Statistics of Big Data, Fall 2021-22

Class notes 6

Sources for today's material:
Least Angle Regression by Efron et al.(2004)
survey by Goldberg et al. on statistical modeling of network data (that appeared in 2010 in the Foundations and Trends in Machine Learning)

LARS-Lasso: continuing algorithm discussion

For the penalized lasso formulations:

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

We reached the optimal solution conditions:

$$|\hat{\beta}(\lambda)_k| > 0 \ \ : \ \ \mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}\beta) = \frac{\lambda}{2} sgn(\hat{\beta}(\lambda)_k) \tag{1}$$

$$|\hat{\beta}(\lambda)_k| = 0 \ \ : \ \ |\mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}\beta)| \leq \frac{\lambda}{2} \tag{2}$$

$$|\mathbb{X}_{\cdot k}^T(\mathbb{Y} - \mathbb{X}\beta)| > \frac{\lambda}{2} \text{ is impossible} \tag{3}$$

Thinking how to use this to track the optimal solution, we realized that:

- For large λ the solution is $\hat{\beta}(\lambda) \equiv 0$
- As we decrease λ and attain equality $2|\mathbb{X}_{\cdot j^*}^T \mathbb{Y}| = \frac{\lambda^*}{2}$, we get that for $\lambda < \lambda^*$, $\triangle\lambda = \lambda^* - \lambda$ we have:
$$\hat{\beta}_{j^*}(\lambda) = \frac{\triangle\lambda}{2\|\mathbb{X}_{\cdot j^*}\|_2^2} \ , \ \ \hat{\beta}_j = 0 \ \forall \ j \neq j^*$$
- This is the case until equality in (2) is achieved for some $j^{**} \neq j^*$ at some $\lambda^{**} < \lambda^*$:
$$|\mathbb{X}_{\cdot j^{**}}^T(\mathbb{Y} - \hat{\beta}_{j^*}(\lambda)\mathbb{X}_{\cdot j^*})| = \frac{\lambda^{**}}{2}.$$
- At this point, if we keep going condition (2) will be violated, so we need to recalculate the direction using $\beta_{j^*}, \beta_{j^{**}}$ to maintain equality condition (1) for both of them.

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 $\left\{\hat{\beta}(\lambda) \ : \ 0 \leq \lambda < \infty\right\}$ 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.x

Network data modeling

A network or graph is a collection of N nodes and E arcs (directed or undirected) between then. The type of questions we want to ask about networks:

1. The nature of the network and connections in it, for example:
   - Is the network connected? What is the length of typical paths between connected nodes ("small world")?
   - Is the network reciprocal: If a node points to others, do they point back to it? If it points to many do many point back?
   - Clusters and high connectivity groups
   - Existence of "hubs" that are close to all
2. The connections between features or properties of the nodes or arcs and the nature of the network: what makes you "popular" etc.

It goes without saying that the answers to questions like this are critical and useful in many areas of science and business, increasingly so as networks (social and others) become central in our lives.

Erdos-Renyi-Gilbert model

This is the simplest and most classical analysis of networks, but still important and relevant. It generally deals with undirected graphs, though the main results also apply to directed.

Assume a graph with N nodes and we randomly generate E undirected edges between pairs of nodes. The formulation has two varieties:

- $G(N, E)$: A random draw from all possible graphs with exactly E edges, each with probability:
$$\frac{1}{\binom{\binom{N}{2}}{E}}.$$
- $G(N, p)$: Each of the $\binom{N}{2}$ arcs is selected with equal probability p, so the probability of a random graph with E edges is
$$p^E(1-p)^{\binom{N}{2}-E},$$
while the overall probability of seeing exactly E edges is:
$$\mathbb{P}(E) = \binom{\binom{N}{2}}{E} p^E(1-p)^{\binom{N}{2}-E}.$$

The main questions they asked about this graph is about the nature of connectivity in this "symmetric" setting. This leads to some powerful and famous results. Denote by λ the average rank for a node in this setting:

$$G(N,p):\ \lambda(N) = Np(N) \quad G(N,E):\ \lambda(N) = 2\frac{E(N)}{N},$$

Then Erdos-Renyi proved the following:

1. If $\lambda(N) < 1$ then as $N \to \infty$ the size of all connected components is $O(\log(N))$ with high probability, meaning the graph will be totally fragmented and most nodes can reach only very few others via the edges.

2. If $\lambda(N) \to 1$ then for large N there will (with high probability) be many components of size $O(N^{2/3})$, meaning the graph will still be highly fragmented, but each node can now reach a decent number of other nodes via the edges.

3. If $\lambda(N) \to c > 1$ then there will (with high probability) be one *huge* component that contains a positive percentage of the points (typically close to 100%), and all other components will be tiny with $O(\log(N))$ nodes. In practice, it means the graph is largely connected.

These results have been very influential, but they have some major simplifying assumption that limit their practical utility. They ignore phenomena that are important and prevalent in real graphs and networks:

- Some nodes are more central and connected than others (hubs)
- In directed graphs, pairs may have mutual relationship: if I have edge Y_{ij} from node i to node j, it is likely to affect (typically make more likely) the edge Y_{ji}.

etc. To build more useful models we have to get away from the completely random assumption and start considering these aspects.

The p_1 model of node properties and edge creation

Consider two nodes i, j and the four possible settings of the edges $Y_{ij} \in \{0, 1\}$, $Y_{ji} \in \{0, 1\}$, as a function of the parameters of the network and the nodes:

- θ: overall rate of connections (like in Erdos-Renyi)
- α_i : *Expansiveness*, measuring how friendly node i is
- β_i : *Popularity*, measuring how attractive node i is
- ρ : *Reciprocity*, measuring how likely $Y_{ij} = Y_{ji}$ is

We also have λ_{ij} a normalization factor. In this setting we can write the four probabilities as a function of the parameters:

$$\begin{aligned}
\log(\mathbb{P}_{ij}(0,0)) &= \lambda_{ij} \\
\log(\mathbb{P}_{ij}(1,0)) &= \lambda_{ij} + \alpha_i + \beta_j + \theta \\
\log(\mathbb{P}_{ij}(0,1)) &= \lambda_{ij} + \alpha_j + \beta_i + \theta \\
\log(\mathbb{P}_{ij}(1,1)) &= \lambda_{ij} + \alpha_i + \alpha_j + \beta_i + \beta_j + \rho + 2\theta
\end{aligned}$$

where λ_{ij} is such that the probabilities sum to 1.

Now note that if we choose $\rho = 0$, $\alpha_i = \beta_i = 0$, $\forall i$, then we get the Erdos-Renyi model with θ only (fixed probability).

To fit this model to data we would write the likelihood as a function of the parameters:

$$\mathcal{L}(\theta, \alpha, \beta, \rho) = Const + \sum_{i,j=1, i \neq j}^{N} y_{ij}(\theta + \alpha_i + \beta_j) + \sum_{i<j} y_{ij} * y_{ji} * rho.$$

This is an exponential family log-likeilhood and we can find the MLE $\hat{\theta}, \hat{\alpha}_i, \hat{\beta}_i, \hat{\rho}$ with standard approaches.