CS/ECE/ME 532
Matrix Methods in Machine Learning

Welcome!

Announcements

- Mid Course Feedback Survey
- This week:
  - ~~Last class: finish Unit 4~~
  - Today: start Unit 5
- Next week: spring break
- After spring break:
  - Tuesday: Assessment 3 review
  - Thursday: Assessment 3
- Assignment 7 is due the week after Assessment 3, but:
  - Q1 on the Eigendecomposition (attempt before the assessment)

Activity 16

Hints for 1: video lecture 2.8

$$\boldsymbol{w}^* = \boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T\boldsymbol{y}$$

$\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = c$ define an ellipse
with $\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$
then columns of $\boldsymbol{V}$ define axis

Before:

loss function *regularizer*

$$\min_{\boldsymbol{w}} ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||^2 + \lambda||\boldsymbol{w}||_2^2$$

(This encourages solutions with small 2-norm)

closed form solution

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

What if we want some other regularizer?

(maybe only a few non-zero entries)

$$\min_{\boldsymbol{w}} ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||^2 + \lambda r(\boldsymbol{w})$$

Or different a loss function all together?

$$\min_{\boldsymbol{w}} f(\boldsymbol{w})$$

In general: no closed-form solution ➞ *Interative methods*

Gradient Descent

Main idea: use the gradient to head downhill

goal: $\min_{\boldsymbol{w}} f(\boldsymbol{w})$ *step size*
for $k = 1 \ldots$
$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \tau\nabla f(\boldsymbol{w})$

Gradient descent for least-squares:

goal: $\min_{\boldsymbol{w}} ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2$
for $k = 1 \ldots$
$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \tau(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}^{(k)} - \boldsymbol{X}^T\boldsymbol{y})$

Convergence guaranteed if:

$$0 < \tau < \frac{2}{\sigma_1^2}$$

Steepest direction is normal to the contour!