}
$$

$$
\begin{aligned}
\frac{\delta \phi}{\delta f} &= \frac{\delta}{\delta f}\int(\frac{d^2 f}{d^2 x})^2 dx \\
&= \int\frac{d^4 f}{dx^4}dx
\end{aligned}
$$

$$
\begin{aligned}
\frac{\delta H}{\delta f} &= \frac{\delta E}{\delta f} + \lambda\frac{\delta \phi}{\delta f} \\
&= \int(\sum_{i=1}^{N}(f(x)-y_i)\delta(x-x_i) + \lambda\frac{d^4 f}{dx^4})dx
\end{aligned}
$$

Regularization continued

To minimize $H[f]$,

$$\begin{aligned} & \frac{\delta H}{\delta f} = 0 \\ \Rightarrow \quad & \sum_{i=1}^{N} (f(x) - y_i)\delta(x - x_i) + \lambda \frac{d^4 f}{dx^4} = 0 \\ \Rightarrow \quad & \frac{d^4 f}{dx^4} = \frac{1}{\lambda} \sum_{i=1}^{N} (y_i - f(x))\delta(x - x_i) \end{aligned}$$

To solve this differential equation, we calculate the *Green's function* $G(x, \xi)$:

$$\begin{aligned} & \frac{d^4 G(x,\xi)}{dx^4} = \delta(x - \xi) \\ \Rightarrow \quad & G(x, \xi) = |x - \xi|^3 + o(x^2) \end{aligned}$$

We are almost there...

Regularization continued

The solution to $\frac{d^4 f}{dx^4} = \frac{1}{\lambda}\sum_{i=1}^{N}(y_i - f(x))\delta(x - x_i)$ can now be constructed from the Green's function:

$$\begin{aligned} f(x) &= \int \tfrac{1}{\lambda}\textstyle\sum_{i=1}^{N}(y_i - f(\xi))\delta(\xi - \lambda)G(x,\xi)d\xi \\ &= \int \tfrac{1}{\lambda}\textstyle\sum_{i=1}^{N}(y_i - f(\xi))\delta(\xi - \lambda)|x - \xi|^3)d\xi \\ &= \tfrac{1}{\lambda}\textstyle\sum_{i=1}^{N}(y_i - f(x_i))|x - x_i|^3 \end{aligned}$$

The solution turns out to be the *cubic spline*! Oh, one more thing: we need to consider the *null space* of ϕ.

$$Nul(\phi) = \{\psi_1, \psi_2\} = \{1, x\} \qquad (k = 2)$$

$$f(x) = \sum_{i=1}^{N} \frac{y_i - f(x_i)}{\lambda} G(x, x_i) + \sum_{\alpha=1}^{k} d_\alpha \psi_\alpha(x)$$

Solving for the weights

The general solution for minimizing $H[f] = E[f] + \lambda\phi[f]$ is:

$$f(x) = \sum_{i=1}^{N} w_i G(x, x_i) + \sum_{\alpha=1}^{k} d_\alpha \psi_\alpha(x)$$

$$w_i = \frac{y_i - f(x_i)}{\lambda} \qquad (*)$$

where G is the Green's function for the differential operator ϕ, k is the dimension of the null space of ϕ, and ψ_α's are the members of the null space.

But how do we calculate w_i?

$$\begin{aligned} (*) \quad &\Rightarrow \quad \lambda w_i = y_i - f(x_i) \\ &\Rightarrow \quad y_i = f(x_i) + \lambda w_i \end{aligned}$$

Computing w_i continued

$y_i = f(x_i) + \lambda w_i$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} w_i G(x_1, x_i) \\ \vdots \\ \sum_{i=1}^{N} w_i G(x_N, x_i) \end{bmatrix} + \Psi^T d + \lambda \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} G(x_1, x_1) & \dots & G(x_1, x_N) \\ \vdots & & \vdots \\ G(x_N, x_1) & \dots & G(x_N, x_N) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} + \Psi^T d + \lambda w$$

Computing w_i continued

The last statement in matrix form:

$$y = (G + \lambda I)w + \Psi^T d$$

$$0 = \Psi d$$

or,

$$\left(\begin{array}{c|c} G + \lambda I & \Psi \\ \hline \Psi^T & 0 \end{array} \right) \left(\begin{array}{c} w \\ \hline d \end{array} \right) = \left(\begin{array}{c} y \\ \hline 0 \end{array} \right)$$

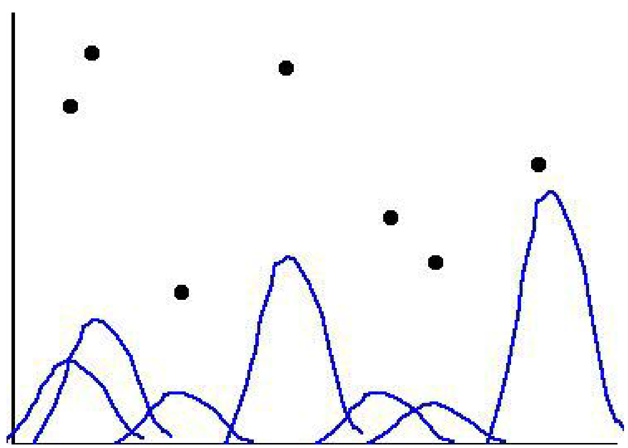
In the special case when the null space is empty (such as the Gaussian kernel),

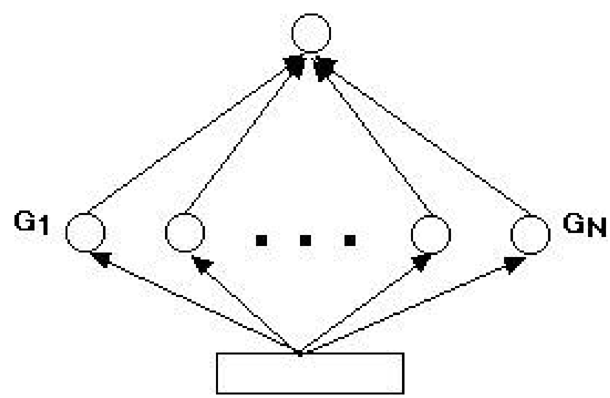
$$w = (G + \lambda I)^{-1} y$$

Interpretations of regularization

The regularized solutions can be understood as:

1. Interpolation with kernels
2. Neural networks (Regularization networks)
3. Data smoothing (*equivalent kernels* as convolution filters)





More stablizers

Various interpolation methods and neural networks can be derived from regularization theory:

- If we require that $\phi[f(x)] = \phi[f(Rx)]$, where R is a rotation matrix, G is radial symmetric. It is the *Radial Basis Function* (RBF). This reflects a priori assumption that all variables have the same relevance, and there are no priviledged directions.
- If

$$\phi[f] = \int e^{\frac{|s|^2}{\beta}} \left| \tilde{f}(s) \right|^2 ds$$

  we get Gaussian kernels.
- Thin plate splines, polynomial splines, multiquadric kernel ...etc.

The probablistic interpretation of RN

Suppose that g is a set of random samples drawn from the function f, in the presence of noise.

- $P[f|g]$ is the probability of function f given the examples g.
- $P[g|f]$ is the the model of noise. We assume Gaussian noise, so $P[g|f] \propto e^{-\frac{1}{2\sigma^2}\sum_i (y_i - f(x_i))^2}$
- $P[f]$ the *a priori* probability of f. This embodies our *a priori* knowledge of the function. Let $P[f] \propto e^{-\alpha\phi[f]}$.

Probabilistic interpretation cont.

By the Bayes Rule,

$$\begin{aligned} P[f|g] &\propto P[g|f]P[f] \\ &\propto e^{-\frac{1}{2\alpha^2}\left(\sum_i (y_i - f(x_i))^2 + 2\alpha\sigma^2\phi[f]\right)} \end{aligned}$$

The MAP estimate of f is therefore the minimizer of:

$$H[f] = \sum_i (y_i - f(x_i))^2 + \lambda\phi[f]$$

where $\lambda = 2\sigma^2\alpha$. It determines the trade-off between the level of noise and the strength of the *a priori* assumption about the solution.

Generalized Regularization Networks

$$w = (G + \lambda I)^{-1} y$$

but calculating $(G + \lambda I)^{-1}$ can be costly, if the number of data points is large. *Generalized Regularization Networks* approximates the regularized solution by using fewer kernel functions.

Applications in early vision

Edge detection

Optical flow

Surface reconstruction

Stereo

...etc.