l_1 regularization path for functional features

Manuel Loth, Philippe Preux, INRIA, LIFL, CNRS, University of Lille, France, philippe.preux@inria.fr

Abstract: We consider the LASSO problem. Here, we propose ECON, a LARS-like algorithm that deals with parametrized features and finds their best parametrization.

1. From LARS to ECON

1.1 LARS

The LASSO problem:

- given N samples $(x_i, y_i) \in \mathcal{D} \times \mathbb{R}$, find $\hat{y} \equiv \sum_{k=0}^{k=K} w_k \phi_k$, with $\phi_j : \mathcal{D} \to \mathbb{R}$, that minimizes:

$$\sum_i (\hat{y}(x_i) - x_i)^2 + \lambda \sum_k |w_k|$$

- K is not fixed: it should be adjusted;
- value constant of regularization λ: ???

The LARS algorithm [1]:

- removes the problem of *a priori* setting the value of λ by computing the whole path of regularization, that is $\{(\lambda, w(\lambda))\}_{\lambda \in \mathbb{R}^+}$
- the l_1 regularization yields very sparse solutions (though very accurate, $\hat{y}$ is still very sparse)
- the algorithm is very efficient w.r.t. the number of potential features, $\Phi \equiv \{\phi_k\}$.
- originally formulated with $\phi_j \equiv j^{\text{th}}$ attribute; kernelized since then ($\phi_j(.) \equiv \kappa(x_j, .)$, with κ a kernel function).

Idea of LARS:

- $\lambda \leftarrow +\infty$
- set of active features: $\mathcal{A} \leftarrow \emptyset$
- set of potential features: $\mathcal{P} \leftarrow \Phi \backslash \mathcal{A}$
- compute the bias $w_0 \leftarrow \frac{1}{N} \sum_i y_i$ and set $\phi_0 \equiv \mathbf{1}$ (identity function)
- number of active features: $K \equiv |\mathcal{A}|$
- form $\hat{y} \equiv \sum_{k=0}^{k=K} w_k \phi_k$
- while stopping criteria not fullfilled
  - compute the residual $\mathbf{r}$ on the training set
  - check wether the weight of an active feature has been nullifyed; if yes, remove it from $\mathcal{A}$, and put it back in $\mathcal{P}$
  - otherwise, select $\phi_{K+1} \equiv \phi^*$ the feature among $\mathcal{P}$ which is most correlated with $\mathbf{r}$
  - compute its weight w_{K+1}, add it to $\mathcal{A}$, remove it from $\mathcal{P}$
  - update the weights of all active features, set $K++$, update λ.

Key point of the LARS: the weight w_{K+1} is set so that this newly made active feature is as much correlated with the current residual as the other active features, and **not** as much as to minimize the current residual. See (kernel) basis pursuit algorithm [2].

At a certain iteration, the change in λ, $\Delta\lambda$ is computed exactly. The minimum is found by **exhaustive** search.

Misc, but very important: some active feature may become inactive while riding the regularization path because its weight goes down to 0 (in magnitude). Such a feature leaves $\mathcal{A}$ and returns to $\mathcal{P}$.

For λ in between the λ's computed at two subsequent iterations, the weight of the active features varies linearly.

1.2 ECON

Because the minimization involved in the LARS is based on an exhaustive search, the LARS can not deal with an infinite number of features, not even with a very large one.

The features have some hyper-parameters: the ϕ's are really ϕ_θ where $\theta \in \Theta \subset \mathbb{R}^T$ is some hyper-parameters.

Usually, a finite, and small, set of hyper-parametrizations is chosen *a priori*, and the LARS is run with them.

To the opposite, ECON deals with these infinite number of potential features, and selects the best combination of features, along with their hyper-parametrizations.

The downside of ECON is that while LARS is solving exactly the minimization problem, ECON is not solving exactly the problem because of a lack of closed-form solution to this problem in general. So, we have to use a global optimizer as a heuristic to solve the problem numerically. We use DiRect to optimize this [3] which acts by dividing the domain recursively, is guaranteed to converge to the global optimum asymptotically, and the longer is run, the better the found solution.

The upside of ECON are however numerous:

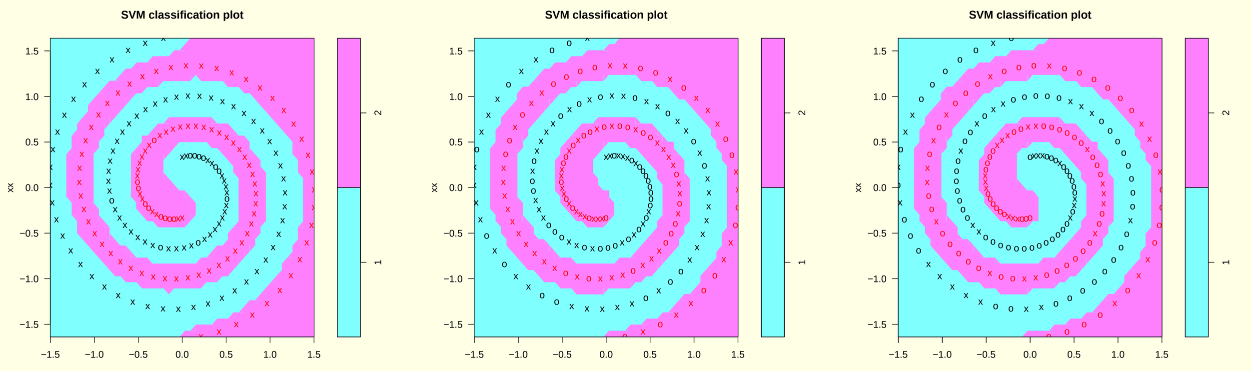
- as the LARS itself, does not ask for *a priori* selection of the constant of regularization
- ECON unique ability to select the best, or a least a good, hyper-parametrization of the features made active
- rather fast and efficient in practice
- once the kernel has been chosen, there is not parameter to hand-tune,
- produce very sparse solutions that seem to capture the complexity of the dataset quite well (saturation of K even though more and more data are acquired)

2. Experiments

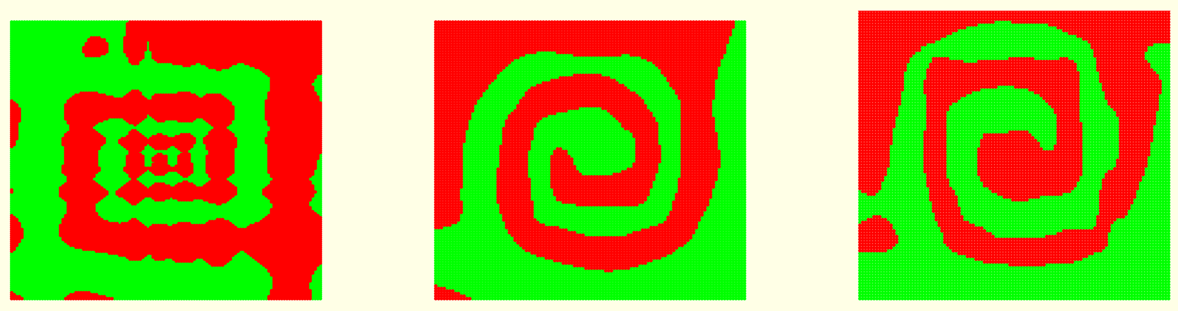
2.1 A toy classification problem

We use the 2 spirals problem, and compare a SVM to ECON. For this problem, we actually use 3 datasets: the original one made of 194 points; two others, with 500, and 1000 points, each point belonging to either one of the 2 spirals. In each dataset, half of the examples belongs to each class. We perform 300 iterations of ECON on the whole dataset.

dataset size	194	500	1000



SVM nb. SV	175	322	505



ECON nb. bases	128	97	118

- SVM: the larger the dataset, the larger the number of support vectors, the same accuracy
- ECON: number of bases remains almost constant, accuracy improves.

ECON uses only one functional kernel (2D Gaussian).

To conclude, ECON has captured the complexity of the dataset (which is fixed, whenever enough data is available — 194 is enough here), and using more data improves the accuracy. Furthermore, the solution given by ECON is sparser than the one given by SVM.

We have used `libsvm` implementation in package `e1071` in R, with automatic selection of the "best" model.

2.2 Friedman's functions

Friedman's function	SVM	RVM	[5]	ECON	
	ave.MSE	ave.MSE	ave.MSE	ave.MSE	Median MSE
F1	$2.92/116.2$	$2.80/59.4$	$2.84/73.5$	$1.99/76$	$1.97/71$
F2	$4140/110.3$	$3505/6.9$	$3808/14.2$	$4845/54$	$4696/55$
F3	$0.0202/106.5$	$0.0164/11.5$	$0.0192/16.4$	$0.0115/53$	$0.0113/53$

Friedman's functions: F1 has $P = 10$ attributes, the domain being $\mathcal{D} = [0, 1]^{10}$. The function is noisy, and 5 of the attributes are useless. F2 and F3 are defined on some subset of $\mathbb{R}^4$ and are also noisy. For each of these problems, we generate training sets made of 240 examples, and the same test set of 1000 data. For these problems, we compare our results with those obtained by [5], on a LASSO algorithm, and also reported for Support Vector Machine, and Relevance Vector Machine. We measure the mean-squared error on the data-set. As ECON computes the whole regularization path, we compare the results obtained, in the same experimental settings, by ECON and the one proposed by [5] who obtains a solution with a certain number of terms in $\hat{y}$ (K).

On F1, ECON by far obtains the best results among the 4 algorithms. Only 17 terms provide an accuracy better than 2.8, the best accuracy mentioned for the other 3 algorithms. On F2, the results are less good, but are still rather good. To stick to the results published in [5], the figures in the table are averages; however, if we consider the best accuracy, or better, the histogram of accuracies, it is skewed towards better accuracies.

2.3 Larger regression problems

We compare the performance of ECON with those published in [5] again. For Abalone, we provide the accuracy measured on a test set of 1000 data / the number of terms (aka, the number of support vectors) in $\hat{y}$, averaged on 100 runs for ECON. For house-price-8L, the training set is successively made of 1000, 2000, and 4000 data, the test set of 4000 data.

dataset	SVM	RVM	[5]	ECON
Abalone	$4.37/972$	$4.35/12$	$4.35/19$	$4.31/71$, $4.30/100$
house-price-8L, 1K	$1.062/597$	$1.099/33$	$1.075/61$	$0.57/20$
house-price-8L, 2K	$1.022/1307$	$1.048/36$	$1.054/63$	$0.41/38$
house-price-8L, 4K	$1.012/2592$	NA	$1.024/69$	$0.40/40$

3. Conclusion

- ECON computes the regularization path of the LASSO problem, using functional features
- ECON provides very sparse expansions, yet yielding highly accurate estimators
- ECON is very easy to use: no parameter to hand-tune

4. Future work

- investigate further the global optimization
- hybrid it with a gradient descent

References

[1] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least-angle regression. *Annals of statistics*, 32(2):407–499, 2004.

[2] V. Guigue, A. Rakatomamonjy, and S. Canu. Kernel basis pursuit. In *Proc. ECML*, volume 3720, pages 146–157. Springer, LNAI, 2005.

[3] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Aplications*, 79(1):157–181, October 1993.

[4] M. Loth and Ph. Preux. The equi-correlation network: a new kernelized-lars with automatic kernel parameters tuning. Research Report RR-6794, INRIA, January 2009.

[5] V.Roth. Generalizd lasso. *IEEE Trans. on Neural Networks*, 15(1):16–28, January 2004.