Regularization via Spectral Filtering

Lorenzo Rosasco

MIT, 9.520 Class 12

March 19, 2012

About this class

Goal To discuss how a class of regularization methods originally designed for solving ill-posed inverse problems, give rise to regularized learning algorithms. These algorithms are kernel methods that can be easily implemented and have a common derivation, but different computational and theoretical properties.

Plan

- From ERM to Tikhonov regularization.
- Linear ill-posed problems and stability.
- Spectral Regularization and Filtering.
- Example of Algorithms.

Basic Notation

- training set $S = \{(x_1, y_1), ..., (x_n, y_n)\}$.
- X is the n by d input matrix.
- $Y = (y_1, \ldots, y_n)$ is the output vector.
- k denotes the kernel function , K the n by n kernel matrix with entries $K_{ij} = k(x_i, x_j)$ and $\mathcal{H}$ the RKHS with kernel k.
- RLS estimator solves

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 .$$

Representer Theorem

We have seen that RKHS allow us to write the RLS estimator in the form

$$f_S^\lambda(x) = \sum_{i=1}^{n} c_i k(x, x_i)$$

with

$$(K + n\lambda I)c = Y$$

where $c = (c_1, \ldots, c_n)$.

The Role of Regularization

We observed that adding a penalization term can be interpreted as way to to control smoothness and avoid overfitting

$$\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \Rightarrow \min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 .$$

Empirical risk minimization

Similarly we can prove that the solution of empirical risk minimization

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

can be written as

$$f_S(x) = \sum_{i=1}^{n} c_i k(x, x_i)$$

where the coefficients satisfy

$$Kc = Y.$$

The Role of Regularization

Now we can observe that adding a penalty has an effect from a numerical point of view:

$$Kc = Y \Rightarrow (K + n\lambda I)c = Y$$

it stabilizes a possibly ill-conditioned matrix inversion problem.

This is the point of view of regularization for (ill-posed) inverse problems.

Ill-posed Inverse Problems

Hadamard introduced the definition of ill-posedness. Ill-posed problems are typically inverse problems.

If $g \in G$ and $f \in F$, with G, F Hilbert spaces, a linear, continuous operator L, consider the equation

$$g = Lf.$$

The direct problem is is to compute g given f; the inverse problem is to compute f given the data g.

The inverse problem of finding f is well-posed when

- the solution exists,
- is unique and
- is stable, that is depends continuously on the initial data g.

Otherwise the problem is ill-posed.

Ill-posed Inverse Problems

Hadamard introduced the definition of ill-posedness. Ill-posed problems are typically inverse problems.
If $g \in G$ and $f \in F$, with G, F Hilbert spaces, a linear, continuous operator L, consider the equation

$$g = Lf.$$

The direct problem is is to compute g given f; the inverse problem is to compute f given the data g.
The inverse problem of finding f is well-posed when

- the solution exists,
- is unique and
- is stable, that is depends continuously on the initial data g.

Otherwise the problem is ill-posed.

Ill-posed Inverse Problems

Hadamard introduced the definition of ill-posedness. Ill-posed problems are typically inverse problems.
If $g \in G$ and $f \in F$, with G, F Hilbert spaces, a linear, continuous operator L, consider the equation

$$g = Lf.$$

The direct problem is is to compute g given f; the inverse problem is to compute f given the data g.
The inverse problem of finding f is well-posed when

- the solution exists,
- is unique and
- is stable, that is depends continuously on the initial data g.

Otherwise the problem is ill-posed.

Linear System for ERM

In the finite dimensional case the main problem is numerical stability.

For example, in the learning setting the kernel matrix can be decomposed as $K = Q\Sigma Q^T$, with $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$, $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$ and $q_1, \ldots, q_n$ are the corresponding eigenvectors.
Then

$$c = K^{-1}Y = Q\Sigma^{-1}Q^TY = \sum_{i=1}^{n} \frac{1}{\sigma_i}\langle q_i, Y\rangle q_i.$$

In correspondence of small eigenvalues, small perturbations of the data can cause large changes in the solution. The problem is ill-conditioned.

Linear System for ERM

In the finite dimensional case the main problem is numerical stability.

For example, in the learning setting the kernel matrix can be decomposed as $K = Q\Sigma Q^T$, with $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$, $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$ and $q_1, \ldots, q_n$ are the corresponding eigenvectors.
Then

$$c = K^{-1}Y = Q\Sigma^{-1}Q^T Y = \sum_{i=1}^{n} \frac{1}{\sigma_i} \langle q_i, Y \rangle q_i.$$

In correspondence of small eigenvalues, small perturbations of the data can cause large changes in the solution. The problem is ill-conditioned.

Regularization as a Filter

For Tikhonov regularization

$$\begin{aligned} c &= (K + n\lambda I)^{-1} Y \\ &= Q(\Sigma + n\lambda I)^{-1} Q^T Y \\ &= \sum_{i=1}^{n} \frac{1}{\sigma_i + n\lambda} \langle q_i, Y \rangle q_i. \end{aligned}$$

Regularization filters out the undesired components.
For $\sigma \gg \lambda n$, then $\frac{1}{\sigma_i + n\lambda} \sim \frac{1}{\sigma_i}$.
For $\sigma \ll \lambda n$, then $\frac{1}{\sigma_i + n\lambda} \sim \frac{1}{\lambda n}$.

Matrix Function

Note that we can look at a scalar function $G_\lambda(\sigma)$ as a function on the kernel matrix.

Using the eigen-decomposition of K we can define

$$G_\lambda(K) = QG_\lambda(\Sigma)Q^T,$$

meaning

$$G_\lambda(K)Y = \sum_{i=1}^{n} G_\lambda(\sigma_i)\langle q_i, Y\rangle q_i.$$

For Tikhonov

$$G_\lambda(\sigma) = \frac{1}{\sigma + n\lambda}.$$

Regularization in Inverse Problems

- In the inverse problems literature many algorithms are known besides Tikhonov regularization.
- Each algorithm is defined by a suitable filter function G_λ.
- This class of algorithms is known collectively as spectral regularization.
- Algorithms are not necessarily based on penalized empirical risk minimization.

Algorithms

- Gradient Descent or Landweber Iteration or L2 Boosting
- ν-method, accelerated Landweber.
- Iterated Tikhonov
- Truncated Singular Value Decomposition (TSVD) Principal Component Regression (PCR)

The spectral filtering perspective leads to a unified framework.

Properties of Spectral Filters

Not every scalar function defines a regularization scheme.

Roughly speaking a good filter function must have the following properties:

- as λ goes to 0, $G_\lambda(\sigma) \to 1/\sigma$ so that

$$G_\lambda(K) \to K^{-1}.$$

- λ controls the magnitude of the (smaller) eigenvalues of $G_\lambda(K)$.

Spectral Regularization for Learning

We can define a class of Kernel Methods as follows.

Spectral Regularization

We look for estimators

$$f_S^\lambda(X) = \sum_{i=1}^{n} c_i k(x, x_i)$$

where

$$c = G_\lambda(K) Y.$$

Gradient Descent

Consider the (Landweber) iteration:

gradient descent

set $c^0 = 0$

for i$= 1, \ldots, t-1$

$$c^i = c^{i-1} + \eta(Y - Kc^{i-1})$$

If the largest eigenvalue of K is smaller than n the above iteration converges if we choose the step-size $\eta = 2/n$.

The above iteration can be seen as the minimization of the empirical risk

$$\frac{1}{n} \|Y - Kc\|_2^2$$

via gradient descent.

Gradient Descent as Spectral Filtering

Note that $c^0 = 0$, $c^1 = \eta Y$,

$$c^2 = \eta Y + \eta(I - \eta K)Y$$

$$\begin{aligned} c^3 &= \eta Y + \eta(I - \eta K)Y + \eta(Y - K(\eta Y + \eta(I - \eta K)Y)) \\ &= \eta Y + \eta(I - \eta K)Y + \eta(I - 2\eta K + \eta^2 K^2)Y \end{aligned}$$

One can prove by induction that the solution at the $t-$th iteration is given by

$$c = \eta \sum_{i=0}^{t-1} (I - \eta K)^i Y.$$

The filter function is

$$G_\lambda(\sigma) = \eta \sum_{i=0}^{t-1} (I - \eta\sigma)^i.$$

Landweber iteration

Note that $\sum_{i\geq 0} x^i = 1/(1-x)$, also holds replacing x with the a matrix. If we consider the kernel matrix (or rather $I - \eta K$) we get

$$K^{-1} = \eta \sum_{i=0}^{\infty} (I - \eta K)^i \sim \eta \sum_{i=0}^{t-1} (I - \eta K)^i.$$

The filter function of Landweber iteration corresponds to a truncated power expansion of K^{-1}.

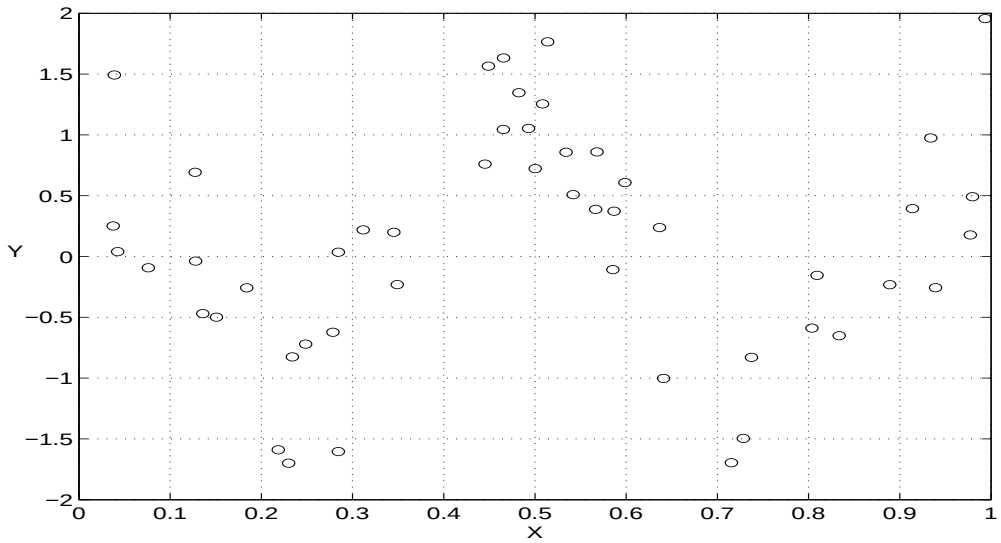
Early Stopping

The regularization parameter is the number of iteration.
Roughly speaking $t \sim 1/\lambda$.

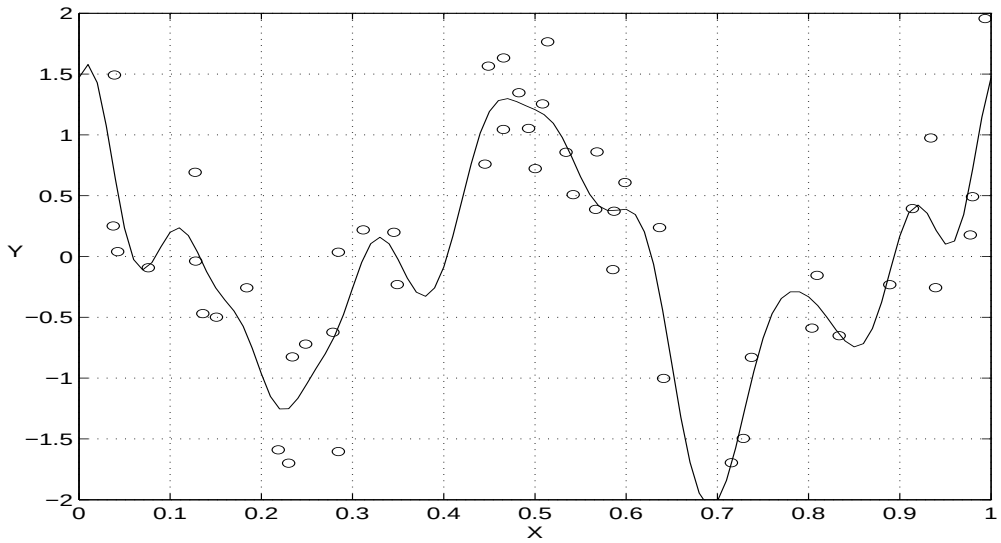
- Large values of t correspond to minimization of the empirical risk and tend to overfit.
- Small values of t tends to oversmooth, recall we start from $c = 0$.

Early stopping of the iteration has a regularization effect.

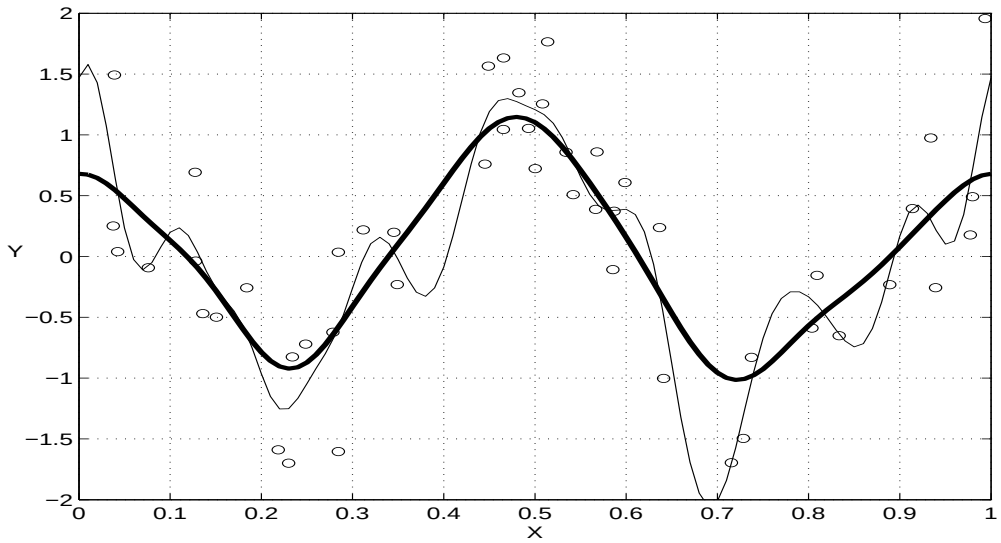
Gradient Descent at Work



Gradient Descent at Work



Gradient Descent at Work



Connection to L_2 Boosting

Landweber iteration (or gradient descent) has been rediscovered in statistics with name of L_2 Boosting.

Boosting

- Then name *Boosting* denotes a large class of methods building estimators as linear (convex) combinations of weak learners.
- Many boosting algorithms can be seen as gradient descent minimization of the empirical risk on the linear span of some basis function.

For Landweber iteration the weak learners are $k(x_i, \cdot), i = 1, \ldots, n$.

ν-method

One can consider an accelerated gradient descent where the The method is implemented by the following iteration.

gradient descent

set $c_0 = 0$

$\omega_1 = (4\nu + 2)/(4\nu + 1)$

$c_1 = c_0 + \frac{\omega_1}{n}(Y - Kc_0)$

for i$= 2, \ldots, t-1$

$$c_i = c_{i-1} + u_i(c_{i-1} - c_{i-2}) + \frac{\omega_i}{n}(Y - Kc_{i-1})$$

$$u_i = \frac{(i-1)(2i-3)(2i+2\nu-1)}{(i+2\nu-1)(2i+4\nu-1)(2i+2\nu-3)}$$

$$\omega_i = 4\frac{(2i+2\nu-1)(i+\nu-1)}{(i+2\nu-1)(2i+4\nu-1)}$$

We need $\sqrt{t}$ iterations to get the same solution that gradient descent would get after t iterations.

Truncated Singular Value Decomposition

This method is one of the oldest regularization techniques and is also called spectral cut-off.

TSVD

- Given the eigen-decomposition $K = Q\Sigma Q^t$, a regularized inverse of the kernel matrix is built discarding all the eigenvalues before the prescribed threshold λn.
- It is described by the filter function $G_\lambda(\sigma) = 1/\sigma$ if $\sigma \geq \lambda/n$ and 0 otherwise.

Dimensionality Reduction and Generalization

Interestingly enough, one can show that TSVD is equivalent to the following procedure:

- (unsupervised) projection of the data using (kernel) PCA.
- Empirical risk minimization on projected data without any regularization.

The only free parameter is the number of components we retain for the projection.

Dimensionality Reduction and Generalization (cont.)

Projection Regularizes!

Doing KPCA and then RLS is redundant.

If data are centered Spectral regularization (also Tikhonov) can see as filtered projection on the principal components.

Comments on Complexity and Parameter Choice

- Iterative methods perform matrix vector multiplication $O(n^2)$ at each iteration and the regularization parameter is the number of iteration itself.
- There is not a closed form for leave one out error.
- Parameter tuning is different from method to method.
  - Compared to RLS in iterative and projected methods the regularization parameter is naturally *discrete*.
  - TSVD has a natural range for the search of the regularization parameter.
  - For TSVD the regularization parameter can be interpreted in terms of dimensionality reduction.

Filtering, Regularizartion and Learning

The idea of using regularization from inverse problems in statistics (see Wahba) and machine learning (see Poggio and Girosi) is now well known.

Ideas coming from inverse problems regarded mostly the use of Tikhonov regularization.

The notion of filter function was studied in machine learning and gave a connection between function approximation in signal processing and approximation theory. The work of Poggio and Girosi enlighted the relation between neural network, radial basis function and regularization.

Filtering was typically used to define a penalty for Tikhonov regularization, in the following it is used to define algorithms different though similar to Tikhonov regularization.

Final remarks

- Many different principles lead to regularization: penalized minimization, iterative optimization, projection. The common intuition is that they enforce stability of the solution.
- All the methods are implicitly based on the use of square loss. For other loss function different notion of stability can be used.

Appendices

- Appendix 1: Other examples of Filters: accelerated Landweber and Iterated Tikhonov.
- Appendix 2: TSVD and PCA.
- Appendix 3: Some thoughts about Generalization of Spectral Methods.

Appendix 1 :ν-method

The so called ν-method or accelerated Landweber iteration can be thought as an accelerated version of gradient descent.

The filter function is $G_t(\sigma) = p_t(\sigma)$ with p_t a polynomial of degree $t - 1$.

The regularization parameter (think of $1/\lambda$) is $\sqrt{t}$ (rather than t): fewer iterations are needed to attain a solution.

ν-method (cont.)

The method is implemented by the following iteration.

gradient descent

set $c_0 = 0$

$\omega_1 = (4\nu + 2)/(4\nu + 1)$

$c_1 = c_0 + \frac{\omega_1}{n}(Y - Kc_0)$

for i$= 2, \ldots, t-1$

$$c_i = c_{i-1} + u_i(c_{i-1} - c_{i-2}) + \frac{\omega_i}{n}(Y - Kc_{i-1})$$

$$u_i = \frac{(i-1)(2i-3)(2i+2\nu-1)}{(i+2\nu-1)(2i+4\nu-1)(2i+2\nu-3)}$$

$$\omega_i = 4\frac{(2i+2\nu-1)(i+\nu-1)}{(i+2\nu-1)(2i+4\nu-1)}$$

Iterated Tikhonov

The following method can be seen a combination of Tikhonov regularization and gradient descent.

gradient descent

set $c_0 = 0$

for i$= 0, \ldots, t-1$

$$(K + n\lambda I)c_i = Y + n\lambda c_{i-1}$$

The filter function is:

$$G_\lambda(\sigma) = \frac{(\sigma+\lambda)^t - \lambda^t}{\sigma(\sigma+\lambda)^t}.$$

Iterated Tikhonov (cont.)

Both the number of iteration and λ can be seen as regularization parameters.

It can be used to enforce more smoothness on the solution.

Tikhonov regularization suffers from a *saturation* effect: it cannot exploit the regularity of the solution beyond a certain critical value.

Appendix 2: TSVD and Connection to PCA

Principal component Analysis is a well known dimensionality reduction technique often used as preprocessing in learning.

PCA

- Assuming centered data, $X^T X$ is the covariance matrix and its eigenvectors $(V^j)_{j=1}^d$ are the principal components.
- PCA amounts to map each example x_i in

$$\tilde{x}_i = (x_i^T V^1, \ldots, x_i^T V^m)$$

where $m < \min\{n, d\}$.

notation: x_i^T is the transpose of the first row (example) of X.

PCA (cont.)

The above algorithm can be written using only the linear kernel matrix XX^T and its eigenvectors $(U^i)_{i=1}^n$.

The eigenvalues of XX^T and X^TX are the same and

$$V^j = \frac{1}{\sqrt{\sigma_i}} X^T U^j.$$

Then

$$\tilde{x}_i = (\frac{1}{\sqrt{\sigma_i}} \sum_{j=1}^{n} U_j^1 x_i^T x_j), \ldots, \frac{1}{\sqrt{\sigma_n}} \sum_{j=1}^{n} U_j^m x_i^T x_j).$$

Note that $x_i^T x_j = k(x_i, x_j)$.

Kernel PCA

We can perform a non linear principal component analysis, namely KPCA, by choosing non linear kernel functions.

Using $K = Q\Sigma Q^T$ we can rewrite the projection in vector notation.

If we let $\Sigma_M = diag(\sigma_1, \cdots, \sigma_m, 0, \cdots, 0)$ then the projected data matrix $\tilde{X}$ is

$$\tilde{X} = KQ\Sigma_m^{-1/2}$$

Principal Component Regression

ERM on the projected data

$$\min_{\beta \in \mathbb{R}^m} \left\| Y - \beta \tilde{X} \right\|_n^2,$$

is *equivalent* to perform truncated singular values decomposition on the original problem.

Representer Theorem tells us that

$$\beta^T \tilde{x}_i = \sum_{j=1}^{n} \tilde{x}_j^T \tilde{x}_i c_j$$

with

$$c = (\tilde{X}\tilde{X}^T)^{-1} Y.$$

Dimensionality Reduction and Generalization

Using $\tilde{X} = KQ\Sigma_m^{-1/2}$ we get

$$\tilde{X}\tilde{X}^T = Q\Sigma Q^T Q\Sigma_m^{-1/2}\Sigma_m^{-1/2} Q^T Q\Sigma Q^T = Q\Sigma_m Q^T.$$

so that

$$c = Q\Sigma_m^{-1} Q^T Y = G_\lambda(K)Y,$$

where G_λ is the filter function of TSVD.

The two procedure are equivalent. The regularization parameter is the eigenvalue threshold in one case and the number of components kept in the other case.

Appendix 3: Why Should These Methods Learn?

we have seen that

$$G_\lambda(\mathbf{K}) \to K^{-1} \quad \text{if} \quad \lambda \to 0$$

anyway usually, we DON'T want to solve

$$Kc = Y$$

since it would simply correspond to an over-fitting solution

stability vs generalization

how can we show that **stability** ensures **generalization**?

Population Case

It is useful to consider what happens if we know the **true** distribution.

integral operator

for n large enough

$$\frac{1}{n}K \sim L_k f(s) = \int_X k(x,s) f(x) p(x) dx$$

the ideal problem

for n large enough we have

$$Kc = Y \sim L_k f = L_k f_\rho$$

where f_ρ is the regression (target) function defined by $f_\rho(x) = \int_Y y p(y|x) dy$

Regularization in the Population Case

it can be shown that which is the least squares problem associated to $L_k f = L_k f_\rho$.

tikhonov regularization in this case is simply

or equivalently

$$f^\lambda = (L_k f + \lambda I)^{-1} L_k f_\rho$$

Fourier Decomposition of the Regression Function

fourier decomposition of f_ρ and f^λ

if we diagonalize L_k to get the eigensystem $(t_i, \phi_i)_i$ we can write

$$f_\rho = \sum_i \langle f_\rho, \phi_i \rangle \, \phi_i$$

perturbations affect high order components.
tikhonov regularization can be written as

$$f^\lambda = \sum_i \frac{t_i}{t_i + \lambda} \langle f_\rho, \phi_i \rangle \, \phi_i$$

sampling IS a perturbation

stabilizing the problem with respect to random discretization (sampling) we can recover f_ρ