1 Introduction

Here we discusses how a class of regularization methods originally designed to solve 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. However, they have different computational and theoretical properties. In particular, we discuss:

- ERM in the context of Tikhonov Regularization
- Linear ill-posed problems and stability
- Spectral regularization and filtering
- Examples of algorithms

2 Tikhonov Regularization and ERM

Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, X be an $n \times d$ input matrix, and $Y = (y_1, \ldots, y_n)$ be an output vector. k denotes the kernel function, and K is the $n \times n$ kernel matrix with entries $K_{i,j} = k(x_i, x_j)$. $\mathcal{H}$ is the RKHS with kernel k. The RLS estimator solves

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

By the Representer Theorem, we know that we can write the RLS estimator in the form

$$f_S^\lambda(x) = \sum_{i=1}^{n} c_i k(x, x_i) \tag{2}$$

with

$$(K + n\lambda I)c = Y, \tag{3}$$

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

Likewise, the solution to ERM

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{4}$$

can be written as

$$f_S(x) = \sum_{i=1}^{n} c_i k(x, x_i) \tag{5}$$

where the coefficients satisfy

$$Kc = Y. \tag{6}$$

We can interpret the kernel problem to be ERM with a smoothness term, $\lambda \|f\|_{\mathcal{H}}^2$. The smoothness term helps avoid overfitting.

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \Longrightarrow \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2. \tag{7}$$

The corresponding relationship between the equations for the coefficients, c, is

$$Kc = Y \Longrightarrow (K + n\lambda I)c = Y. \tag{8}$$

From a numerical point of view, the regularization factor $n\lambda I$ stabilizes a possibly ill-conditioned inverse problem.

3 Ill-posed Inverse Problems

Hadamard introduced the definition of ill-posedness. Ill-posed problems are typically inverse problems. Let G and F be Hilbert spaces, and L a continuous linear operator between them. Let $g \in G$ and $f \in F$, where

$$g = Lf. \tag{9}$$

The direct problem is to compute g given f, the inverse problem is to compute f given the data g. As we know, the inverse problem of finding f is well-posed when

- the solution exists,
- is unique, and
- is stable (depends continuously on the data g).

Otherwise, the problem is ill-posed.

3.1 Regularization as a Filter

In the finite-dimensional case, the main problem is numerical stability. For example, let the kernel matrix have $K = Q\Sigma Q^t$, where Σ is the diagonal matrix $\text{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$ and $q_1, \ldots, q_n$ the corresponding eigenvectors. Then,

$$c = K^{-1}Y = Q\Sigma^{-1}Q^tY = \sum_{i=1}^{n} \frac{1}{\sigma_i} \langle q_i, Y \rangle q_i. \tag{10}$$

Terms in this sum with small eigenvalues σ_i give rise to numerical instability. For instance, if there are eigenvalues of zero, the matrix will be impossible to invert. As eigenvalues tend toward zero, the matrix tends toward rank-deficiency, and inversion becomes less stable. Statistically, this will correspond to high variance of the coefficients c_i.

For Tikhonov regularization,

$$\begin{aligned} c &= (K + n\lambda I)^{-1}Y & (11)\\ &= Q(\Sigma + n\lambda I)^{-1}Q^tY & (12)\\ &= \sum_{i=1}^{n} \frac{1}{\sigma_i + n\lambda} \langle q_i, Y \rangle q_i. & (13) \end{aligned}$$

This shows that regularization as the effect of suppressing the influence of small eigenvalues in computing the inverse. In other words, *regularization filters out the undesired components.*

- If $\sigma \gg \lambda n$, then $\frac{1}{\sigma_i + n\lambda} \sim \frac{1}{\sigma_i}$.
- If $\sigma \ll \lambda n$, then $\frac{1}{\sigma_i + n\lambda} \sim \frac{1}{\lambda n}$.

We can define more general filters. Let $G_\lambda(\sigma)$ be a function on the kernel matrix. We can eigendecompose K to define

$$G_\lambda(K) = QG_\lambda(\Sigma)Q^t, \tag{14}$$

meaning

$$G_\lambda(K)Y = \sum_{i=1}^{n} G_\lambda(\sigma_i) \langle q_i, Y \rangle q_i. \tag{15}$$

For Tikhonov Regularization

$$G_\lambda(\sigma) = \frac{1}{\sigma + n\lambda}. \tag{16}$$

3.2 Regularization Algorithms

In the inverse problems literature, many algorithms are known besides Tikhonov regularization. Each algorithm is defined by a suitable filter G. This class of algorithms performs *spectral regularization*. They are not necessarily based on penalized empirical risk minimization (or regularized ERM).

In particular, the spectral filtering perspective leads to a unified framework for the following algorithms:

- Gradient Descent (or Landweber Iteration or L_2 Boosting)
- ν-method, accelerated Landweber
- Iterated Tikhonov Regularization
- Truncated Singular Value Decomposition (TSVD) and Principle Component Regression (PCR)

Not every scalar function G defines a regularization scheme. Roughly speaking, a good filter must have the following properties:

- As $\lambda \to 0$, $G_\lambda(\sigma) \to 1/\sigma$, so that
$$G_\lambda(K) \to K^{-1}. \tag{17}$$
- λ controls the magnitude of the (smaller) eigenvalues of $G_\lambda(K)$.

Definition 1 *Spectral Regularization techniques are Kernel Methods with estimators*

$$f_S^\lambda = \sum_{i=1}^{n} c_i k(x, x_i) \tag{18}$$

that have

$$c = G_\lambda(K)Y. \tag{19}$$

3.3 The Landweber Iteration

Consider the Landweber Iteration, a numerical algorithm for solving linear systems:

`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 g the above iteration converges if we choose the step size $\eta = 2/g$.

The above algorithm can be viewed as using gradient descent to iteratively minimize the empirical risk,

$$\frac{1}{n}\|Y - Kc\|_2^2, \tag{20}$$

and stopping at iteration t. This is because the derivative of empirical risk with respect to c is precisely $(2/n)(Y - Kc)$. Note that

$$\begin{aligned} c_i &= \nu Y + (I - \nu K)c_{i-1} && (21)\\ &= \nu Y + (I - \nu K)[\nu Y + (I - \nu K)c_{i-2}] && (22)\\ &= \nu Y + (I - \nu K)\nu Y + (I - \nu K)^2 c_{i-2} && (23)\\ &= \nu Y + (I - \nu K)\nu Y + (I - \nu K)^2 \nu Y + (I - \nu K)^3 c_{i-3} && (24) \end{aligned}$$

Continuing this expansion and noting that $c_1 = \nu Y$, it is easy to see that the solution at the t-th iteration is given by

$$c = \eta \sum_{i=0}^{t-1} (I - \eta K)^i Y. \tag{25}$$

Therefore, the filter function is

$$G_\lambda(\sigma) = \eta \sum_{i=0}^{t-1} (I - \eta\sigma)^i. \tag{26}$$

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

$$K^{-1} = \eta \sum_{i=0}^{\infty} (I - \eta K)^i \sim \eta \sum_{i=0}^{t-1} (I - \eta K)^i. \tag{27}$$

The filter function of the Landweber iteraiton corresponds to a truncated power expansion of K^{-1}. The regularization parameter is the number of iterations. 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 tend to oversmooth, recall we start from $c = 0$.

Therefore, early stopping has a regularizing effect, see Figure 1.

The Landweber iteration was rediscovered in statistics under the name L_2 Boosting.

Definition 2 *Boosting methods build estimators as convex combinations of weaks learners.*

Many boosting algorithms are gradient descent minimization of the empirical risk or the linear span of a basis function. For the Landweber iteration, the weak learners are $k(x_i, \cdot)$, for $i = 1, \ldots, n$.

An version of gradient descent is called the ν method, and requires $\sqrt{t}$ iterations to get the same solution that gradient descent would after t iterations.

`set` $c_0 = 0$
$\omega_1 = (4\nu + 2)/(4\nu + 1)$
$c_1 = c_0 + \omega_1(Y - Kc_0)/n$
`for` $i = 2, \ldots, t-1$
  $c_i = c_{i-1} + u_i(c_{i-1} - c_{i-2}) + \omega_i(Y - Kc_{i-1})/n$
  $u_i = \frac{(i-1)(2i-3)(2i+2\nu-1)}{(i+2\nu-1)(2i+4\nu-1)(2i+2\nu-3)}$
  $\omega_i = 4 \cdot \frac{(2i+2\nu-1)(i+\nu-1)}{(i+2\nu-1)(2i+4\nu-1)}$

3.4 Truncated Singular Value Decomposition (TSVD)

Also called "spectral cut-off," the TSVD method works as follows: Given the eigendecomposition $K = Q\Sigma Q^t$, a regularized inverse of the kernel matrix is built by discarding all the eigenvalues before the prescribed threshold λn. It is described by the filter function

$$G_\lambda(\sigma) = \begin{cases} 1/\sigma & \text{if } \sigma \geq \lambda n \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

It can be shown that TSVD is equivalent to

- unsupervised projection of the data by kernel PCA (KPCA), and

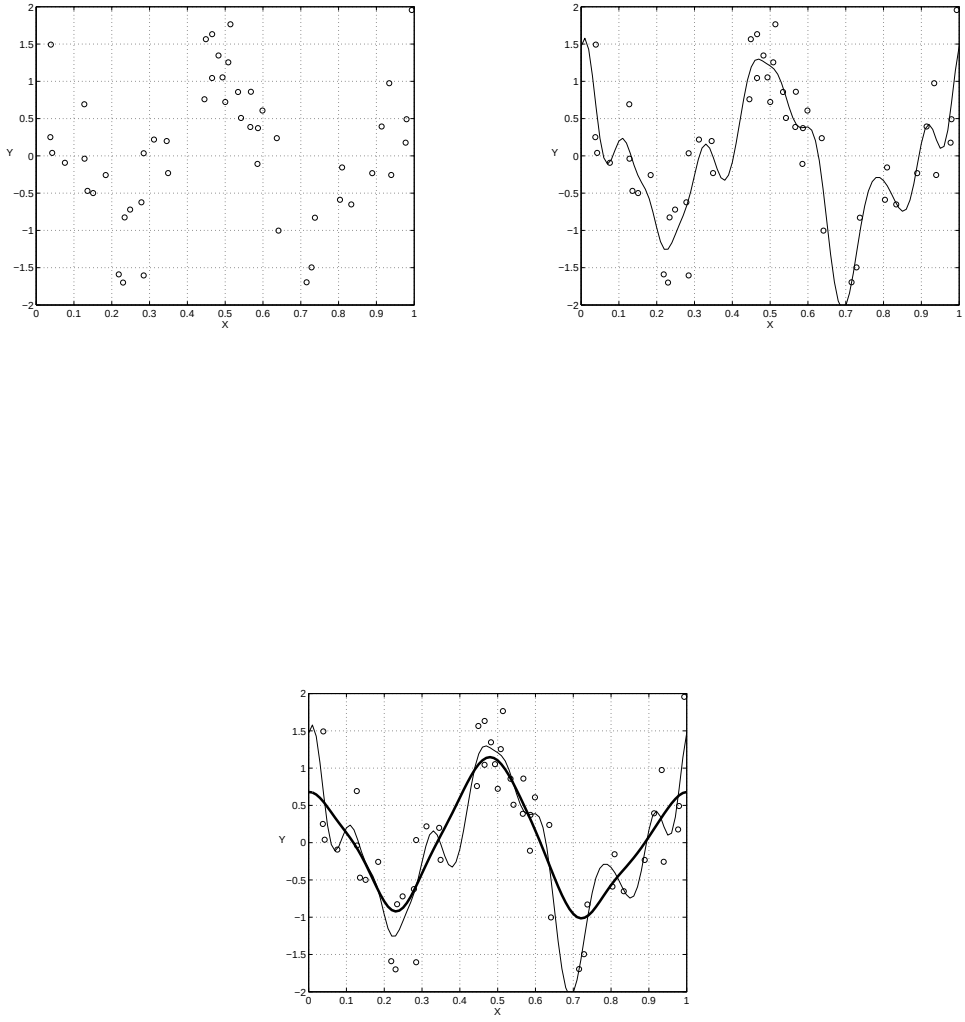

Figure 1: Data, overfit, and regularized fit.

- ERM on projected data without regularization

The only free parameter is the number of components retained for the projection. Thus, doing KPCA and then RLS is redundant. If the data is centered, Spectral and Tikhonov regularization can be seen as filtered projection on the principle components.

3.5 Complexity and Parameter Choice

Iterative methods perform matrix-vector multiplication ($O(n^2)$ operations) at each iteration, and the regularization parameter is the number of iterations. There is no closed form for LOOCV, making parameter tuning expensive.

Tuning differs from method to method. In iterative and projected methods the tuning parameter is naturally discrete, unlike in RLS. TSVD has a natural search-range for the parameter, and choosing it can be interpreted in terms of dimensionality reduction.

4 Conclusion

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

The idea of using regularization from inverse problems in statistics [**?**] and machine learning [**?**] is now well known. The ideas from inverse problems usually regard the use of Tikhonov regularization. Filter functions were studied in machine learning and gave a connection between function approximation in signal processing and approximation theory. It was typically used to define a penalty for Tikhonov regularization or similar methods. [**?**] showed the relationship between the neural network, the radial basis function, and regularization.

5 Appendices

There are three appendices, which cover:

- Appendix 1: Other examples of Filters: accelerated Landweber and Iterated Tikhonov.
- Appendix 2: TSVD and PCA.
- Appendix 3: Some thoughts about the generalization of spectral methods.

5.1 Appendix 1: The ν-method

The ν-method or accelerated Landweber iteration is 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.

It is implemented by the following iteration (repeating from before):

`set` $c_0 = 0$
$\omega_1 = (4\nu + 2)/(4\nu + 1)$
$c_1 = c_0 + \omega_1(Y - Kc_0)/n$
`for` $i = 2, \ldots, t-1$
$c_i = c_{i-1} + u_i(c_{i-1} - c_{i-2}) + \omega_i(Y - Kc_{i-1})/n$
$u_i = \frac{(i-1)(2i-3)(2i+2\nu-1)}{(i+2\nu-1)(2i+4\nu-1)(2i+2\nu-3)}$

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

The following method, Iterated Tikhonov, is a combination of Tikhonov regularization and gradient descent:

`set` $c_0 = 0$
`for` $i = 1, \ldots, t-1$
$(K + n\lambda I)c_i = Y + n\lambda c_{i-1}$

The filter function is:

$$G_\lambda(\sigma) = \frac{(\sigma+\lambda)^t - \lambda^t}{\sigma(\sigma+\lambda)^t}. \tag{29}$$

Both the number of iterations and λ can be seen as regularization parameters, and can be used to enforce smoothness. However, Tikhonov regularization suffers from a *saturation* effect: it cannot exploit the regularity of the solution beyond a certain critical value.

5.2 Appendix 2: TSVD and Connection to PCA

Principle Component Analysis (PCA) is a well known dimensionality reduction technique often used as preprocessing in learning.

Definition 3 *Assuming centered data X, X^tX is the covariance matrix and its eigenvectors $(V^j)_{j=1}^d$ are the principle components. x_j^t is the tranposes of the first row (example) in X. PCA amounts to mapping each example x_j into*

$$\tilde{x}_j = (x_j^t V^1, \ldots, x_j^t V^m) \tag{30}$$

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

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 XX^t are the same and

$$V^j = \frac{1}{\sqrt{\sigma_j}} X^t U^j. \tag{31}$$

Then,

$$\tilde{x}_j = \left(\frac{1}{\sqrt{\sigma_1}} \sum_{j=1}^n U_j^1 x_j^t x_j, \ldots, \frac{1}{\sqrt{\sigma_n}} \sum_{j=1}^n U_j^m x_j^t x_j. \right) \tag{32}$$

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

We can perform a nonlinear version of PCA, KPCA, using a nonlinear kernel. Let K eigendecompose $K = Q\Sigma Q^t$. We can rewrite the projection in vector notation:

Let $\Sigma_M = \text{diag}(\sigma_1, \ldots, \sigma_m, 0, \ldots, 0)$, then the projected data matrix $\tilde{X}$ is

$$\tilde{X} = KQ\Sigma_M^{-1/2}. \tag{33}$$

Doing ERM on the projected data,

$$\min_{\beta \in \mathbb{R}^m} \|Y - \beta \tilde{X}\|_n^2 \tag{34}$$

is equivalent to performing TSVD on the original problem. The Representer Theorem tells us that

$$\beta^t \tilde{x}_i = \sum_{j=1}^n \tilde{x}_j^t \tilde{x}_i c_j \tag{35}$$

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

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

$$\tilde{X}\tilde{X}^t = Q\Sigma Q^t Q\Sigma_m^{-1/2}\Sigma_m^{-1/2}Q^t Q\Sigma Q^t = Q\Sigma_m Q^t, \tag{36}$$

so that

$$c = Q\Sigma_m^{-1}Q^t Y = G_\lambda(K)Y, \tag{37}$$

where G_λ is the filter function of TSVD.

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

5.3 Appendix 3: Why Should These Methods Learn?

We have seen that

$$G_\lambda(K) \to K^{-1} \text{ as } \lambda \to 0, \tag{38}$$

and usually we *don't* want to solve

$$Kc = Y, \tag{39}$$

since it would simply correspond to an over-fitting solution. It is useful to consider what happens if we know the *true* distribution. Using integral-operator notation, if n is large enough,

$$\frac{1}{n}K \sim L_k f(s) = \int_X k(x,s)f(x)p(x)dx. \tag{40}$$

In the ideal problem, if n is large enough, we have

$$Kc = Y \sim L_k f = L_k f_\rho, \tag{41}$$

where f_ρ is the regression (target) function defined by

$$f_\rho(x) = \int_Y y \cdot p(y|x)dy. \tag{42}$$

It can be shown that which is the least square problem associated to $L_k f = L_k f_\rho$. Tikhonov regularization in this case is simply

$$MISSINGFROMSLIDES. \tag{43}$$

or equivalently

$$f^\lambda = (L_k f + \lambda I)^{-1} L_k f_\rho. \tag{44}$$

If we diagonalize L_k to get the eigensystem $\{(t_j, \phi_j)\}_j$, we can write

$$f_\rho = \sum_j \langle f_\rho, \phi_j \rangle \, \phi_j. \tag{45}$$

Perturbations affect higher order components. Tikhonov Regularization can be written as

$$f^\lambda = \sum_j \frac{t_j}{t_j + \lambda} \langle f_\rho, \phi_j \rangle \, \phi_j. \tag{46}$$

Sampling is a perturbation. Stabilizing the problem with respect to random discretization (sampling), we can recover f_ρ.