
EECS 16A

Designing Information Devices and Systems I

Discussion 14B

Fall 2018

1. (Optional) Orthogonal Matching Pursuit Demo

Follow along with your discussion TA.

2. Pollster: Regularized Least Squares

Least squares techniques are useful for many different kinds of prediction problems (also called *regression*). Here, we'll explore how least squares can be used for polling prediction.¹

Your job to predict how each county will vote, either for candidate A or candidate B. To do this you will build a linear model that predicts how each county will vote based on the (un)importance of several topics (the economy, healthcare, education, pineapple pizza, etc). Each topic is graded on an importance scale where a negative value means that the topic is not important and a positive value means that the topic is important. The magnitude of the number corresponds to how important/unimportant the topics are to members of the county.

Each county is represented by a “feature vector” of length 10. The value of each element of the vector captures the importance of that feature to the county. The dataset you are given contains the “feature vectors” for 100 counties as well as how those 100 counties will vote, a value of +1 if the county votes for candidate A and a value of -1 if the county votes for candidate B. We want you to find a “good” linear model using the first 90 counties (this is known as training data) and test the “good”ness of your model on the remaining 10 counties (testing data). Your job in this problem is to find the “good” linear model, with weights α_i , using the training data’s “feature vectors”, \vec{f} , and voting decision, b , to minimize the mean square prediction error of the testing data.

$$b = \alpha_1 f_1 + \alpha_2 f_2 + \dots + \alpha_{10} f_{10} = \vec{\alpha}^T \vec{f} \quad (1)$$

- (a) **Using only the training data**, set up the least squares problem ($\mathbf{A}\vec{x} = \vec{b}$). How are \mathbf{A} , \vec{x} , and \vec{b} constructed? What are the dimensions of \mathbf{A} , \vec{x} and \vec{b} ?
- (b) Using IPython, build these matrices and solve for \hat{x} using linear least squares with the provided function `doLeastSquares(A, b)`. Evaluate what the mean square prediction error is on the **training** data using your linear model (i.e. $\frac{\|\vec{e}\|^2}{N} = \frac{\|\vec{b} - \mathbf{A}\hat{x}\|^2}{N}$). Also, evaluate what the mean square prediction error is on the **testing** data using your linear model.
- (c) A real life problem when building models can be the data itself. In a country with two very polarizing candidates, knowing how a county feels about one topic allows us to predict how important they consider *every* topic. This issue makes the columns of \mathbf{A} almost linearly dependent—something we observed in the image stitching homework problem and in the imaging lab! Let us analyze it further by looking at the eigenvalues of $\mathbf{A}^T \mathbf{A}$ denoted λ_i . Show that the magnitude of the prediction error,

¹The core ideas we learned in class have been extensively further developed—if you're interested you **should take** EE127A (convex optimization), CS189 (machine learning), EE221 (linear systems)! In these classes you learn ideas that build off of the basic least squares problem for applications in finance, healthcare, advertising, image processing, control, and many other fields.

$$\|\vec{e}\| = \|\vec{b} - \mathbf{A}\vec{x}\| = \left\| \vec{b} - \mathbf{A} \left(\frac{\beta_1}{\lambda_1} \vec{v}_1 + \frac{\beta_2}{\lambda_2} \vec{v}_2 + \dots + \frac{\beta_N}{\lambda_N} \vec{v}_N \right) \right\|. \quad (2)$$

The β_i are the coordinates of the vector $\mathbf{A}^T \vec{b}$ in the eigenbasis of $(\mathbf{A}^T \mathbf{A})^{-1}$,

$$\mathbf{A}^T \vec{b} = \beta_1 \vec{v}_1 + \beta_2 \vec{v}_2 + \dots + \beta_N \vec{v}_N \quad (3)$$

(Hint: Consider how the eigenbasis for $(\mathbf{A}^T \mathbf{A})^{-1}$ is related to the eigenbasis of $\mathbf{A}^T \mathbf{A}$). The issue described above will make some of the $\lambda_i \approx 0$. What happens to the magnitude of the prediction error when there exist eigenvalues $\lambda_i \approx 0$? For this problem assume all eigenvalues are distinct and unique.

- (d) In IPython, plot the eigenvalues of $\mathbf{A}^T \mathbf{A}$. Do we encounter the problem described in part (c)? For this problem, an eigenvalue, λ_i is close to 0 when $\lambda_{max} > 100\lambda_i$.
- (e) There are many solutions to this issue, but a common one involves including prior knowledge. We introduce a value γ which we will use to try and rectify the error from part (c). We are given that the weights of our linear model tend to be small (close to zero). Written in equation form,

$$\sqrt{\gamma} \alpha_1 = 0 \dots \sqrt{\gamma} \alpha_{10} = 0 \quad (4)$$

and as a matrix,

$$\sqrt{\gamma} \mathbf{I} \vec{\alpha} = \vec{0}. \quad (5)$$

Let us concatenate the new equations to the bottom of our matrix equation, $\mathbf{A}\vec{x} = \vec{b}$, from the previous parts to create the augmented matrix, $\tilde{\mathbf{A}}$, and the augmented vector, $\tilde{\vec{b}}$. Our modified matrix equations will be,

$$\tilde{\mathbf{A}}\vec{x} = \tilde{\vec{b}} \rightarrow \left[\begin{array}{c} \mathbf{A} \\ \sqrt{\gamma} \mathbf{I} \end{array} \right] \vec{x} = \left[\begin{array}{c} \vec{b} \\ \vec{0} \end{array} \right] \quad (6)$$

What are the dimensions of $\tilde{\mathbf{A}}$, $\tilde{\vec{b}}$? Using IPython, create these matrices using `np.concatenate`.

- (f) Using the solution to the linear least squares problem derived in class, show that solution to the modified linear least squares problem is

$$\vec{x} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T \tilde{\vec{b}} = (\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I})^{-1} \mathbf{A}^T \vec{b}. \quad (7)$$

Hint: Matrix-matrix multiplication can be handled in blocks! However, there are restrictions on the dimensions of the matrices when written in this block format, certain dimensions must agree. In this setting, $\mathbf{W} \in \mathbb{R}^{n \times m}$, $\mathbf{Y} \in \mathbb{R}^{m \times l}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, and $\mathbf{Z} \in \mathbb{R}^{p \times l}$. User beware: Remember that matrix-matrix multiplication does not, in general, commute.

$$\left[\mathbf{W} \mid \mathbf{X} \right] \left[\begin{array}{c} \mathbf{Y} \\ \mathbf{Z} \end{array} \right] = \mathbf{WY} + \mathbf{XZ}$$

(g) Show that the new total prediction error

$$\|\vec{e}\| = \|\vec{b} - \mathbf{A}\vec{x}\| = \left\| \vec{b} - \mathbf{A} \left(\frac{\beta_1}{\lambda_1 + \gamma} \vec{v}_1 + \frac{\beta_2}{\lambda_2 + \gamma} \vec{v}_2 + \dots + \frac{\beta_N}{\lambda_N + \gamma} \vec{v}_N \right) \right\|, \quad (8)$$

using our modified least squares solution, $\vec{x} = (\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I})^{-1} \mathbf{A}^T \vec{b}$, the eigenvalues λ_i of $\mathbf{A}^T \mathbf{A}$, and the eigenvectors \vec{v}_i of $(\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I})^{-1}$. The β_i are the coordinates of the vector $\mathbf{A}^T \vec{b}$ in the eigenbasis of $(\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I})^{-1}$,

$$\mathbf{A}^T \vec{b} = \beta_1 \vec{v}_1 + \beta_2 \vec{v}_2 + \dots + \beta_N \vec{v}_N \quad (9)$$

(Hint: Consider how the eigenbasis for $\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I}$ is related to the eigenbasis of $\mathbf{A}^T \mathbf{A}$. Think about diagonalization of $\mathbf{A}^T \mathbf{A}$.) For this problem assume all eigenvalues are distinct and unique.

- (h) In IPython, in a **single plot** display the eigenvalues of $\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I}$ for several values of γ (e.g. 0, 10, 100). What does this do to the eigenvalues of $\mathbf{A}^T \mathbf{A}$?
- (i) In IPython, let us now find the “best” γ to improve our testing total prediction error. Evaluate the total testing prediction error using different values of γ (use the list in the IPython Notebook). What is the best choice of γ (ie. which gamma minimizes the total testing prediction error)? How does the total testing prediction error using modified least squares with the best choice of γ compare with the total testing prediction error from part (b)?

3. One Magical Procedure (Fall 2015 Final)

This problem will be gone over during linear algebra review session.

Suppose that we have a vector $\vec{x} \in \mathbb{R}^5$ and an $N \times 5$ measurement matrix \mathbf{M} defined by column vectors $\vec{c}_1, \dots, \vec{c}_5$, such that:

$$\mathbf{M}\vec{x} = \begin{bmatrix} | & & | \\ \vec{c}_1 & \dots & \vec{c}_5 \\ | & & | \end{bmatrix} \vec{x} \approx \vec{b}$$

We can treat the vector $\vec{b} \in \mathbb{R}^N$ as a noisy measurement of the vector \vec{x} , with measurement matrix \mathbf{M} and some additional noise in it as well.

You also know that the true \vec{x} is sparse – it only has two non-zero entries and all the rest of the entries are zero in reality. Our goal is to recover this original \vec{x} as best we can.

However, your intern has managed to lose not only the measurements \vec{b} but the entire measurement matrix \mathbf{M} as well!

Fortunately, you have found a backup in which you have all the pairwise inner products $\langle \vec{c}_i, \vec{c}_j \rangle$ between the columns of \mathbf{M} and each other as well as all the inner products $\langle \vec{c}_i, \vec{b} \rangle$ between the columns of \mathbf{M} and the vector \vec{b} . Finally, you also know the inner product $\langle \vec{b}, \vec{b} \rangle$ of \vec{b} with itself.

All the information you have is captured in the following table of inner products. (These are not the vectors themselves.)

$\langle \cdot, \cdot \rangle$	\vec{c}_1	\vec{c}_2	\vec{c}_3	\vec{c}_4	\vec{c}_5	\vec{b}
\vec{c}_1	2	0	1	-1	1	1
\vec{c}_2		2	1	-1	-1	-5
\vec{c}_3			2	0	-1	2
\vec{c}_4				2	-1	6
\vec{c}_5					2	-1
\vec{b}						29

(So, for example, if you read this table, you will see that the inner product $\langle \vec{c}_2, \vec{c}_3 \rangle = 1$, that the inner product $\langle \vec{c}_3, \vec{b} \rangle = 2$, and that the inner product $\langle \vec{b}, \vec{b} \rangle = 29$. By symmetry of the real inner product, $\langle \vec{c}_3, \vec{c}_2 \rangle = 1$ as well.)

Your goal is to find which entries of \vec{x} are non-zero and what their values are.

- Use the information in the table above to answer which of the $\vec{c}_1, \dots, \vec{c}_5$ has the largest magnitude inner product with \vec{b} .
- Let the vector with the largest magnitude inner product with \vec{b} be \vec{c}_a . Let \vec{b}_p be the projection of \vec{b} onto \vec{c}_a . Write \vec{b}_p symbolically as an expression only involving \vec{c}_a, \vec{b} , and their inner products with themselves and each other.
- Use the information in the table above to find which of the column vectors $\vec{c}_1, \dots, \vec{c}_5$ has the largest magnitude inner product with the residue $\vec{b} - \vec{b}_p$.
Hint: The linearity of inner products might prove useful.
- Suppose that the vectors we found in parts (a) and (c) are \vec{c}_a and \vec{c}_c . These correspond to the components of \vec{x} that are non-zero, that is, $\vec{b} \approx x_a \vec{c}_a + x_c \vec{c}_c$. However, there might be noise in the measurements \vec{b} , so we want to find the linear least squares estimates \hat{x}_a and \hat{x}_c . Write a matrix expression for $\begin{bmatrix} \hat{x}_a \\ \hat{x}_c \end{bmatrix}$ in terms of appropriate matrices filled with the inner products of $\vec{c}_a, \vec{c}_c, \vec{b}$.
- Compute the numerical values of \hat{x}_a and \hat{x}_c using the information in the table.