Statistics of Big Data, Fall 2023-24

Class notes 6

Sources for today's material:
Chapters 2-3 of the Elements of Statistical Learning by Hastie et al.
Review by Rosset (2013) on practical sparse modeling
Least Angle Regression by Efron et al.(2004)
Candes et al. (2005)
Meinshausen and Yu (2009)

Compressed sensing

Let's start with a problem formulation from signal processing (SP) rather than statistics/ML, in the following setup:

1. We have a large basis/dictionary of size p, that is as usual $X_i \in \mathbb{R}^p$
2. Our signal is an exact linear function of X, that is $Y = X^T\beta$
3. β is a sparse vector with $\|\beta\|_0 \leq k <<< p$

In this setting, if we observe $n = p$ training observations, then regardless of sparsity we can solve n equations with $p = n$ unknowns $\mathbb{Y} = \mathbb{X}\beta$ and get the true β (assuming $\mathbb{X}$ is full rank). Having $n << p$ observations seems hopeless, but it is not because we also know there is sparsity.

Naively, if we observe only $n = k$ training observations, then for the correct set of k coordinates, denote it I, solving: $\mathbb{Y} = \mathbb{X}_{\cdot I}\beta_I$ will give the correct solution, and all other subsets of size k will not give a solution! So we can solve this using only $k <<< p$ observations, at the cost of having to try an exponential $\binom{p}{k}$ number of possible models.

The fundamental result of compressed sensing is that with $n \approx k$ (a bit bigger), we can solve the following single *convex* problem in p-dimensional space:

$$\min \|\beta\|_1 \text{ s.t.} \|\mathbb{Y} - \mathbb{X}\beta\| = 0,$$

(that is, finding the minimum ℓ_1 norm interpolator) will give the correct solution with k-sparsity with high probability. In other words, using ℓ_1 efficiently solves the problem that can be only impractically solved with ℓ_0.

This is the initial statement in Donoho and Candes and Tao's work that rocked the world about 15 years ago.

To adjust this to our setting of interest, we have to get rid of assumptions $1, 2$ above: deal with having noise, and not having a predetermined basis (or control of $\mathbb{X}$ as sometimes assumed in SP).

The next step is adding noise so that $\mathbb{Y} = \mathbb{X}\beta + \epsilon$, while still controlling $\mathbb{X}$. Candes et al. showed that if we get to choose $\mathbb{X}$ in a smart way and $n = O(k \log p)$ then solving:

$$\min \|\beta\|_1 \text{ s.t.} \|\mathbb{Y} - \mathbb{X}\beta\|_2 \leq \epsilon_0,$$

will give the correct sparsity pattern (and close to accurate values of β) with high probability.

Note that this is already a Lasso formulation exactly, through the Lagrange form:

$$\hat{\beta} = \min \|\beta\|_1 \text{ s.t.} \|\mathbb{Y} - \mathbb{X}\beta\|_2 \leq \epsilon_0 \quad \Leftrightarrow \quad \hat{\beta} = \min \|\mathbb{Y} - \mathbb{X}\beta\|_2 \text{ s.t.} \leq \|\beta\|_1 \leq \delta_0,$$

for some mapping $\epsilon_0 \Leftrightarrow \delta_0$ (that is problem dependent, but exists).

However, we are focused on also not controlling $\mathbb{X}$ but observing it, in addition to also having noise. It turns out that also in this setting we can replace controlling $\mathbb{X}$ with making assumptions on $\mathbb{X}$ and still be able to obtain pretty strong results. Our setting of interest here:

1. $\mathbb{Y} = \mathbb{X}\beta + \epsilon$
2. $\|\beta\|_0 = k <<< p$
3. An additional assumption that the columns of $\mathbb{X}$ are *weakly correlated.* This assumption has different names in different papers: *Irrepresentability* in Meinshausen and Yu (2009), *incoherence* in Candes and Plan (2009) etc. It essentially assumes that the k non-zero columns of $\mathbb{X}$ have low correlation or inner product with the $p - k$ columns that correspond to zero coordinates of β.

Under these assumptions they were able to prove that if we only observe $n = O(k \log p) << p$ observations, there is a Lasso solution that gives the correct sparsity pattern with high probability:

$$\hat{\beta} = \|\mathbb{Y} - \mathbb{X}\beta\|_2^2 + \lambda_0\|\beta\|_1 \quad \Rightarrow \quad \Big|\{j : \beta_j \neq 0\} \triangle \{j : \hat{\beta}_j \neq 0\}\Big| = 0 \text{ w.p. } 1 - \delta.$$

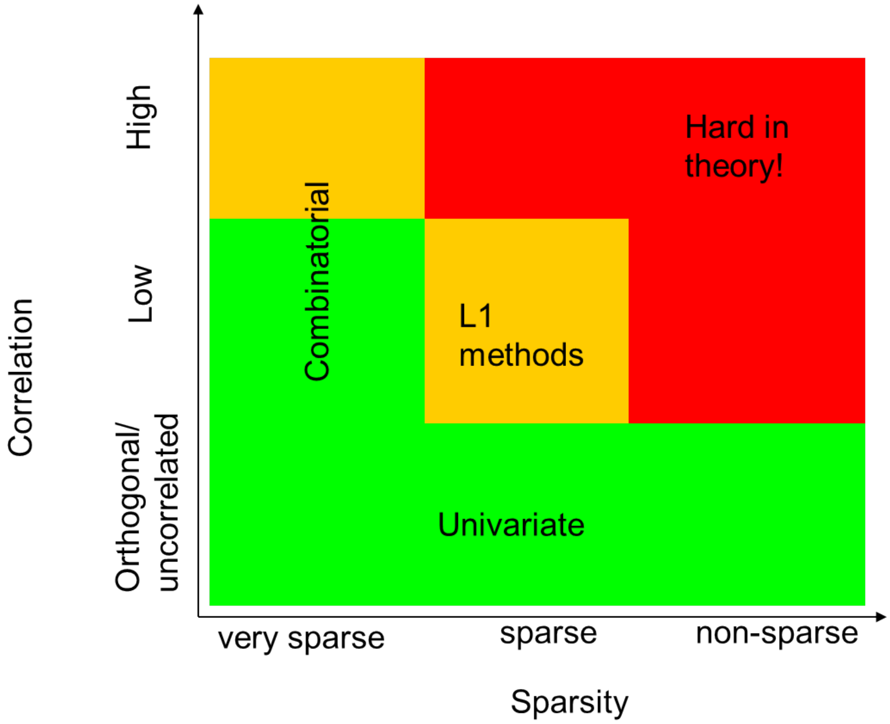
The importance of the last condition on incoherence is clear: if columns j_1, j_2 are too similar it is inherently hard to determine if $\beta_{j_1} \neq 0$ or $\beta_{j_2} \neq 0$, so keeping the correlation low is critical to be able to use the compressed sensing paradigm in predictive modeling! Remember we still have to find a needle of size k in a haystack of size p.

Summary of sparsity modeling discussion

We have discussed various methods for dealing with $n << p$ under various assumptions. In general we have to ask ourselves two main questions:

- How sparse is the true model? If $k \approx 1$ is extremely sparse, we can still hope to do a combinatorial search over all $\binom{p}{k}$ possible models and solve the ℓ_0 problem. If $k <<< p$ but beyond combinatorial search we have to add assumptions on correlation.
- How correleated/coherent etc. are the columns of $\mathbb{X}$? If they are essentially uncorrelated, then we can use marginal regression as we discussed last week (and then sparsity is less critical). If they have low correlation ("incoherence") then sparsity is critical and we are in the sweet spot of compressed sensing with $n = O(k \log p)$ observations.

This is summarized in the following plot from Rosset (2013):



Back to Lasso: Statistical properties and computation

The lasso formulations:

$$\hat{\beta}^{pen}(\lambda) = \arg\min_{\beta} RSS(\beta) + \lambda \sum_j |\beta_j| \ , \ \ \hat{\beta}^{con}(s) = \arg \min_{\beta : \sum_j |\beta_j| \leq s} RSS(\beta).$$

Unlike ridge it does not have an algebraic solution (for example, the penalized Lagrange version is not differentiable). A key observation is that this is now a *quadratic programming* (QP) problem, with quadratic objective and linear constraints. This can be seen from the constrained version, which can equivalently be written as:

$$\begin{array}{ll} \min & RSS((\beta^+ - \beta^-) \\ \text{s.t.} & \sum_{j=1^p} \beta_j^+ + \beta_j^- \leq s \\ & \beta_j^-, \beta_j^+ \geq 0 \forall j, \end{array}$$

(the problems are equivalent since in the optimal solution, it is guaranteed that either $\beta_j^+ = 0 \Rightarrow \beta_j = -\beta_j^-$ or $\beta_j^- = 0 \Rightarrow \beta_j = \beta_j^+$.)
Since QP is a standard problem in convex optimization, standard solvers can be used for Lasso (and in fact the original paper by Tibshirani(1996) proposes a special QP variant that fits the structure of this problem).

However, in the early 2000's several groups realized that the problem can be solved with linear algebra tools, by following the set of solutions to the penalized problems $\hat{\beta}^{pen}(\lambda), 0 \leq \lambda < \infty$. We will present this approach, best known by the name Least Angle Regression (LARS), as it is interesting both for computation and statistical interpretation.

Define for simplicity of notation $PRSS(\beta) = RSS(\beta) + \lambda \sum_j |\beta_j|$ the penalized objective. Note that if we evaluate $PRSSS(\beta)$ at a value where $\beta_k \neq 0$ then the function is differentiable in the kth coordinate:

$$\begin{aligned} \left. \frac{\partial PRSS(\beta)}{\partial \beta_k} \right|_{\beta_k > 0} &= \underbrace{-2 \sum_i X_{ik}(Y_i - \sum_j X_{ij}\beta_j)}_{L_k(\beta)} + \lambda \\ \left. \frac{\partial PRSS(\beta)}{\partial \beta_k} \right|_{\beta_k < 0} &= L_k(\beta) - \lambda \end{aligned}$$

This already means that if $\hat{\beta}(\lambda)$ is an optimal solution and $\hat{\beta}_k > 0$ we have:

$$\left. \frac{\partial PRSS(\beta)}{\partial \beta_k} \right|_{\beta = \hat{\beta}} = L_k(\hat{\beta}) + \lambda = 0 \ \ \Rightarrow \ \ \mathbb{X}_{\cdot k}^T (\mathbb{Y} - \mathbb{X}\beta) = \frac{\lambda}{2},$$

and similar with opposite sign for the case $\hat{\beta}_k < 0$.

What happens if $\beta_k = 0$? This is a non-differentiability point of the penalty, however we know what happens to the derivative if we go either left or right, it will attain one of the values above, in other words we can write informally the sub-differential formula:

$$-\lambda \leq L_k(\hat{\beta}) \leq \lambda.$$

Since in the optimal solution the sub-differential contains 0, we know that by definition if $\hat{\beta}_k = 0$ it implies:

$$L_k(\hat{\beta}) - \lambda \leq 0 \leq L_k(\hat{\beta}) + \lambda \quad \Rightarrow \quad |L_k| \leq \lambda.$$

Note that $|L_k| > \lambda$ is not possible for an optimal solution (makes sense — it means that we can gain a lot from changing this coordinate, more than the penalty cost).

Summarizing the optimal solution conditions:

$$\begin{aligned} |\hat{\beta}(\lambda)_k| > 0 \quad &: \quad \mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}\beta) = \frac{\lambda}{2} sgn(\hat{\beta}(\lambda)_k) && (1) \\ |\hat{\beta}(\lambda)_k| = 0 \quad &: \quad |\mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}\beta)| \leq \frac{\lambda}{2} && (2) \\ & \quad\; |\mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}\beta)| > \frac{\lambda}{2} \text{ is impossible} && (3) \end{aligned}$$

For those who learned optimization, this will seem familiar as the stationarity + complementary slackness KKT conditions.

Now, we want to use this understanding to find an efficient way to "track" the set of solutions $\left\{\hat{\beta}(\lambda) : \lambda \in \mathbb{R}_+\right\}$ — solve not a single QP but a continuum of QPs using the algebra and geometry of the problem.

If we start from $\lambda = \infty$, we know that only (2) can hold by definition, so not surprisingly $\hat{\beta}(\lambda) \equiv 0$. Now we decrease λ, when will something interesting happen? When we reach:

$$\lambda^* = \max_k 2|\mathbb{X}_{\cdot k}^T \mathbb{Y}|,$$

because then if we keep decreasing λ we will violate (3). Denote $j^* = \arg\max_k 2|\mathbb{X}_{\cdot k}^T \mathbb{Y}|$ So now we need to start changing $\hat{\beta}_{j^*}$ when $\lambda = \lambda^*$ to preserve (3). For $\lambda < \lambda^*$ we need to have condition (1) continue to hold:

$$\begin{aligned} & \mathbb{X}_{\cdot j^*}^T(\mathbb{Y} - \mathbb{X}_{\cdot j^*}\beta_{j^*}) = \frac{\lambda}{2} sgn(\hat{\beta}(\lambda)_{j^*}) \\ \text{(assume WLOG positive sign)} \qquad & \mathbb{X}_{\cdot j^*}^T\mathbb{Y} - \mathbb{X}_{\cdot j^*}^T\mathbb{X}_{\cdot j^*}\beta_{j^*} = \frac{\lambda}{2} \\ & \hat{\beta}_{j^*}(\lambda) = \frac{\lambda^* - \lambda}{2\mathbb{X}_{\cdot j^*}^T\mathbb{X}_{\cdot j^*}} = \frac{\lambda^* - \lambda}{2\|\mathbb{X}_{\cdot j^*}\|_2^2}, \end{aligned}$$

so we know exactly how to proceed for now, and will keep going while condition (2) holds:

$$|\mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}_{\cdot j^*}\hat{\beta}_{j^*}(\lambda))| < \frac{\lambda}{2} \;\; \forall k \neq j^*.$$

Instead of writing specifically the formula for this next stage, let's treat it generically now as an "induction" step. Assume that for some λ_1 we have an optimal solution $\hat{\beta}(\lambda_1)$ complying with the conditions (1)-(3). We want to continue generating the solution for $\lambda < \lambda_1$. Denote the set of *active variables* that comply with condition (1) at λ_1 by $\mathcal{A}$:

$$\mathcal{A} = \left\{j : \hat{\beta}(\lambda_1)_j \neq 0\right\},$$

and correspondingly by $\mathbb{X}_{\mathcal{A}}$ the relevant columns of $\mathbb{X}$. We want to make sure we maintain (1) for the set $\mathcal{A}$ as we change λ :

$$\begin{aligned}
&\mathbb{X}_{\mathcal{A}}^T\left(\mathbb{Y}-\mathbb{X}_{\mathcal{A}}\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}\right)=\frac{\lambda_1-\triangle\lambda}{2}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}})\\
&\mathbb{X}_{\mathcal{A}}^T\left(\mathbb{Y}-\mathbb{X}_{\mathcal{A}}\left[\hat{\beta}(\lambda_1)_{\mathcal{A}}+(\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}-\hat{\beta}(\lambda_1)_{\mathcal{A}})\right]\right)=\frac{\lambda_1-\triangle\lambda}{2}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}})\\
&\mathbb{X}_{\mathcal{A}}^T\left(\mathbb{Y}-\mathbb{X}_{\mathcal{A}}\hat{\beta}(\lambda_1)_{\mathcal{A}}\right)-\mathbb{X}_{\mathcal{A}}^T\mathbb{X}_{\mathcal{A}}\left(\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}-\hat{\beta}(\lambda_1)_{\mathcal{A}}\right)=\frac{\lambda_1}{2}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}})-\frac{\triangle\lambda}{2}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}}),
\end{aligned}$$

In the last row we notice the first terms on LHS and RHS are equal by the optimality at λ_1, denote $\triangle\hat{\beta}_{\mathcal{A}}=(\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}-\hat{\beta}(\lambda_1)_{\mathcal{A}})$ so we get the simpler characterization:

$$\mathbb{X}_{\mathcal{A}}^T\mathbb{X}_{\mathcal{A}}\triangle\hat{\beta}_{\mathcal{A}}=\frac{\triangle\lambda}{2}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}})\ \Rightarrow\ \triangle\hat{\beta}_{\mathcal{A}}=\frac{\triangle\lambda}{2}\left(\mathbb{X}_{\mathcal{A}}^T\mathbb{X}_{\mathcal{A}}\right)^{-1}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}}).$$

Critically, this last expression has the form: $\triangle\hat{\beta}_{\mathcal{A}}=\frac{\triangle\lambda}{2}v$ for a *fixed* direction v that does not change as λ changes. Hence we conclude that the solution $\hat{\beta}$ is moving *in a straight line* as λ changes, explicitly:

$$\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}=\hat{\beta}(\lambda_1)_{\mathcal{A}}-\frac{\triangle\lambda}{2}\underbrace{\left(\mathbb{X}_{\mathcal{A}}^T\mathbb{X}_{\mathcal{A}}\right)^{-1}\mathrm{sgn}(\hat{\beta}(\lambda_1)_{\mathcal{A}})}_{v_{\mathcal{A}}}.$$

This makes sure condition (1) is maintained for $\mathcal{A}$, but we also have to make sure we do not violate condition (2) for $j\in\bar{\mathcal{A}}$:

$$-\frac{\lambda_1-\triangle\lambda}{2}\ <\ \mathbb{X}_{\cdot j}^T\left(\mathbb{Y}-\mathbb{X}_{\mathcal{A}}\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}\right)\ <\ \frac{\lambda_1-\triangle\lambda}{2}.$$

Note that these are linear functions of $\triangle\lambda$, therefore finding for which $\triangle\lambda$ we reach equality is solving two linear equalities (only one will have a positive solution):

$$\mathbb{X}_{\cdot j}^T\left(\mathbb{Y}-\mathbb{X}_{\mathcal{A}}\hat{\beta}(\lambda_1-\triangle\lambda)_{\mathcal{A}}\right)=\pm\frac{\lambda_1-\triangle\lambda}{2}.$$

Denote the solution to this by $\triangle\lambda_j$, then we need to find the first (smallest) $\triangle\lambda$ for which equality is reached:

$$j^*=\arg\min_j\triangle\lambda_j,$$

and we know that at $\lambda=\lambda_1-\triangle\lambda_{j^*}$ is the point where the active set will change:

$$\mathcal{A}\to\mathcal{A}\cup\{j^*\},$$

and then we can recalculate the direction $v_{\mathcal{A}}$, and we have completed the induction step.

All in all, we have described the set of Lasso solutions $\left\{\hat{\beta}(\lambda)\ :\ 0\le\lambda<\infty\right\}$ through a collection of *knots* $\infty>\lambda_1>\lambda_2>\ldots>0$ such that for $\lambda_j>\lambda>\lambda_{j+1}$ we have

$$\hat{\beta}(\lambda)=\hat{\beta}(\lambda_j)+\frac{\lambda_j-\lambda}{2}v_j.$$

In other words, the solution path is a collection of straight lines with direction v_j, which change direction everytime it reaches a knot. We also know that the set of active variables is monotone increasing as we reach equality in (2) and add a variable each time.

The important benefits of this understanding of the Lasso algorithm:

1. Computational: For the algorithm as we described it so far there are the most $\min(n, p)$ steps because we only add variables to $\mathcal{A}$, and if $n < p$ once we reach $|\mathcal{A}| = n$ variables, we have that the columns of $\mathbb{X}_{\mathcal{A}}$ are a basis of $\mathbb{R}^n$, so the correlations are maintained for all variables. At each step we need to invert $\mathbb{X}_{\mathcal{A}}^T \mathbb{X}_{\mathcal{A}}$ with one more column in $\mathbb{X}_{\mathcal{A}}$, and this can be calculated efficiently based on the previous inverse (Sherman-Morrison-Woodbury) Lemma. Solving the linear equalities to find $\triangle\lambda_{j^*}$ is cheap, and overall Efron et al.(2004) argue that in this setting finding the entire Lasso pass has comparable computational complexity to solving one OLS problem: $O(np\min(n, p))$. However, this is ignoring some complications we will mention briefly below.

2. From a geometrical and statistical perspective, we can learn a lot about the Lasso and the nature of its solutions from analyzing the solution path. For example, the LARS paper and followup work have used it to analyze the connection between Lasso and Boosting — an important modern approach to predictive modeling, which can be interpreted as an approximation of a LARS-Lasso algorithm in (very) high dimension.

3. It turns out that the pathwise approach can be expanded to other problems beyond this simple Lasso, and yield computationally efficient and statistically insightful algorithms for them as well. This has been done for Support vector machines (Hastie et al., 2004), and investigated for general loss-penalty families (Rosset and Zhu 2007).

Our description touches on the main general aspects, but it is missing one important point: it is not accurate to assume that variables only enter $\mathcal{A}$ as λ increases and never come out. The reason is the term $\text{sgn}(\hat{\beta}(\lambda)_{\mathcal{A}})$ which seems innocent, but is critical: It is possible and indeed happens that as move in direction $v_{\mathcal{A}}$, some of the coefficients in $\mathcal{A}$ can cross zero! In this setting if we keep going then (1) will no longer hold since it has the wrong (opposite) sign! It can be shown that in this setting, the variable should come out of $\mathcal{A}$ and then the conditions will be maintained. In other words variables can both enter and exit $\mathcal{A}$. For the computational complexity it means in theory it can be exponential instead of being OLS-like, and indeed some people have been able to come up with exponential counter-examples (which are completely unrealistic as real data of course). The bottom line is that the statement on OLS-like complexity can be inaccurate and in high dimension very inaccurate, unfortunately.

Another point worth mentioning is the inclusion of a non-penalized intercept $\hat{\beta}_0$ — this does not change the problem substantially and is easily added, but complicates notations.