# Optimization Models

## EECS 127 / EECS 227AT

Laurent El Ghaoui

EECS department
UC Berkeley

Fall 2018

# LECTURE 9

# Applications and Limitations of Linear Algebra

*There is hardly any theory which is more elementary than linear algebra, in spite of the fact that generations of professors and textbook writers have obscured its simplicity by preposterous calculations with matrices.*

J. Dieudonné (1906—1992)

# What we'll do

Explore some applications of linear algebra using Linear Algebra tools (e.g., SVD, least-squares, etc):

- Auto-regressive prediction models
- Fast cross-validation in Ridge regression (uses inversion lemma)

We'll also explore some crucial limitations of linear algebra models, including:

- Inability to deal with inequality constraints (example with non-negative LS)
- Inability to deal with non-Euclidean norms (example with $l_\infty$ regression)

# AR models

- Auto-Regressive (AR) models try to describe a time series $y(k)$, $k = 0, 1, \ldots$, according to the model

$$y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + e(k),$$

where $e(k)$ is an error term, assumed to have zero mean.

- If we observe the outputs (regressors)

$$\varphi(k)^\top \doteq [y(k-1)\ y(k-2)\ \cdots\ y(k-n)]$$

and we know the model parameters $a^\top \doteq [a_1\ a_2\ \cdots\ a_n]$, we can *predict* the output value at $k$ as

$$\hat{y}(k) = \varphi(k)^\top a.$$

- The *prediction error* is

$$\epsilon(k) = y(k) - \hat{y}(k) = y(k) - \varphi(k)^\top a$$

# AR models

IDEA:

- Use observed data $\varphi(1), \ldots, \varphi(N)$ to estimate a value $\hat{a}$ of the parameter $a$ which minimizes the prediction errors in LS sense.
- That is, we solve

$$\min_a \sum_{k=1}^{N} (y(k) - \varphi(k)^\top a)^2$$

- This is a LS problem

$$\min_a \|y - \Phi a\|_2^2,$$

with

$$y = [y(1) \ \cdots y(N)]^\top, \quad \Phi = \begin{bmatrix} \varphi(1)^\top \\ \vdots \\ \varphi(N)^\top \end{bmatrix}.$$

- Ridge regression is obtained by adding a $\ell_2$ regularization parameter:

$$\min_a \|y - \Phi a\|_2^2 + \lambda \|a\|_2^2.$$

# Ridge Regression

$$\min_a \|y - \Phi a\|_2^2 + \lambda \|a\|_2^2.$$

- Solvable via linear algebra:

$$\hat{a}_\lambda = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top y$$

- We may avoid computing the inverse for different $\lambda$-values, and use SVD instead. By setting $\Phi = UDV^\top$, one can prove that

$$\begin{aligned} \hat{a}_\lambda &= (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top y \\ &= V \operatorname{diag}\left( \frac{d_j}{d_j^2 + \lambda} \right) U^\top y \end{aligned}$$

  and

$$\hat{y}_\lambda = \Phi \hat{a}_\lambda = \sum_j \left( u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^\top \right) y \doteq S_\lambda y$$

- Smoother matrix:

$$S_\lambda = \Phi (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top.$$

# Fast Cross-Validation

- Choose $\lambda$ so to have good out-of-sample prediction performance
- Train the model on all but one datum (leave-one-out estimation), and evaluate the prediction error on that datum, then average these errors
- One may prove that

$$CV_1(\lambda) = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(k)_\lambda^{(-k)})^2 = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{y(k) - \hat{y}(k)_\lambda}{1 - S_{\lambda_{kk}}}\right)^2,$$

where $\hat{y}(k)_\lambda^{(-k)}$ is the output estimate we obtain by removing the $k$-th observation from the batch, and

$$S_{\lambda_{kk}} = \varphi(k)^\top(\Phi^\top\Phi + \lambda I)^{-1}\varphi(k)$$

- Plotting $CV_1(\lambda)$ as a function of $\lambda$ allows us to select the $\lambda$ value that minimizes the cross-validation error.

# Limits of the Linear Algebra Approach

- Consider Ridge regression, and assume we have a-priori information on the coefficients. For instance, we know that they are positive.
- The problem becomes
$$\min_{a \geq 0} \|y - \Phi a\|_2^2 + \lambda \|a\|_2^2$$
- The constraint $a \geq 0$ makes the problem "a little harder." No longer we have a "closed-form," linear algebra solution.
- Another variation, using an $\ell_1$ regularization term:
$$\min_{a} \|y - \Phi a\|_2^2 + \lambda \|a\|_1$$
Again, no "linear algebra" solution...
- We need new tools for attacking these (and many other) problems!
- It turns out that these problems can still be solved very efficiently, with a computational effort comparable to that of "linear algebra" solutions...