A Practical Use of Regularization for Supervised Learning with Kernel Methods

Marco Prato[1], Luca Zanni[1]

1. Dipartimento di Scienze Fisiche, Informatiche e Matematiche, Università di Modena e Reggio Emilia, Italy

Optimization Techniques for Inverse Problems II
Modena (Italy), September 20-21, 2012

Abstract

In several supervised learning applications, it happens that reconstruction methods have to be applied repeatedly before being able to achieve the final solution. In these situations, the availability of learning algorithms able to provide effective predictors in a very short time may lead to remarkable improvements in the overall computational requirement. Here we consider the kernel ridge regression problem and we look for predictors given by a linear combination of kernel functions plus a constant term, showing that an effective solution can be obtained very fastly by applying specific regularization algorithms directly to the linear system arising from the Empirical Risk Minimization problem.

Learning with the quadratic loss

Regularized least squares (RLS) for learning: given a training set

$$S = \{(\boldsymbol{x}_i, y_i) : i = 1, \dots, n\} \subset X \times Y, \quad X \subset \mathbb{R}^d, \quad Y \subset \mathbb{R}$$

find the decision function $f : X \to Y$ to predict the label y of new examples **x** by solving

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

where: - λ is a positive regularization parameter,
- $\mathcal{H}$ is a Reproducing Kernel Hilbert Space with kernel K [1].

Representer Theorem: the solution of (1) in a RKHS assumes the form

$$f_0(\boldsymbol{x}) = \sum_{i=1}^{n} c_i K(\boldsymbol{x}, \boldsymbol{x}_i), \qquad (2)$$

where **c** is the solution of the linear system [2]

$$(\boldsymbol{K} + n\lambda \boldsymbol{I})\boldsymbol{c} = \boldsymbol{y}, \qquad (3)$$

being $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

More general prediction function (SVM choice):

$$f(\boldsymbol{x}) = f_0(\boldsymbol{x}) + b, \qquad b \in \mathbb{R} \qquad (4)$$

Two different generalizations of (1) [3,4,5]:

(a)
$$\min_{f_0 \in \mathcal{H},\ b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_0(\boldsymbol{x}_i) - b)^2 + \lambda \|f_0\|_{\mathcal{H}}^2 \qquad (5)$$

(constant b not penalized) (constant b penalized)

(b)
$$\min_{f_0 \in \mathcal{H},\ b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_0(\boldsymbol{x}_i) - b)^2 + \lambda (\|f_0\|_{\mathcal{H}}^2 + b^2) \qquad (6)$$

leading, respectively, to the linear systems (where $\mathbf{1} = (1,\dots,1)^t$)

$$\begin{cases} (\boldsymbol{K} + n\lambda \boldsymbol{I})\boldsymbol{c} + b\mathbf{1} = \boldsymbol{y} \\ \mathbf{1}^t \boldsymbol{c} = 0 \end{cases} \quad (7) \qquad \begin{cases} (\boldsymbol{K} + \mathbf{1}\mathbf{1}^t + n\lambda \boldsymbol{I})\boldsymbol{c} + b\mathbf{1} = \boldsymbol{y} \\ b = \mathbf{1}^t \boldsymbol{c} \end{cases} \quad (8)$$

CPU time required by these approaches: $\mathcal{O}(4n^3/3)$, due to the eigendecomposition of **K** or $\mathbf{K} + \mathbf{1}\mathbf{1}^t$.

Tikhonov and Conjugate Gradient

From $\mathcal{H}$ to $\mathbb{R}^n$: problem (3) is the Lavrentiev [6] regularized version of the linear system **Kc** = **y**, that arises from the minimization of the Least Squares functional when the desired solution is in the form (2).

With solutions in the form (4), such linear system becomes

$$\widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{c}} = \boldsymbol{y}, \qquad \widetilde{\boldsymbol{K}} = (\boldsymbol{K}\ \mathbf{1}), \qquad \widetilde{\boldsymbol{c}} = (c_1, \dots, c_n, b)^t \qquad (9)$$

Instead of adding a penalty term on f – as in (5), (6) –, we can directly apply a regularization algorithm to the linear system (9), e.g.:

- Tikhonov

$$(\widetilde{\boldsymbol{K}}^t \widetilde{\boldsymbol{K}} + n\lambda \boldsymbol{I})\widetilde{\boldsymbol{c}} = \widetilde{\boldsymbol{K}}^t \boldsymbol{y}$$

- Conjugate Gradient

$$\begin{aligned}
&\widetilde{\boldsymbol{c}}^{(0)} \in \mathbb{R}^{n+1}, \widetilde{\boldsymbol{r}}^{(0)} = \boldsymbol{y} - \widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{c}}^{(0)}, \widetilde{\boldsymbol{d}}^{(0)} = \widetilde{\boldsymbol{K}}^t \widetilde{\boldsymbol{r}}^{(0)} \\
&\text{for } i = 1, \dots, t \\
&\qquad \alpha_i = \|\widetilde{\boldsymbol{K}}^t \widetilde{\boldsymbol{r}}^{(i-1)}\|_2^2 / \|\widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{d}}^{(i-1)}\|_2^2 \\
&\qquad \widetilde{\boldsymbol{c}}^{(i)} = \widetilde{\boldsymbol{c}}^{(i-1)} + \alpha_i \widetilde{\boldsymbol{d}}^{(i-1)} \\
&\qquad \widetilde{\boldsymbol{r}}^{(i)} = \widetilde{\boldsymbol{r}}^{(i-1)} - \alpha_i \widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{d}}^{(i-1)} \\
&\qquad \beta_i = \|\widetilde{\boldsymbol{K}}^t \widetilde{\boldsymbol{r}}^{(i)}\|_2^2 / \|\widetilde{\boldsymbol{K}}^t \widetilde{\boldsymbol{r}}^{(i-1)}\|_2^2 \\
&\qquad \widetilde{\boldsymbol{d}}^{(i)} = \widetilde{\boldsymbol{K}}^t \widetilde{\boldsymbol{r}}^{(i)} + \beta_i \widetilde{\boldsymbol{d}}^{(i-1)} \\
&\text{end}
\end{aligned}$$

Why Tikhonov: the regularized solution can be written in term of the SVD of $\widetilde{\boldsymbol{K}}$:

$$\widetilde{\boldsymbol{c}}_\lambda = \sum_{\widetilde{\sigma}_i \neq 0} \frac{\widetilde{\sigma}_i (\boldsymbol{y}^t \widetilde{\boldsymbol{u}}_i)}{\widetilde{\sigma}_i^2 + n\lambda} \widetilde{\boldsymbol{v}}_i$$

For rank-deficient kernels (e.g., the linear kernel when d << n) many singular values are zero, thus allowing to avoid the calculation of the corresponding singular vectors.

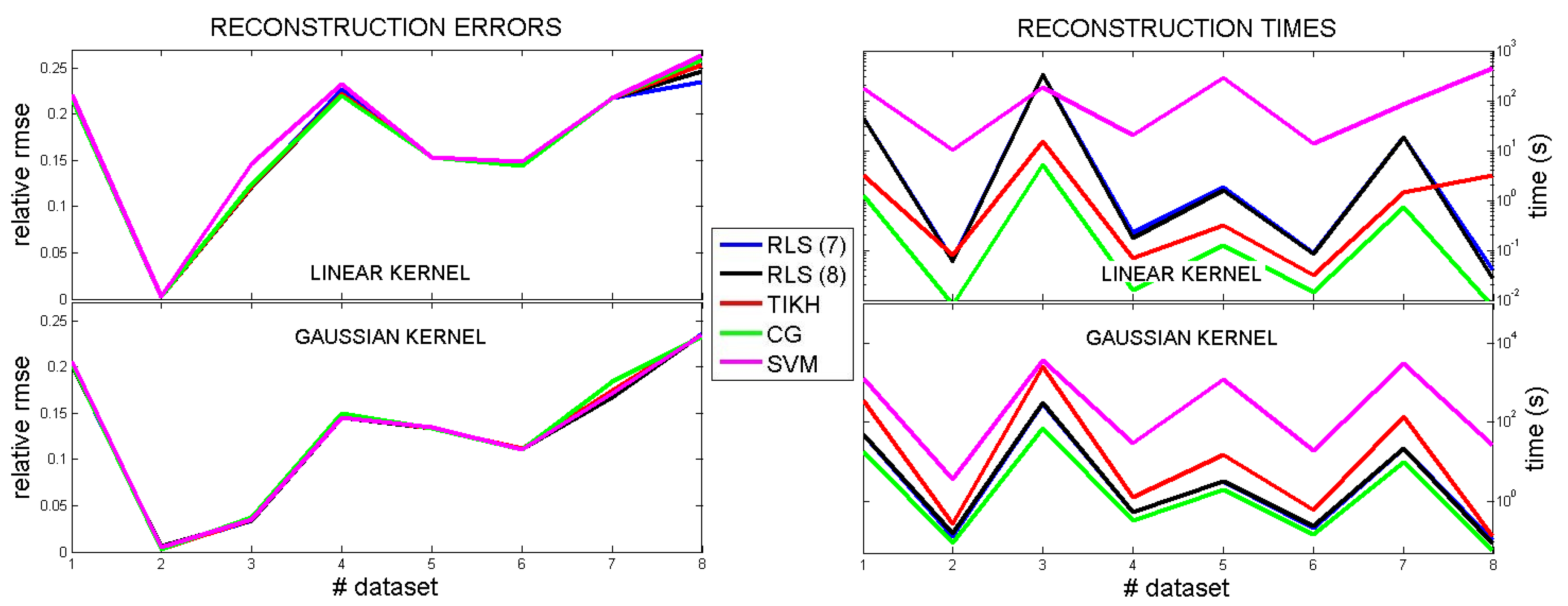
Why Conjugate Gradient: $\mathcal{O}(tn^2)$ instead of the $\mathcal{O}(n^3)$ required by the "direct" approaches, with t low due to regularization.

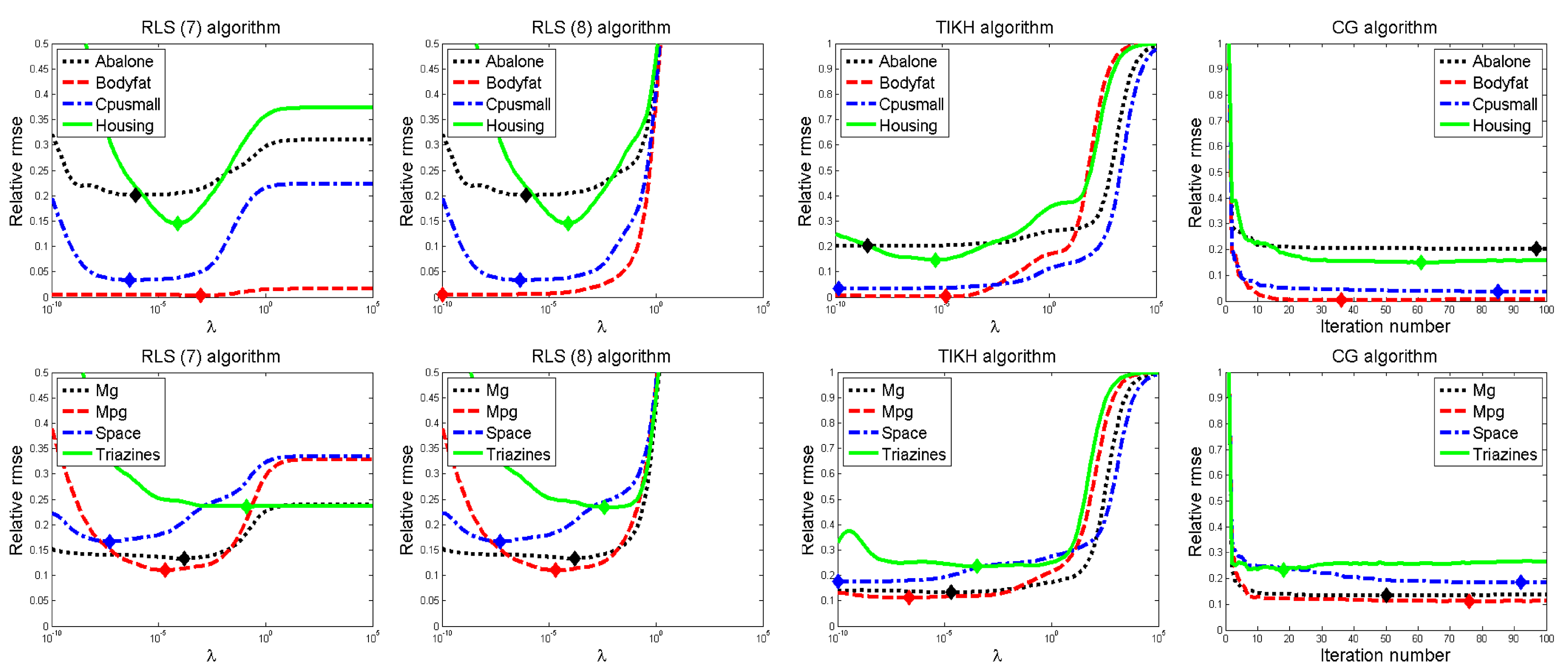
Numerical experiments

	Dataset	# examples	# features
1	Abalone	4177	8
2	Bodyfat	252	14
3	Cpusmall	8192	12
4	Housing	506	13
5	Mg	1385	6
6	Mpg	392	7
7	Space	3107	6
8	Triazines	186	60

Test settings

- 2/3 training, 1/3 test
- 100 values for λ geometrically distributed in $[10^{-10}, 10^5]$
- 50 max iterations for CG
- 5x5 grid for the parameters (C,ε) of SVM*light* [7], within $[1,500] \times [10^{-3}, 10^{-1}]$
- RLS, TIKH and CG implemented in Matlab R2010a
- PC 1.60 GHz Intel Core i7, Windows 7 environment





Figures

Top: reconstruction errors and times for the eight datasets in the case of linear and Gaussian kernels

Bottom: reconstruction errors as functions of λ or t in the case of Gaussian kernel

Conclusions

- Regularization approaches are faster than SVM
- TIKH outperforms RLS for rank-deficient kernels, while becomes heavier if the full SVD has to be computed
- CG is always the faster
- TIKH and CG provide flatter error vs λ / t curves, thus allowing a more stable construction of the solution, e.g. with cross validation.

References

[1] Hofmann T, Schölkopf B, Smola AJ 2008. Kernel methods in machine learning. Ann. Stat. 36 (3), 1171–1220
[2] Schölkopf B, Smola AJ 2002. Learning with Kernels. MIT Press, Cambridge, MA
[3] De Vito E et al 2004. Some properties of regularized kernel methods. J Mach Learn Res 5, 1363–1390
[4] Evgeniou T, Pontil M, Poggio T 2000. Regularization Networks and Support Vector Machines. Adv. Comput. Math. 13 (1), 1–50
[5] Poggio T et al 2002. B. In: Winkler J, Niranjan M Eds, Uncertainty in Geometric Computations. Kluwer Acad Publ, Dordrecht, 131–141
[6] Lavrentiev MM 1967. Some improperly posed problems of mathematical physics. Springer, New York, NY
[7] Joachims T 1999. Making large–scale SVM learning practical. In: Schölkopf B, Burges CJC, Smola AJ Eds, Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge, MA, 169–184