EE227A project

Optimization and Sparsity

March 20, 2018

This year's EE227A final project will be decomposed in two steps. First, we expect all students to accomplish a certain number of tasks on the particular topic of Graphical Model selection (75 % of the grade). Second, students will dive into another particular subtopic of their choice on sparse optimization either from a more practical standpoint (25 % of the grade).

1 Project Overview

The goal of this project is to understand the heterogeneity of point of views from different news journals. We will focus on how actors of stories are differently introduced by the journalists. As the data is high-dimensional, that will be the opportunity to learn about sparsity in optimization. Firstly, we will focus on a specific preprocessing using sentiment analysis and study this heterogeneity using the graphical lasso. Secondly, you will be asked to criticize the preprocessing made, refine it or propose a new one, and use another tool of your choice to analyze this dataset.

Overall, this project will be the opportunity for you :

1. manipulate complex and specific data
2. understand some theoretical properties of an algorithm
3. implement an algorithm from scratch

2 Introduction

2.1 Sparse optimization

Sparsity is a particular topic of convex optimization with a lot of theoretical analyses (convergence of algorithms, statistical consistency, connections to hypothesis testing, interesting complexity bounds from high-dimensional statistics...).

2.2 Actors annotation dataset

For this project, we will use extracted data from newsLens `http://newslens.berkeley.edu` (Laban et al. 2017). News can be annotated and automatically separated into stories. Our analysis will focus on the Brexit (18,561 articles) and ISIS war (39,206 articles). For each of these stories, we can extract the different actors (person in the stories) and link them across different journal by comparing how they are introduced. Our datasets therefore consist of short introductions (= part of a sentence) for different individuals in a given number of articles.

Example “The new Philippine president, Rodrigo Duterte”
“President Duterte, a notoriously foul-mouthed former mayor”
“Rodrigo Duterte, one of Americaś closest allies in Asia”

The three examples above refer to the same entity: Rodrigo Duterte. Example 1 is descriptive and does not reflect the author’s opinion. Examples 2 and 3 reflect a positive and negative opinion the authors express on the entity.

As newsLens scrapes on real-time several journals, (we will focus on a subset of 22 of them), we are interested in the heterogeneity of these introductions (total 387427 introductions for around 37557 persons) when looking at the source (= journal identity). We might assume that depending on the political orientation or the country from which the journal is originated, the same person might be introduced substantially differently.

3 Graphical Model selection

One notable application of sparse optimization is graphical model selection. Graphical models are tools at the intersection if probability theory and graph theory that can be used to model conditional independence in high-dimensional data.

In our configuration, our data matrix will have as features the different journals and as samples the different actors of one given story. Edges in the graph will encode conditional dependencies between our journals. As a consequence, if two journals are well-connected in the inferred graph, we will be able to say that they generally influence themselves in the way they present actors.

Usually in the graphical model literature, the graph is given and one has to make inference over a model. Here we take a different standpoint (common in text analysis, bioinformatics...) where we write a convex optimization problem to recover the structure of the graph.

In this part of the project, we will ask you to prove some results about the optimization procedure, write your own implementation, run the algorithm on the two different stories provided and analyze the results.

3.1 Data preprocessing

Our database — as described earlier — has one more level of granularity that what we would expect. We would have to aggregate the introduction of people over different articles of the same source in order to obtain a design matrix actors by sources. We are given triplets (*apposition*, *source*, *entity*) and our goal is to produce a single discrete (-1, 0, 1) average sentiment score *source* displays for *entity*. Sentiment for a single apposition is obtained using the VADER model for sentiment analysis.

The sentiment scores for an entity from a given source are stored in a list $S[src, sent]$. They are turned into a single averaged sentiment in the following way:

1. First we compute the number of true positives, true negatives, and total count:
$$P[src, ent] = |\{A | A \in S[src, ent], A > 0.25\}|$$
$$N[src, ent] = |\{A | A \in S[src, ent], A < -0.25\}|$$
$$T[src, ent] = |S[src, ent]|$$

2. Afterwards, an average sentiment $D[src, ent]$ is computed:
$$D[src, ent] = \begin{cases} 1, & \text{if } P[src, ent] > 2N[src, ent] \text{ and } P[src, ent] > 0.15T[src, ent] \\ -1, & \text{if } N[src, ent] > 2P[src, ent] \text{ and } N[src, ent] > 0.15T[src, ent] \\ 0, & \text{otherwise} \end{cases}$$
This means that in order for a source to have negative average sentiment for an entity, it must have twice as many true negative appositions as true positives, and have at least 15% of its appositions be negatives.

Examine the code given for the preprocessing of appositions, quickly comment the choice of sentiment analysis, the choice of VADER and the constants used for preprocessing.

3.2 Theory of graphical model selection

The book by Tibshirani, Hastie and Wainwright *Statistical Learning with Sparsity* summarizes the theory of graphical model selection via the Lasso in Chapter 9.

Choose three exercises to solve from this list (9.2, 9.3, 9.4, 9.5, 9.6, 9.7)

3.3 Implementation

In the same chapter, the authors present two different algorithms for graphical model selection.

Choose one algorithm to implement. You will describe all the different steps of your procedure and write the code from scratch. *Hint: the staff suggests getting familiar with estimating the coefficients of a Elastic Net regression, described in Section 4.2. Section 5.4 on coordinate descent can also be important. Then implement Algorithm 9.2 following the book.*

3.4 Applying the graphical lasso to your dataset

You can now run the algorithm to your data with different hyperparameters (L1 regularization) and observe the graph using a spring layout. One would expect journals from the same country to be close in the graph as they would convey same opinions. Also, journals with same political orientations should be close.

For each story, plot the resulting graph with the best hyperparameters. Comment on your results.

One way to check that your results are not random (and therefore be comfortable on the choice of your hyperparameter) is to take a random slice of the data (say 80%) and make sure your edge set is stable when the random slice vary.

4 A more refined view of the problem

For this second part of the problem, you are free to compose your project as you wish, staying focused on our stories and on sparse optimization. You are free to choose one of the following orientation:

- Focus exclusively on applications: change the preprocessing, work with word-count matrices and use sparse multivariate methods (sparse CCA, sparse PCA...). Meaningful comments on the data will be expected here. We do not expect you here to implement any algorithms.
- Focus on one unique algorithm: study the convergence of a sparse multivariate method (solving one supplementary exercise in the book, sparse CCA, sparse PCA...), implement it and comment on the results.

For these projects, you might want to subsample the data as you wish (choose given actors, given stories, focus on two journals, etc...). You will want to spend a fair amount of time explaining your choice of data, preprocessing, and explain why you would choose the method you used to analyze the data.