22nd International Conference on Artificial Neural Networks, Lausanne, 2012

DOI: https://doi.org/10.1007/978-3-642-33266-1

Sparse Linear Wind Farm Energy Forecast

Carlos M. Alaíz, Alberto Torres, and José R. Dorronsoro

Departamento de Ingeniería Informática & Instituto de Ingeniería del Conocimiento,
Universidad Autónoma de Madrid, 28049 Madrid, Spain
carlos.alaiz@uam.es,alberto.torres@iic.uam.es,jose.dorronsoro@uam.es

Abstract. In this work we will apply sparse linear regression methods to forecast wind farm energy production using numerical weather prediction (NWP) features over several pressure levels, a problem where pattern dimension can become very large. We shall place sparse regression in the context of proximal optimization, which we shall briefly review, and we shall show how sparse methods outperform other models while at the same time shedding light on the most relevant NWP features and on their predictive structure.

Keywords: Sparse Methods, Lasso, Group Lasso, Elastic–Net, Group Elastic–Net, Wind Farm Energy Forecast

1 Introduction

Most modelling problems of interest in practical machine learning involve high dimensional data, sometimes coupled with large samples. The overall goal in those problems is to achieve good predictive models but large sizes and dimensions often preclude the straight use of strong but complex methods such as neural networks or support vector machines. The simplest choice is to use linear models, that while probably inferior to other alternatives, offer a first quality standard against other approaches that can be benchmarked. Moreover, linear models can also help to pinpoint which variables are most influential. This can be exploited by selecting the most important features to be used as inputs of stronger methods and, also, to gain further knowledge of the problem under study.

In the past decade the Stanford school of Breiman, Friedman, Hastie and Tibshirani has proposed a series of sparse enforcing linear models, such as the Lasso [8], Group Lasso [9], and Elastic–Net [10]. They add to the standard square error loss function an ℓ_1 penalty (Lasso), a mixed $\ell_{2,1}$ penalty (Group Lasso) or combine the ℓ_1 penalty with an ℓ_2 regularizer term (Elastic–Net). The resulting optimization problems are still convex but no longer differentiable, and a series of ad–hoc methods have been proposed to solve them. From a general point of view the combined criterion function has the form

$$\min_W \quad \left\{ \frac{1}{2}\hat{L}_{\mathcal{S}}(W) + \lambda_1 \hat{R}(W) + \frac{\lambda_2}{2}\|W\|_2^2 \right\} \ , \tag{1}$$

with $\hat{L}_{\mathcal{S}}(W)$ denoting the quadratic loss of a linear model with weights W over a sample $\mathcal{S}$ and $\hat{R}(W)$ the non–differentiable but still convex ℓ_1 or $\ell_{2,1}$ norms of W. This general formulation places the above problems under the scope of Proximal Optimization [3] which exploits the concept of proximal operators to arrive to a general optimization procedure. Moreover, it makes quite easy to extend the previous methods. For instance we will consider here a group version of Elastic–Net simply by mixing in (1) the $\ell_{2,1}$ and ℓ_2 norms of W. We shall apply this set up to the problem of predicting wind farm energy production, of considerable interest nowadays and fitting squarely in the previous set up.

The usual approach uses historical wind energy production data and forecasts derived from numerical weather prediction (NWP) systems, in our case, the Agencia Española de Meteorología (AEMET; [1]). These systems provide forecasts for the nodes of a geographical grid, typically at resolutions that start at 0.16° degrees or even finer, for either a surface level derived from a smooth orographical model or for several constant pressure levels that go from sea level to a height of about 20 Km. These forecasts are usually given every three hours. Moreover, many meteorological variables are available at each level and the number of possible features may clearly become very large. To manage this, a first obvious approach is to fix a square of grid points centred at the wind farm and consider for each grid point a number of surface variables. However, these are given typically 10 m above the grid point height, but that may bear little relationship with the actual altitude of a wind farm. The alternative is to consider NWP forecasts for a number of pressure levels but this will of course augment feature dimension and, thus, make the sparse methods attractive modelling tools.

This approach will be applied to the study of wind energy forecasting at the Sotavento wind farm, situated in the Galicia region of north–western Spain. We shall work with a 6×6 grid with a 0.25° degree resolution, 6 pressure layers and 5 meteorological variables. Total dimension is thus $1,080$. We will consider a one–year long training sample, whose size is thus $2,920$, i.e., about three times pattern dimension and well below the linear regression rule of thumb of having 10 patterns per dimension. Regularization is thus mandatory but sparse regression also comes in very naturally. In fact, and as we shall see, sparse models beat ridge regression using a quite small number of the features available. Moreover, sparse models also shed light on the predictive structure of NWP variables, something that could be exploited when considering stronger, more complex methods than standard regression. The paper is organized as follows. In Sect. 2 we will briefly review the theory of Proximal Optimization and its training algorithms, and describe how the previous sparse regression problems fit in this set up. In Sect. 3 these models will be applied to wind energy prediction for the Sotavento farm and the paper ends with a discussion and conclusions section.

2 Proximal Methods for Regularized Linear Models

Assume a training set $\mathcal{S} = \left\{(X^{(p)}, y^{(p)})\right\}_{p=1}^{N}$ with $(X^{(p)}, y^{(p)}) \in \mathbb{R}^D \times \mathbb{R}$, for which we want to build a model $f_W : \mathbb{R}^D \to \mathbb{R}$, in a certain Hilbert space $f_W \in \mathcal{H}$

and parametrized by a weight vector W, such that $f_W(X^{(p)}) \approx y^{(p)}, \forall p$. To make this more precise, we introduce a convex loss function $L_{\mathcal{S}} : \mathcal{H} \to \mathbb{R}$ and look in principle for a f_W^* that minimizes $L_{\mathcal{S}}(f_W)$. However, we may also want to control model complexity for which we may introduce a sparsity controlling convex term $R(f_W)$ as well as a regularization term $\|f_W\|_{\mathcal{H}}^2$. These considerations lead to the general optimization problem

$$\min_{f_W \in \mathcal{H}} \quad \left\{ \frac{1}{2} L_{\mathcal{S}}(f_W) + \lambda_1 R(f_W) + \frac{\lambda_2}{2} \|f_W\|_{\mathcal{H}}^2 \right\} \ , \tag{2}$$

where λ_1 and λ_2 are the parameters which will determine the relative importance of the regularization terms against the error term. Notice that if $\lambda_2 \neq 0$, the objective function is strictly convex. While the first and third terms are differentiable, $R(f_W)$ will usually not be so. To deal with this, we will consider the problem (2) under the framework of Proximal Methods, a set of techniques to solve non–differentiable optimization problems in an iterative way. The starting point is the fact [6] that the solution f_W^* of (2) satisfies for any $\eta > 0$ the fixed point equation

$$f_W^* = \text{prox}_{\frac{\lambda_1}{\eta};R} \left(\left(1 - \frac{\lambda_2}{\eta} \right) f_W^* - \frac{1}{2\eta} \nabla L_{\mathcal{S}}(f_W^*) \right) \ , \tag{3}$$

where the proximity operator $\text{prox}_{\lambda;F}(x)$ of a function F at a point $x \in \mathbb{R}^D$ with step λ is defined as

$$\text{prox}_{\lambda;F}(x) = \arg\min_y \quad \left\{ \lambda F(y) + \frac{1}{2} \|x - y\|_2^2 \right\} \ . \tag{4}$$

The solution of (4) is problem dependent and finding it is the main issue when applying proximal optimization. If it is known, (3) justifies an iterative algorithm based on the steps

$$f_W^{(t)} = \text{prox}_{\frac{\lambda_1}{\eta_t};R} \left(\left(1 - \frac{\lambda_2}{\eta_t} \right) f_W^{(t-1)} - \frac{1}{2\eta_t} \nabla L_{\mathcal{S}}(f_W^{(t-1)}) \right) \ .$$

There are several general purpose algorithms that apply this iterative scheme. Here we will use the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA; [2]) which automatically determines the step length η_t.

Sparse regularized linear regression fits nicely in this set up. In that case, $\mathcal{H} = \mathbb{R}^D$, the basic model is $f_W(X) = X \cdot W$ and the loss function is $L_{\mathcal{S}}(f_W) = \frac{1}{N} \|\mathcal{X}W - Y\|_2^2$, where $\mathcal{X}$ is the matrix collecting all the inputs $X^{(p)}$ in its rows, and Y is the vector of all the desired outputs $y^{(p)}$. All the mentioned linear sparse models can be derived for particular choices of the functional R and the parameters λ_1 and λ_2, as summarized in Table 1. The simplest case is to fix $\lambda_1 = \lambda_2 = 0$, which leads to the Ordinary Least Squares (OLS) model. The resulting optimization problem can be easily solved analytically (see Table 1), but if no regularization is included, OLS models are likely to over–fit the sample when the feature dimension D is comparable with sample size. The simplest way

Table 1: Correspondence between the regularized linear models and problem (2).

Model	$\mathbf{L_S(f_W)}$	$\mathbf{R(f_W)}$	$\mathbf{\|f_W\|^2_{\mathcal{H}}}$	**Solution**
OLS	$\frac{1}{N}\|\mathcal{X}W - Y\|_2^2$	$\times$	$\times$	$W^o = (\mathcal{X}^T\mathcal{X})^{-1}\mathcal{X}^T Y$
RLS	$\frac{1}{N}\|\mathcal{X}W - Y\|_2^2$	$\times$	$\frac{1}{D}\|W\|_2^2$	$W^o = (\mathcal{X}^T\mathcal{X} + \frac{N\lambda_2}{D})^{-1}\mathcal{X}^T Y$
LA	$\frac{1}{N}\|\mathcal{X}W - Y\|_2^2$	$\frac{1}{D}\|W\|_1$	$\times$	FISTA
GL	$\frac{1}{N}\|\mathcal{X}W - Y\|_2^2$	$\frac{1}{D}\|W\|_{2,1}$	$\times$	FISTA
ENet	$\frac{1}{N}\|\mathcal{X}W - Y\|_2^2$	$\frac{1}{D}\|W\|_1$	$\frac{1}{D}\|W\|_2^2$	FISTA
GENet	$\frac{1}{N}\|\mathcal{X}W - Y\|_2^2$	$\frac{1}{D}\|W\|_{2,1}$	$\frac{1}{D}\|W\|_2^2$	FISTA

to avoid this is just to take some $\lambda_2 > 0$ while keeping $\lambda_1 = 0$. This leads to Regularized Least Squares (RLS; [4]) which has also a closed form solution.

A first choice for the functional R is to use the ℓ_1 norm, $R_{\mathrm{LA}}(f_W) = \frac{1}{D}\|W\|_1$. Setting $\lambda_1 > 0$ and $\lambda_2 = 0$ and using this functional, we recover the Lasso (LA; [8]) algorithm. The ℓ_1 norm encourages sparse models, something that can be seen as an implicit feature selection, because the inputs associated with zero coefficients are just discarded. Because of its non–differentiability, LA models will be trained using FISTA, as explained above. The proximal operator for the ℓ_1 norm is given by soft thresholding [5] as $(\mathrm{prox}_{\lambda;\|\cdot\|_1}(x))_i = x_i(1 - \frac{\lambda}{|x_i|})^+$. Notice that in LA all coefficients are treated individually. In certain circumstances we may want to have a grouping effect in the features, so as to detect relevant groups. A way to achieve this is to enforce that all the coefficients in a group should be active or inactive at the same time. This is what the Group Lasso (GL; [9]) algorithm obtains using a mixed $\ell_{2,1}$ norm as regularizer, i.e., $R_{\mathrm{GL}}(f_W) = \frac{1}{D}\|W\|_{2,1} = \frac{1}{D}\sum_{i=1}^{\frac{D}{V}}\sqrt{\sum_{v=1}^{V} w_{i,v}^2}$, where $w_{i,v}$ is the component corresponding to the v–th variable of the i–th group, and the space $\mathbb{R}^D$ is decomposed in $\frac{D}{V}$ groups of V variables. Notice that this $\ell_{2,1}$ norm is just the ℓ_1 norm of the ℓ_2 group norms; thus it precisely enforces group sparsity. Again, this problem will be solved using FISTA with the proximal operator being now $(\mathrm{prox}_{\lambda;f_1}(x))_i = x_{i,v}(1 - \frac{\lambda}{\|x_{i,\cdot}\|_2})^+$.

While the ℓ_1 norm enforces sparseness, the regularizing effect of the ℓ_2 norm has its own advantages and combining both seems sensible. This is what Elastic–Net (ENet; [10]) does, setting $\lambda_1 > 0, \lambda_2 > 0$ and using the R_{LA} regularizer of LA. On the other hand, the Proximal Optimization approach easily allows to define a group version of this, the Group Elastic–Net (GENet), that combines the ℓ_2 norm with the R_{GL} regularizer. Both ENet and GENet will be also solved by FISTA.

3 Numerical Experiments

In this section we will apply the previous algorithms to study the prediction of the energy production of a wind farm. We will work with the Sotavento wind farm [7], located at 43.34° N, 7.86° W and that makes production data publicly

available. The usual features in wind power forecasting are surface predictions of meteorological variables. We shall work first with the following: V, the norm of the wind speed, V_x, its x component, V_y, its y component, the temperature T and pressure P. They will be considered over a rectangular, 0.25° resolution, 6×6 point grid surrounding Sotavento. The dimension for surface prediction is thus $180 = 6 \times 6 \times 5$, large but not too much so. We will work with a 1 year training set and a 2 months test period. Meteorological forecast are only available every three hours; thus, we have eight patterns per day and training sample size is $2,920$. We normalize the wind energy production target values to the $[0, 1]$ interval as percentages of the total installed wind power in Sotavento.

In any case, many more variables are available on the 17 constant pressure levels for which AEMET gives NWP forecasts, although obviously not all of them will have an effect on the energy. These levels have a 50 hPa resolution and over them pressure is constant and no longer a predictive variable; we substitute it by geopotential heights. A first level selection can be done using Sotavento's elevation, with an average of about 600 m. The first 11 levels are consistently located much higher; moreover, correlation plots with wind farm production (not included) show that they do not contain useful information and we have discarded them outright. We are left with the lowest 6 layers and total feature dimension is then $6 \times 6 \times 6 \times 5 = 1,080$. Thus, sample size is about 3 times the dimension. However, it is not clear that all of these features have the same effect (if any) on the wind energy production and they may handicap full regression models even if they are regularized. Sparse methods can thus help us first to find better models and, second, to better understand which grid points, pressure levels and variables are the most useful to improve predictions.

To do so, we will use the models described in Sect. 2. For the case of group algorithms (GL and GENet), we consider as a group the 5 meteorological variables evaluated over a grid point. As usually done in wind energy, the models are evaluated using the Mean Absolute Error over the test set, $\mathrm{MAE} = \frac{1}{N'} \sum_{p=1}^{N'} |X^{(p)} \cdot W - y^{(p)}|$. We will also report the standard deviation σ_{AE} of the absolute errors, although they are rather conservative as we do not perform any sample size correction (assuming independence for these errors would lead to divide the values given by $\sqrt{N'}$ and, hence, much smaller values). An important issue for most of the algorithms used is the estimation of the hyper–parameters λ_1 and λ_2 that configure each model. This is done as a search over a grid representation of the parameter space, working on a logarithmic scale from 10^{-3} to 10^3 with steps of $10^{0.10}$. For the algorithms that involve a bi–dimensional grid (ENet and GENet), the step size is increased to $10^{0.20}$. At each point of the parameter grid, a 5–fold cross validations is used to evaluate a given model using as fitness the MAE and discarding models above a predefined sparseness level ρ, fixed as the percentage of non–zero weights. Three different values of ρ, 30, 50 and 100 (i.e., no restrictions), are considered for all models.

The comparison of the different algorithm is summarized in Table 2. A first reference are surface models, for which we recall that feature dimension is 180; therefore, we only consider the 100% sparsity level. Their performance is given

Table 2: Results and parameters for 6 pressure levels and minimum 30%, 50% and 100% sparseness, and surface data. Models ordered by MAE.

(a) 6 levels ($\rho = 30\%$).

Method	MAE	σ_{AE}	Act W	λ_1	λ_2
ENet	7.09	6.5	011.7%	−0.40	−2.80
RLS	7.11	6.6	100.0%	×	+1.20
LA	7.14	6.5	010.9%	−0.30	×
GENet	7.52	6.7	017.1%	+0.20	−2.00
GL	7.61	6.8	013.9%	+0.30	×
OLS	8.65	8.6	100.0%	×	×

(b) 6 levels ($\rho = 50\%$).

Method	MAE	σ_{AE}	Act W	λ_1	λ_2
LA	7.08	6.5	014.4%	−0.60	×
ENet	7.10	6.5	016.8%	−0.60	+0.00
RLS	7.11	6.6	100.0%	×	+1.20
GENet	7.31	6.6	023.6%	+0.00	+0.00
GL	7.32	6.6	022.7%	+0.00	×
OLS	8.65	8.6	100.0%	×	×

(c) 6 levels ($\rho = 100\%$).

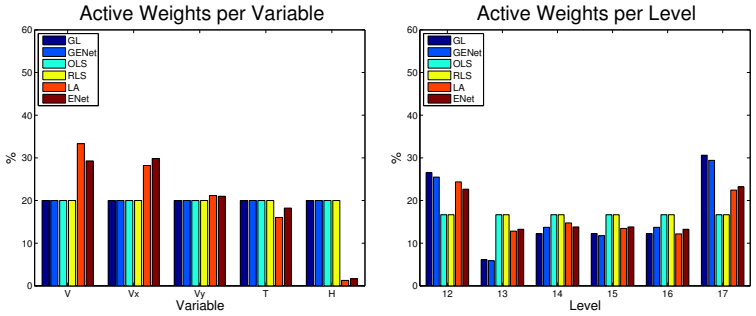
Method	MAE	σ_{AE}	Act W	λ_1	λ_2
GL	7.05	6.5	053.2%	−0.60	×
ENet	7.07	6.5	039.6%	−1.00	+0.60
LA	7.10	6.6	034.4%	−1.00	×
GENet	7.11	6.5	075.5%	−0.60	+0.80
RLS	7.11	6.6	100.0%	×	+1.20
OLS	8.65	8.6	100.0%	×	×

(d) Surface ($\rho = 100\%$).

Method	MAE	σ_{AE}	Act W	λ_1	λ_2
GL	7.12	6.7	100.0%	−2.80	×
LA	7.13	6.7	091.7%	−2.80	×
RLS	7.14	6.7	100.0%	×	−2.80
OLS	7.21	6.8	100.0%	×	×
GENet	7.26	6.6	100.0%	−2.00	−2.60
ENet	7.32	6.6	066.7%	−2.20	−1.80

in Subtable 2d and the best models are GL and LA in that order, although they essentially do not achieve any sparsity enforcing, as their active weights are 100% and 91.7% respectively. Subtables 2a, 2b and 2c give the performance of the multi–level variables. Recall that feature dimension is now $1,080$, making mandatory the use of regularized or sparse models (notice that unregularized linear regression OLS performs very badly due to a clear case of over–fitting). As it can be expected, the best results are obtained at the 100% sparsity level, with the best algorithm being GL that uses about half of the features. If more sparsity is imposed, model performance is just slightly worse, but sparsity greatly increases. At the 50% level, LA is the second best model, with only 14.4% of active weights. At the strictest 30% sparsity level, ENet is the best model, with a sparsity of just 11.7%. Moreover, in all cases the results using pressure level variables are better than the ones obtained using surface variables. As mentioned before, a reason for this is that the 10 m height of surface variables may not be representative of actual wind farm altitude. In any case, the use of sparse methods such as LA and ENet over pressure layers is justified.

We turn now our attention to the structure identified by the sparse models. Figure 1a shows the percentage of the total active weights per variable. The non–sparse algorithms obviously do not perform any kind of variable selection and the same is true for the group methods, the reason being that they essentially select all the variables at a given grid point. On the other hand, LA and ENet favour the V and V_x variables and discard almost completely the geopotential height. This is reasonable as it has a much smaller correlation with respect



(a) Active weights per variable. (b) Active weights per level.

Fig. 1: Active weight % per variable (left) and level (right) for $\rho = 50\%$.

to wind energy production. Figure 1b shows the percentage of the total active weights per pressure level. Now all the sparse methods perform some kind of level selection, favouring the highest and lowest layers. The reason for this is clear, as all levels have high correlations with respect to wind energy while the extreme levels are the most independent. This effect is particularly strong for the group models GL and GENet, as they must focus on actually selecting levels instead of variables. We also point out that sparse methods define some grid structure as they select points which are mostly located in either the centre of the grid (closest to the wind farm) or the grid extremes (points least correlated with the grid centre but still correlated with the wind energy).

Summing things up, it is clear that taking into account different pressure levels yields better models than considering only surface variables. Sparse models help on this and, moreover, automatically select the feature structure better suited for modelling.

4 Conclusions

Most modelling problems of interest in practical machine learning involve high dimensional data. The goal in these problems is not only to achieve good predictive models but also to do so in an economic way using as few features as possible and, moreover, to identify some structure in the predictive variables. A natural approach to this task is to use linear regression models upon which sparsity is enforced; this is the case of the Lasso, Group Lasso and Elastic–Net methods, to which we have added a group version of Elastic–Net. In this work, we have reviewed them under the unifying point of view of Proximal Optimization. This makes possible to apply efficient algorithms and to consider extensions of previous models, as it is the case of the Group Elastic–Net model. Wind energy prediction is a natural field of application for these methods, as NWP makes available a large number of predictive variables. We have analyzed wind energy

prediction for the Sotavento wind farm, located in the Galicia region of north–western Spain, considering NWP values for points in a 6×6 grid over 6 different pressure layers. As our results show, sparse models built over several pressure layers outperform those built using just NWP surface values, even when a strict degree of sparsity is required. Moreover, sparse models also identify the predictive structure in the NWP features and discriminate among the levels considered, thus improving our problem understanding.

In any case, stronger models could clearly yield better predictions, which makes natural to exploit sparse linear regression to select the most relevant features upon which more advanced models can be built. For example, better models can be obtained using standard RLS over features selected by the Lasso. Moreover, the sparse linear methods also have strong theoretical foundations that could be brought to bear on feature selection. We are currently studying these and other related issues.

Acknowledgement. With partial support from grant TIN2010-21575-C02-01 of Spain's Ministerio de Economía y Competitividad and the UAM–ADIC Chair for Machine Learning in Modelling and Prediction. The first author is supported by the FPU–MEC grant AP2008-00167. We thank our colleague Álvaro Barbero for the software used in this work.

References

1. Agencia española de meteorología (2012), http://www.aemet.es
2. Beck, A., Teboulle, M.: A fast iterative shrinkage–thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences 2(1), 183–202 (2009)
3. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. Recherche 49, 1–25 (2009)
4. Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12(12), 55–67 (1970)
5. Kowalski, M., Torrésani, B.: Structured sparsity: from mixed norms to structured shrinkage. In: Gribonval, R. (ed.) SPARS'09 – Signal Processing with Adaptive Sparse Structured Representations. Inria Rennes – Bretagne Atlantique, Saint Malo, France (2009)
6. Mosci, S., Rosasco, L., Santoro, M., Verri, A., Villa, S.: Solving structured sparsity regularization with proximal methods. In: ECML/PKDD (2). pp. 418–433. Berlin, Heidelberg (2010)
7. Sotavento (2012), http://www.sotaventogalicia.com
8. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc. Ser. B 58(1), 267–288 (1996)
9. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society – Series B: Statistical Methodology 68(1), 49–67 (2006)
10. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society – Series B: Statistical Methodology 67(2), 301–320 (2005)