CPSC 340/540 Tutorial 6

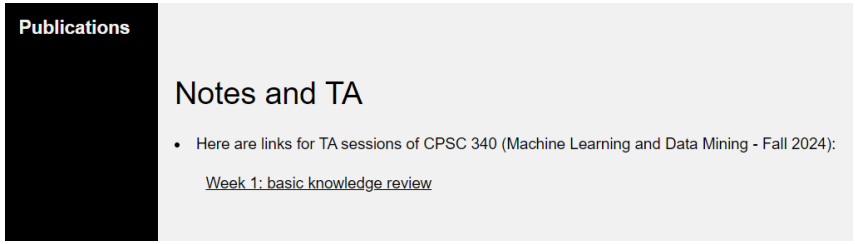
Winter 2024 Term 1

T1A: Tuesday 16:00-17:00;
T1C: Thursday 10:00-11:00;
Office Hour: Wednesday 15:00-16:00

Slides can be found at Piazza and my personal page after T1C.

Yi (Joshua) Ren

https://joshua-ren.github.io/
renyi.joshua@gmail.com
PhD with Danica



Machine Learning:
Learning dynamics, LLM, Compositional Generalization

More helpful on theory

Less helpful on coding

Slides Credit: To various pervious TA's of this course

For students who are lucky enough to see this slides:

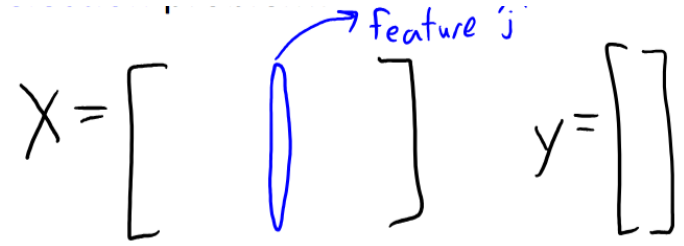
Pay more attention to linear regression and pseudocode!!!

- Feature Selection
- Regularization
- Some mid-term questions

Feature Selection: fundamentals

- What is feature selection:

Egg	Milk	Fish	Wheat	Shellfish	Peanuts	...	Sick?
0	0.7	0	0.3	0	0		1
0.3	0.7	0	0.6	0	0.01		1
0	0	0	0.8	0	0		0
0.3	0.7	1.2	0	0.10	0.01		1



Only use 10 features out of 10k ones

- Role played in traditional ML system (select model → select feature):
  - Model selection: "which model should I use?"
    - KNN vs. decision tree, depth of decision tree, degree of polynomial basis.
  - Feature selection: "which features should I use?"
    - Using feature 10 or not, using x_i^2 as part of basis.

Feature Selection: fundamentals

- Why it is important in traditional ML systems?
  - Some times, the feature space is too big. Recall **curse of dimension**.
  - Some times, there are so many redundant or repeated features. This will introduce **redundant weights.**

gender	mom	dad	mom2	grandma
F	1	0	1	1
M	0	1	0	0
F	0	0	0	0
F	1	1	1	1

$$w_1 x_{mom} + w_2 x_{dad} + w_3 x_{mom2} + w_4 x_{grandma}$$

The model can chose arbitrary combination of (w_1, w_2, w_4) and the prediction will not change.
Then, why we need these extra parameters? (ill-defined problem, etc.)

  - This selection procedure can bring us insights of the problem

    E.g., egg and milk plays an important role in predicting sick or not;
    Shelfish has no influence.

Feature Selection: fundamentals

- Traditionally, manually select features (Association or by experts)

"Association" Approach to Feature Selection

- A simple/common way to do feature selection:
  - For each feature 'j', compute correlation between feature values x^j and 'y'.
    - Say that 'j' is relevant if correlation is above 0.5 or below -0.5.
- Turns feature selection into hypothesis testing for each feature.
    - There are many other measures of "dependence" (Wikipedia).
- Usually gives unsatisfactory results as it ignores variable interactions:
  - Includes irrelevant variables: "Taco Tuesdays".
    - If tacos make you sick, and you often eat tacos on Tuesdays, it will say "Tuesday" is relevant.
  - Excludes relevant variables: "Diet Coke + Mentos Eruption".
    - Diet coke and Mentos don't make you sick on their own, but *together* they make you sick.

Mom, mon2, grandma are all selected!

Feature Selection: fundamentals

- But now, usually consider automatic feature selection
  - Why? Because data is really high-dimensional, indiviual feature is meaningless.
  - How? Using L1-regularization, sparsify w
  - How? Using L0-regularization, stronger pressure, reduce the number of activating features

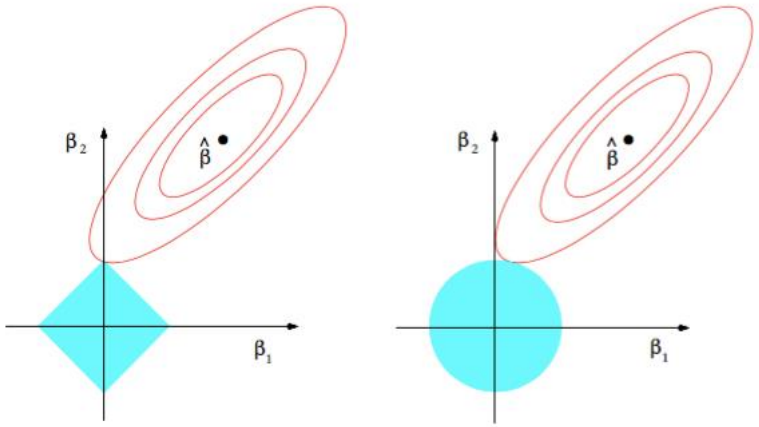


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \le t$ *and* $\beta_1^2 + \beta_2^2 \le t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

In linear models, setting $w_j = 0$ is the same as removing feature 'j':

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \cdots + w_d x_{id}$$

set $w_2 = 0$

$$\hat{y}_i = w_1 x_{i1} + 0 + w_3 x_{i3} + \cdots + w_d x_{id}$$

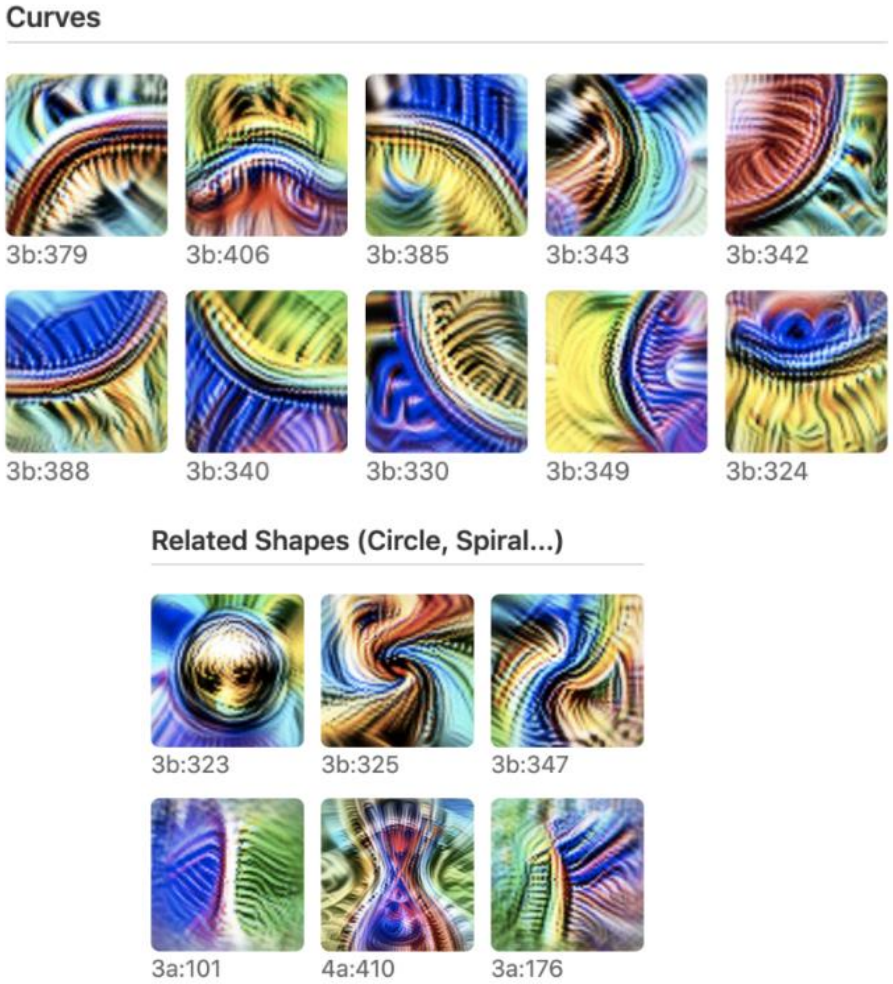
ignore x_{i2}

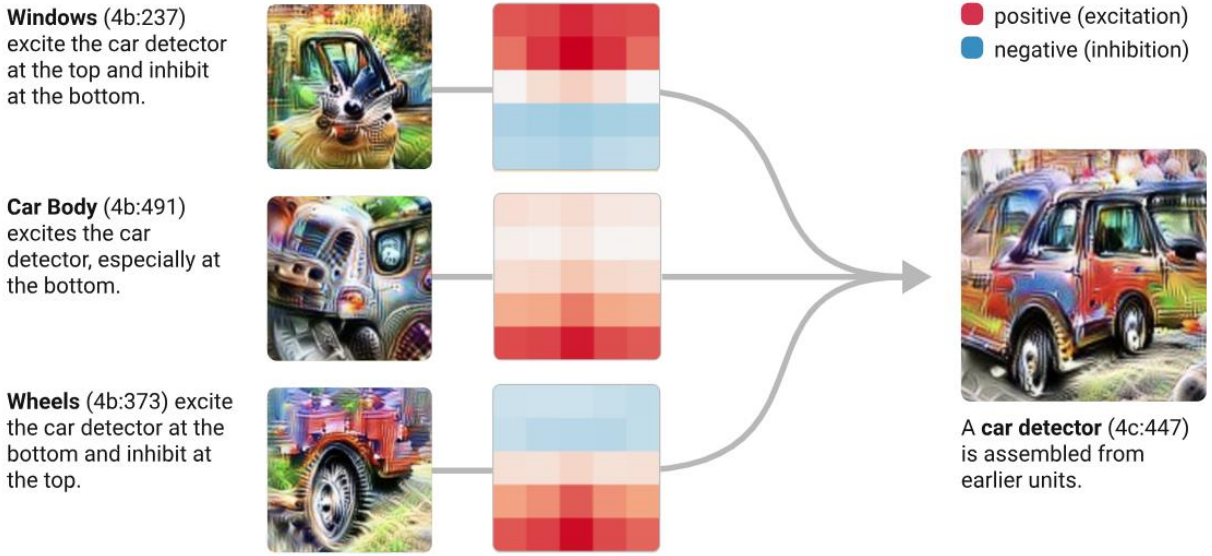
The L0 "norm" is the number of non-zero values ($||w||_0 = \text{size}(S)$).

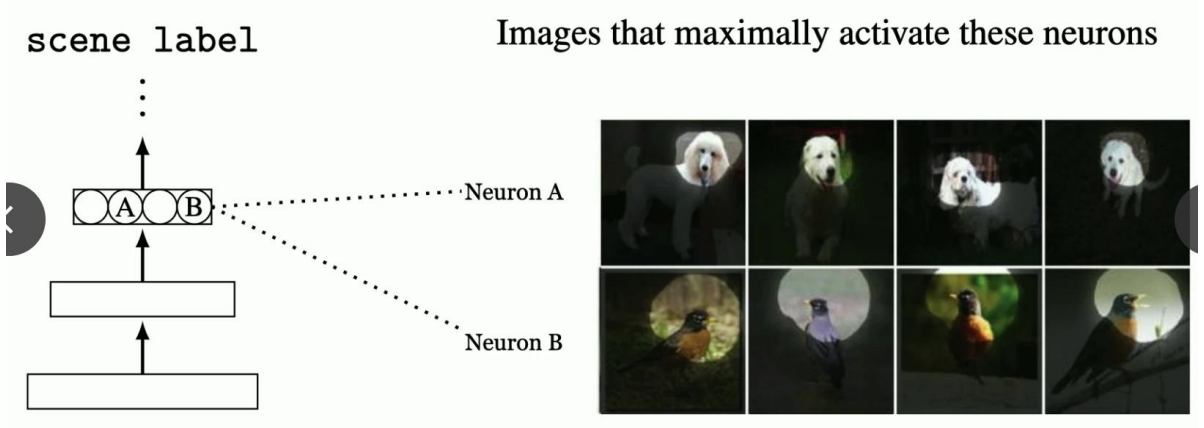
If $w = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 3 \end{bmatrix}$ then $\|w\|_0 = 3$ If $w = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ then $\|w\|_0 = 0$

Feature Selection: fundamentals

- How? Neural network, automatically select







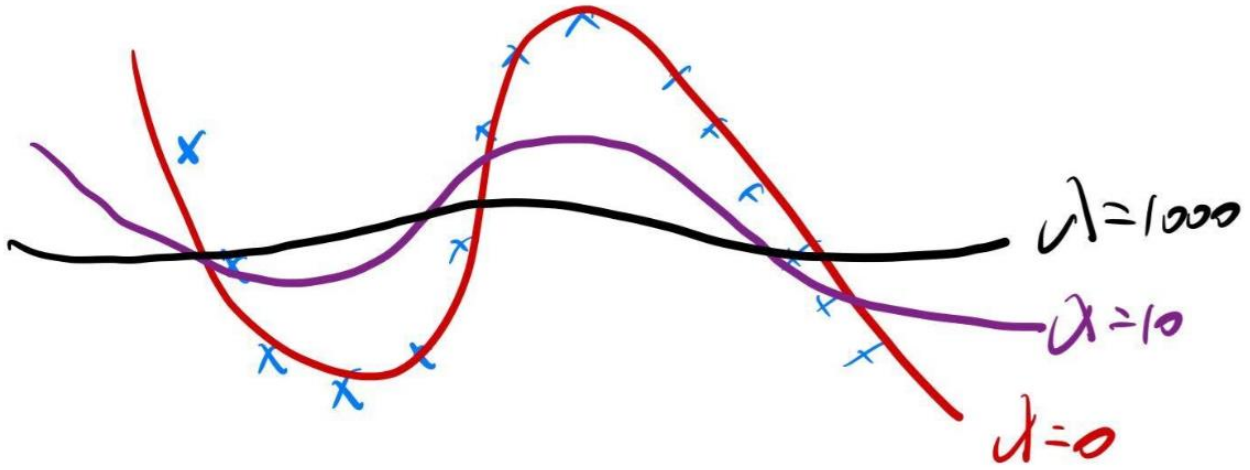
https://distill.pub/2020/circuits/zoom-in/?utm_campaign=The%20Data%20Science%20Roundup&utm_medium=email&utm_source=Revue%20newsletter

- Feature Selection
- Regularization
- Some mid-term questions

Regularization: start from L2-norm

$$f(\boldsymbol{w}; \boldsymbol{x}) = \left|\left|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{Y}\right|\right|_2^2 + \frac{\boldsymbol{\lambda}}{2}\left|\left|\boldsymbol{w}\right|\right|_2^2$$

- Already explained it from "feature selection" perspective (L1, L0)
- Here we understand from the model's capacity (L2)



$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{w}$$

$$\left\|\boldsymbol{Y}\right\| = \left\|\boldsymbol{X}\boldsymbol{w}\right\| \leq \left\|\boldsymbol{X}\right\| * \left\|\boldsymbol{w}\right\|$$

Why use L2-Regularization?

- It's a weird thing to do, but "always use regularization".
  - "Almost always decreases test error" should already convince you.

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

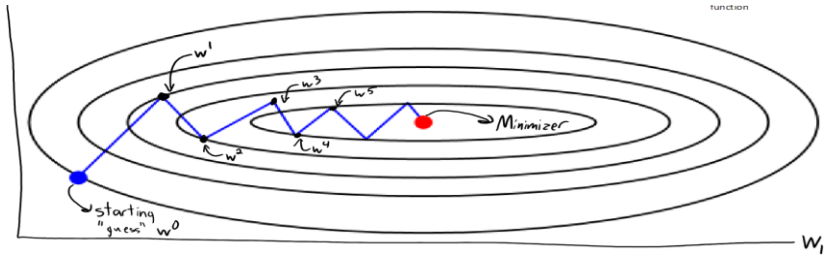
- But here are tons more reasons:
  1. Solution 'w' is unique (if training loss is convex).
  2. $X^T X$ does not need to be invertible (no collinearity issues).
  3. Less sensitive to changes in X or y.
  4. Gradient descent converges faster (bigger λ means fewer iterations
  5. Stein's paradox: if $d \geq 3$, 'shrinking' moves us closer to 'true' w.
  6. Worst case: just set λ small and get the same performance.

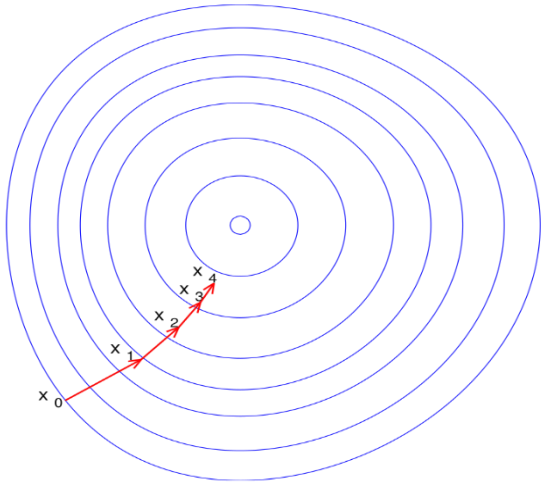
Regularization: standarizing features

- Should we convert to some standard 'unit'?
  - It matters for k-nearest neighbours:
    - "Distance" will be affected more by large features than small features.
  - It matters for regularized least squares:
    - Penalizing $(w_j)^2$ means different things if features 'j' are on different scales.

Egg (#)	Milk (mL)	Fish (g)	Pasta (cups)
0	250	0	1
1	250	200	1
0	0	0	0.5
2	250	150	0

- It also influences gradient descent and the **relative importance** of features.





Regularization: standarizing features

Standardizing Features

- It is common to standardize continuous features:
  - For each feature:
    1. Compute mean and standard deviation:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \qquad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}$$

    2. Subtract mean and divide by standard deviation ("z-score")

Replace x_{ij} with $\frac{x_{ij}-\mu_j}{\sigma_j}$

  - Now measures "standard deviations from mean".
    - And changes in 'w_j' have similar effect for any feature 'j'

- Another interesting reason: **gradient saturation**, e.g., on softmax, Sigmoid, etc.

- Gradient Descent
- Robust Regression
- Some mid-term questions

Midterm topics (non-exhaustive)

- Coverage: up to lecture 14 slide 23
- Linear algebra notation and vector norms (L0, L1, L2, L∞)
- Part 1 (supervised learning): fundamental trade-off, training/validation/test error, overfitting/ optimization bias, golden rule of ML, curse of dimensionality, decision stumps/trees, naive Bayes, knn, ensemble methods (averaging, random forests), bootstrapping, cross validation, IID assumption, parametric vs non-parametric models
- Part 2 (unsupervised learning): clustering, outlier detection, k-means, hierarchical clustering, density-based clustering (DBSCAN)
- Part 3 (linear models): linear regression, least squares, non-linear regression (polynomial basis), gradient descent, error functions, smoothing (log-sum-exp, Huber), robustness to outliers, normal equations, convexity, gradient of linear and quadratic models

2018-Q1

(c) You're working on a machine learning problem and decide you need more data. You collect twice as much training data but end up with the same validation error for your parametric model. Are you likely experiencing underfitting or overfitting? Briefly justify your answer.

Intuitive: Large model need large data →
A model not improved with large data might not be large enough → Underfit

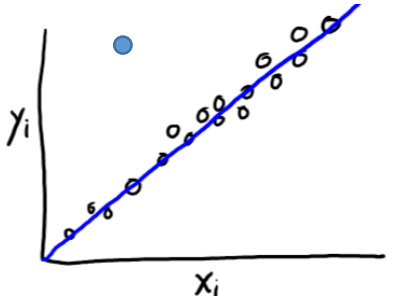
By contra.: If overfit → model recite noise in data → add more data intro. Indep. Noise → fit better (X)
If underfit → e.g., only report the mean → add data keeps the mean → same valid error (V)

2018-Q5

Loss functions.

(a) Describe a situation where using a linear regression model with the squared error could give very misleading results.

Outlier in terms of **Euclidian distance**
Clustering for outlier first, then remove it
L1 loss also helps.



2019-Q2

Increase capacity, overfitting more (wiggly boundary)

(b) Which of the following changes would typically **reduce training error**? Circle all that apply.
Note: for regression and unsupervised models, *assume we use the squared training error* (squared distance to cluster mean for k-means, and squared prediction error for regression).

— ii. increasing k in KNN classification

\+ iii. increasing the maximum tree depth in a random forest

\+ iv. increasing k in k-means clustering

\+ v. using a higher degree polynomial basis for linear regression

\+ vi. running more iterations of gradient descent when fitting least squares (assuming α is small enough)

\+ vii. adding some relevant features to your model

\+ viii. adding some irrelevant features to your model

? ix. switching from the squared error to the absolute error when fitting a linear regression model

2019-Q7 (recap standard version)

Question 7. **(9 points)**

For this question you may find it helpful to use the next page, which is blank.

(a) Consider the following objective, which considers a weighted absolute error with a penalty on the maximum coordinate-wise deviation away from some target value w^0 (that has elements $w_1^0, w_2^0, \ldots w_d^0$),

$$f(w) = \sum_{i=1}^{n} v_i |w^T x_i - y_i| + \lambda \max_{j \in \{1,2,\ldots,d\}} |w_j - w_j^0|,$$

where λ is a non-negative scalar. Re-write this objective function in matrix and norm notation. You can use V as a diagonal matrix with the (non-negative) elements v_i along the diagonal. Make sure your dimensions match.

$$f(w) = \|V(Xw - y)\|_1 + \lambda \|w - w^0\|_\infty.$$

(b) Consider an L2-regularized least squares objective, with a non-diagonal regularizer,

$$f(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{1}{2} w^T \Lambda w,$$

where Λ is a symmetric matrix. Write down a linear system whose solution gives a stationary point of this objective function. Make sure your dimensions match.

$$(X^T X + \Lambda) w = X^T y.$$

Thanks for your time!
Questions?