

446 Section L-2 Norm

TA: Yufei Zhang

Plans for today!

1. This
2. Reminders
3. Problems
 - a. P1 a
 - b. Review Linear Regression
 - c. P1 b
 - d. P2 (if time)

Reminders

- **HW1 due Wed, Jan 28**

Some tips:

- Use office hours to your advantage
 - Student TA OH for homework questions
 - Professor OH for conceptual questions
- Skim the homeworks the day they are assigned and try one problem
 - Motivates you to get things done on time, starting an untouched assignment can be daunting
- Keep a tab open with the lecture slides while you do the homework for reference

Problem 1

Problem 1a

We consider the linear measurement model (parameterized by w), $y_i = x_i^\top w + v_i$ for $i = 1, 2, \dots, m$. The noise v_i for different measurements (x_i, y_i) are all independent and identically distributed. Under our assumption of a linear model, $v_i = y_i - x_i^\top w$. Note Per the principle of maximum likelihood estimation, we seek to maximize

$$\log p_w((x_1, y_1), \dots, (x_m, y_m)) = \log \prod_{i=1}^m p(y_i - x_i^\top w).$$

- (a) Show that when the noise measurements follow a Gaussian distribution ($v_i \sim \mathcal{N}(0, \sigma^2)$), the maximum likelihood estimate of w is the solution to $\min_w \|Xw - Y\|_2^2$. Here each row in X corresponds to a x_i , and each row in Y to y_i .

Problem 1a

Linear model with noise: $y_i = x_i^T w + v_i$

- x_i : features
- w : weights
- v_i : noise

$$Y = XW + \epsilon$$

d : # features
 n : # samples

The diagram illustrates the matrix equation $Y = XW + \epsilon$ with dimensions and element labels:

- Y (n x 1): A vertical vector of samples. Elements are y_1, y_2, \dots, y_n . The top element y_1 is highlighted in red.
- X (n x d): A matrix of features. Elements are x_{11}, \dots, x_{1d} in the first row, and x_{n1}, \dots, x_{nd} in the last row. The top row is highlighted in red.
- W (d x 1): A vertical vector of weights. Elements are w_1, \dots, w_d . The top element w_1 is highlighted in red.
- ϵ (n x 1): A vertical vector of noise. Elements are $\epsilon_1, \dots, \epsilon_n$. The top element ϵ_1 is highlighted in red.

The equation is shown as $Y = XW + \epsilon$ with the matrices and vectors arranged horizontally, separated by an equals sign and a plus sign. The dimensions $n \times 1$, $n \times d$, $d \times 1$, and $n \times 1$ are written below their respective matrices/vectors.

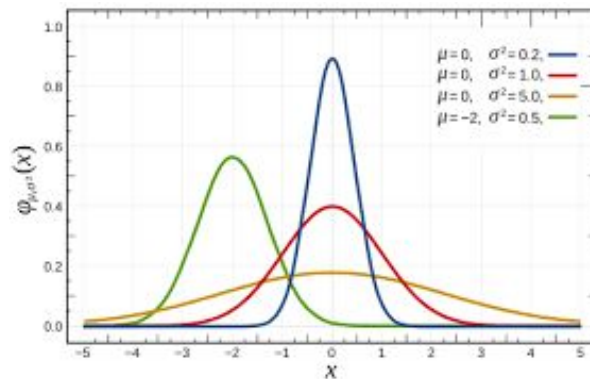
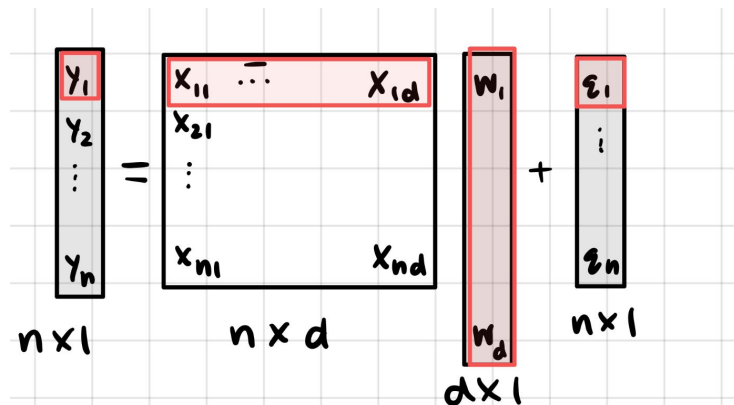
Problem 1a

Linear model with noise: $y_i = x_i^T w + v_i$

- x_i : features
- w : weights
- v_i : noise

Noise (v_i) is i.i.d from the
Gaussian Distribution:

$$p(v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-v^2/2\sigma^2}$$



Problem 1a

Linear model with noise: $y_i = x_i^T w + v_i$

Rewrite as:

$$v_i = y_i - x_i^T w$$

Noise has density $p(v)$: $p(v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-v^2/2\sigma^2}$

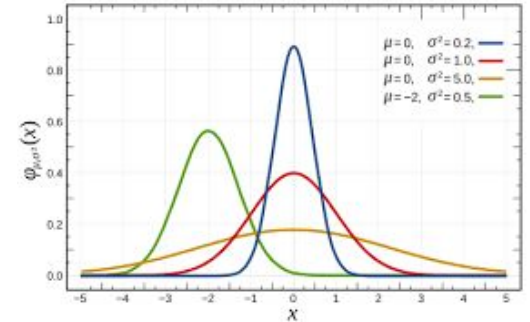
Then:

$$\log p_w((x_1, y_1), \dots, (x_m, y_m)) = \log \prod_{i=1}^m p(y_i - x_i^T w)$$

ie. the probability of observing the data

We will use MLE to maximize the likelihood of seeing the data by finding \hat{w}_{MLE}

$$\begin{matrix} \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{matrix} \end{matrix}_{n \times 1} = \begin{matrix} \begin{matrix} x_{11} & \dots & x_{1d} \\ x_{21} & & \\ \vdots & & \\ x_{n1} & & x_{nd} \end{matrix} \end{matrix}_{n \times d} \begin{matrix} w_1 \\ \vdots \\ w_d \end{matrix}_{d \times 1} + \begin{matrix} v_1 \\ \vdots \\ v_n \end{matrix}_{n \times 1}$$



$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$= \arg \max_w m \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \sum_{i=1}^m -\frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$= \arg \max_w m \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \sum_{i=1}^m -\frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

Fact (Constant offsets do not affect the optimizer).

Example:

$$f(x) = -x^2, \quad g(x) = -x^2 + 10.$$

Both functions are maximized at the same value of x (namely $x = 0$).

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$= \arg \max_w m \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \sum_{i=1}^m -\frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

remove scalar offset

$$= \arg \max_w \sum_{i=1}^m -\frac{1}{2\sigma^2} (y_i - x_i^\top w)^2 \quad (\text{constant offset doesn't affect results})$$

remove positive scaling

$$= \arg \max_w \sum_{i=1}^m -(y_i - x_i^\top w)^2 \quad (\text{constant scalar doesn't affect results})$$

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$= \arg \max_w m \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \sum_{i=1}^m -\frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

remove scalar offset

$$= \arg \max_w \sum_{i=1}^m -\frac{1}{2\sigma^2} (y_i - x_i^\top w)^2 \quad (\text{constant offset doesn't affect results})$$

remove positive scaling

$$= \arg \max_w \sum_{i=1}^m -(y_i - x_i^\top w)^2 \quad (\text{constant scalar doesn't affect results})$$

$$= \arg \min_w \sum_{i=1}^m (y_i - x_i^\top w)^2$$

max likelihood = min -(likelihood)

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$= \arg \max_w m \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \sum_{i=1}^m -\frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

remove scalar offset

$$= \arg \max_w \sum_{i=1}^m -\frac{1}{2\sigma^2} (y_i - x_i^\top w)^2 \quad (\text{constant offset doesn't affect results})$$

remove positive scaling

$$= \arg \max_w \sum_{i=1}^m -(y_i - x_i^\top w)^2 \quad (\text{constant scalar doesn't affect results})$$

$$= \arg \min_w \sum_{i=1}^m (y_i - x_i^\top w)^2 = \arg \min_w \|Xw - Y\|_2^2$$



yay!!

$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

simplify

$$= \arg \max_w \sum_{i=1}^m \log p(y_i - x_i^\top w) \quad [\log(ab) = \log a + \log b]$$

&

$$= \arg \max_w \sum_{i=1}^m \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i^\top w)^2 / 2\sigma^2} \right]$$

substitute

$$= \arg \max_w m \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) + \sum_{i=1}^m -\frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

remove scalar offset

$$= \arg \max_w \sum_{i=1}^m -\frac{1}{2\sigma^2} (y_i - x_i^\top w)^2 \quad (\text{constant offset doesn't affect results})$$

remove positive scaling

$$= \arg \max_w \sum_{i=1}^m -(y_i - x_i^\top w)^2 \quad (\text{constant scalar doesn't affect results})$$

$$= \arg \min_w \sum_{i=1}^m (y_i - x_i^\top w)^2 = \arg \min_w \|Xw - Y\|_2^2$$



now let's solve for the closed form!

MLE → Linear Regression Closed Form

$$\hat{\mathbf{w}} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Find \mathbf{w} that minimizes the SSE. This is $\hat{\mathbf{w}}$.

$$= \underset{w}{\operatorname{argmin}} \sum_{i=1}^n (y_i - (\mathbf{x}_i^\top \mathbf{w} + b))^2$$

Plug in $\hat{y}_i = \mathbf{x}_i^\top \mathbf{w} + b$.

$$= \underset{w}{\operatorname{argmin}} \sum_{i=1}^n (y_i - (\mathbf{x}_i^\top \mathbf{w}))^2$$

Disregard b , since an intercept can be represented by appending a 1 to \mathbf{x} .

We got here
from MLE!

MLE → Linear Regression Closed Form

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Find \mathbf{w} that minimizes the SSE. This is $\hat{\mathbf{w}}$.

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i - (\mathbf{x}_i^\top \mathbf{w} + b))^2$$

Plug in $\hat{y}_i = \mathbf{x}_i^\top \mathbf{w} + b$.

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i - (\mathbf{x}_i^\top \mathbf{w}))^2$$

Disregard b , since an intercept can be represented by appending a 1 to \mathbf{x} .

$$0 = \frac{\partial}{\partial \mathbf{w}} \sum_{i=1}^n (y_i - (\mathbf{x}_i^\top \hat{\mathbf{w}}))^2$$

Find $\hat{\mathbf{w}}$ by taking the partial derivative of the argmin term and setting it equal to 0.

$$= \sum_{i=1}^n \frac{\partial}{\partial \mathbf{w}} (y_i - (\mathbf{x}_i^\top \hat{\mathbf{w}}))^2$$

Property of derivatives.

$$= \sum_{i=1}^n 2(y_i - \mathbf{x}_i^\top \hat{\mathbf{w}})(-\mathbf{x}_i)$$

Derive.

We got here
from MLE!



Let's complete the
proof

MLE \rightarrow Linear Regression Closed Form

$$= \sum_{i=1}^n (y_i - \mathbf{x}_i^\top \hat{\mathbf{w}}) (\mathbf{x}_i)$$

Divide by -2

$$= \sum_{i=1}^n (\mathbf{x}_i) (y_i - \mathbf{x}_i^\top \hat{\mathbf{w}})$$

The term $y_i - \mathbf{x}_i^\top \hat{\mathbf{w}}$ is a scalar. $c\mathbf{x} = \mathbf{x}c$

$$= \sum_{i=1}^n (\mathbf{x}_i y_i - \mathbf{x}_i \mathbf{x}_i^\top \hat{\mathbf{w}})$$

Distribute

$$= \sum_{i=1}^n \mathbf{x}_i y_i - \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \hat{\mathbf{w}}$$

Distribute

$$\left(\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \right) \hat{\mathbf{w}} = \sum_{i=1}^n \mathbf{x}_i y_i$$

Move second term to LHS.

MLE → Linear Regression Closed Form

$$(X^T X) \hat{\mathbf{w}} = X^T \mathbf{y}$$

Convert from vector summation form to matrix form

$$\begin{aligned}(X^T X)^{-1} (X^T X) \hat{\mathbf{w}} &= (X^T X)^{-1} X^T \mathbf{y} \\ \hat{\mathbf{w}} &= (X^T X)^{-1} X^T \mathbf{y}\end{aligned}$$

Left multiply by $(X^T X)^{-1}$

Cancel

I would try and understand this well! It's really cool.

This is mostly in summation form until the very end. The lecture notes have it in matrix form. Look and study the one that makes most sense to you.

Problem 1b: Try it!

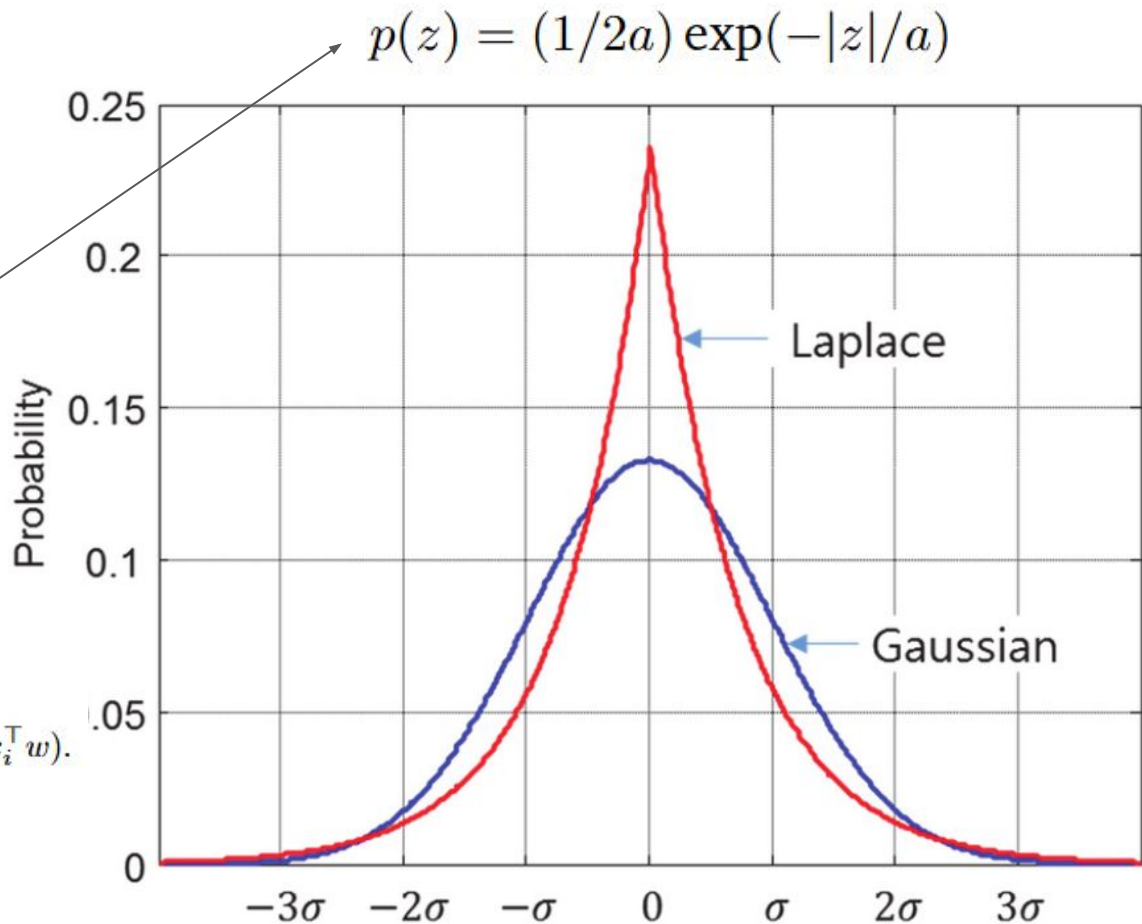
Linear model: $y_i = x_i^\top w + v_i$

Noise (v_i) is i.i.d from the *Laplacian Distribution*

So: $v_i = y_i - x_i^\top w$

We wish to maximize:

$$\log p_w((x_1, y_1), \dots, (x_m, y_m)) = \log \prod_{i=1}^m p(y_i - x_i^\top w).$$



$$\hat{w}_{MLE} = \arg \max_w \log p_w((x_1, y_1), \dots, (x_m, y_m))$$

$$= \arg \max_w \log \prod_{i=1}^m p(y_i - x_i^\top w)$$

$$= \arg \max_w \sum_{i=1}^m \log \underbrace{p(y_i - x_i^\top w)}_{\text{What do we plug in here?}} \quad [\log(ab) = \log a + \log b]$$

Let's look at the steps *before* considering the distribution:

$$= \arg \max_w \sum_{i=1}^m \log \left(\frac{1}{2a} \cdot e^{\frac{-|y_i - x_i^T w|}{a}} \right)$$

$$= \arg \max_w \sum_{i=1}^m \log \left(\frac{1}{2a} \right) - \frac{|y_i - x_i^T w|}{a}$$

$$= \arg \max_w \sum_{i=1}^m -\frac{|y_i - x_i^T w|}{a}$$

constant offset doesn't affect optimization

$$= \frac{1}{a} \cdot \arg \max_w \sum_{i=1}^m -|y_i - x_i^T w|$$

$$= \arg \max_w \sum_{i=1}^m -|y_i - x_i^T w|$$

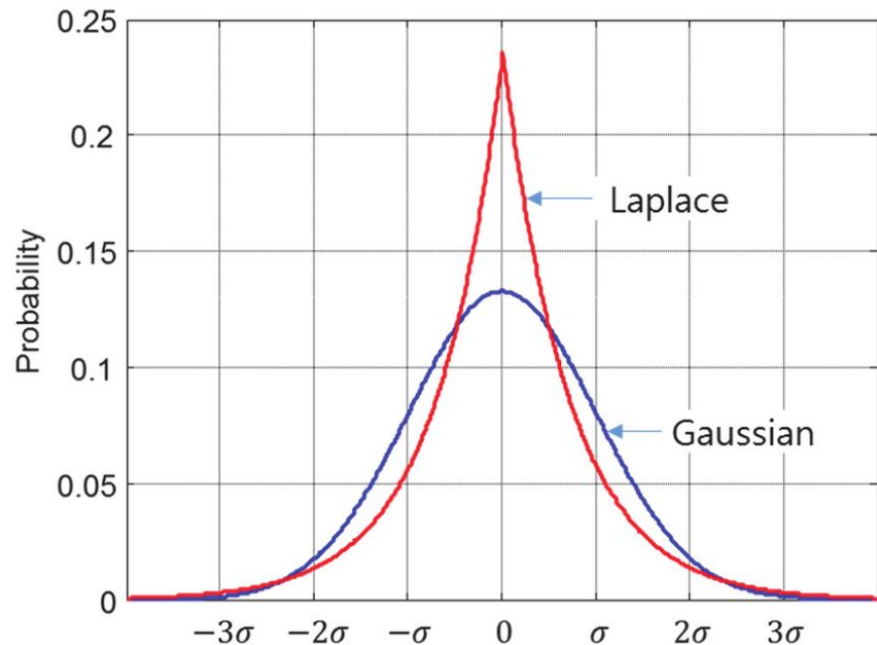
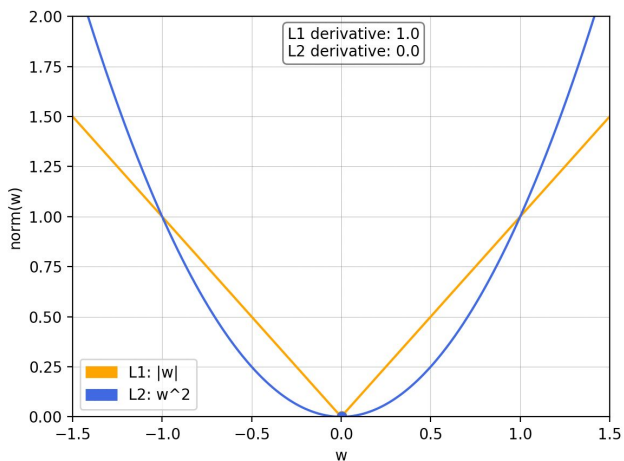
constant offset doesn't affect optimization

$$= \arg \min_w \sum_{i=1}^m |y_i - x_i^T w| = \arg \min_w \|Xw - Y\|_1$$

Therefore the maximum likelihood estimate of w is $\hat{w} = \arg \min_w \|Xw - Y\|_1$.

Cool Connection

Visual similarities exist between the L1 norm vs. the Laplacian, and the L2 norm vs. the Gaussian



Using Laplacian (sharp) as noise results in the L1 norm (sharp) in the optimization equation

Using Gaussian (smooth) as noise results in the L2 norm (smooth) in the optimization equation

Problem 2

What is the difference?

Task/Model Type	Prefer...	Why?
Neural Networks	Normalization	Helps convergence, bounded inputs
KNN, K-Means	Normalization	Distance-based, scale matters
Linear Models (LR, SVM)	Standardization	Assumes normally distributed features
PCA / LDA	Standardization	Sensitive to variance

Standardization

$$\tilde{x}_i^{(j)} = \frac{x_i^{(j)} - \mu_i}{\sigma_i}$$

≠

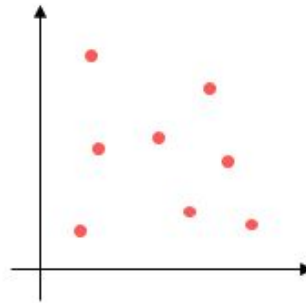
Normalization

$$\tilde{x}_i^{(j)} = \frac{x_i^{(j)} - x_i^{min}}{x_i^{max} - x_i^{min}}$$

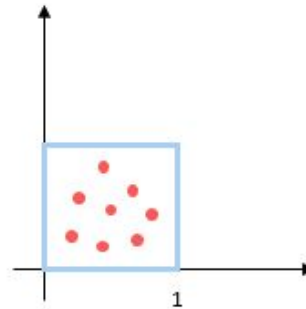
Not the same!

When do we use which?

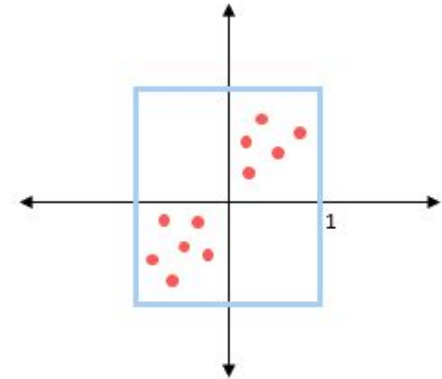
Task/Model Type	Prefer...	Why?
Neural Networks	Normalization	Helps convergence, bounded inputs
KNN, K-Means	Normalization	Distance-based, scale matters
Linear Models (LR, SVM)	Standardization	Assumes normally distributed features
PCA / LDA	Standardization	Sensitive to variance



Actual Data



After normalizing



After standardization

2.1. Data Standardization

Data standardization is the task of transforming each feature in our dataset to have mean 0 and variance 1. The typical way to do this is using the Z-Score, which is defined as below:

$$\tilde{x}_i^{(j)} = \frac{x_i^{(j)} - \mu_i}{\sigma_i}$$

Where μ_i is the mean of each feature and σ_i is the standard deviation of each feature, which are empirically calculated from the data.

Question: what should you do when $\sigma_i = 0$ for some i ?

Solution:

Having $\sigma_i = 0$ for some feature means that the value for that feature is constant in our dataset. If we leave it as 0, we will encounter a divide by 0 error. Since the feature is constant, once we subtract the mean, the new value for the feature will be 0, so we can divide by anything except 0 to avoid this error.

Having $\sigma_i = 0$ is rare, and may be a sign something is wrong with your data or code. One specific case to watch out for is appending your bias column of ones before standardizing.

2.2. Data Normalization

Data normalization refers to the task of rescaling each feature in our dataset to have range [0, 1].

One such method to achieve this is min-max scaling:

$$\tilde{x}_i^{(j)} = \frac{x_i^{(j)} - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

Where x_i^{\min}, x_i^{\max} are the minimum and maximum values of feature i in our dataset, respectively.

When training and evaluating your model, you should calculate the parameters for your normalization or standardization function on the training set **ONLY!**

Question 1: Should we always choose x_i^{\min} and x_i^{\max} based on train data? Can we sometimes do better? Think about cases when we have some underlying information about data.

Solution:

Consider RGB images. These are typically encoded as arrays of shape (3, height, width), with each value being an integer in range [0, 255]. In this case we should just use $x_i^{\max} = 255$ and $x_i^{\min} = 0$ to normalize the data.

In general there can be many cases in which we will know max and/or min values of distribution. **Always examine and visualize data** before transforming it.

Question 2: When can values outside of $[0, 1]$ range in test set cause issues?

Solution:

This might lead to an issue if our model performs any transformations on data that have a limited domain. Consider a model $f(x) = \log(x)^T w$. In this case if test datapoint have a value below 0, this code will fail, as \log has domain $[0, \infty)$.

In general, after you visualize the data, think about what transforms are needed for it to be well behaved. Always pay attention to domains and ranges of each transform since these may lead to NaNs.

2.1 and 2.2 will be important in HW1!

Questions/Chat Time!