STAT 214 Spring 2025
Week 7

Austin Zane

Outline

- GMM and EM
- Submitting jobs to Bridges2
- Train/validation/test splits

GMM & EM

Various resources

Intuition:

- Quick: https://stackoverflow.com/questions/11808074/what-is-an-intuitive-explanation-of-the-expectation-maximization-technique#answer-43561339

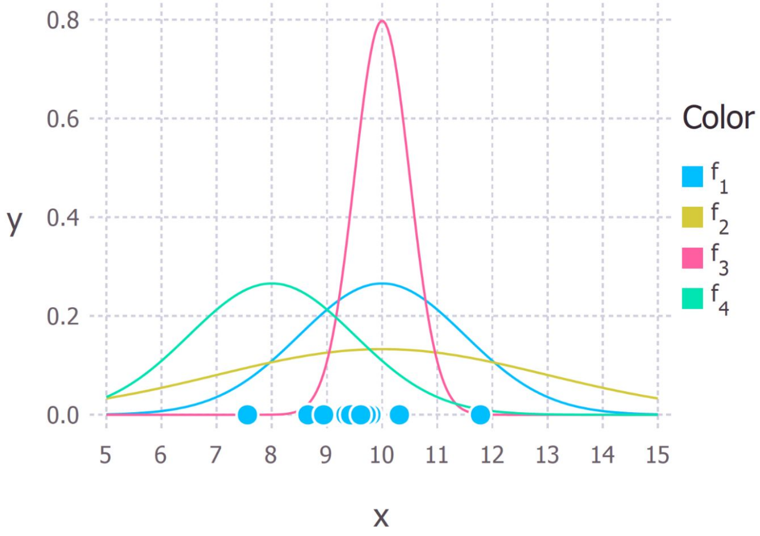
Math:

- Quick overview: http://www.seanborman.com/publications/EM_algorithm.pdf

- More in-depth: https://arxiv.org/pdf/1105.1476.pdf

Maximum Likelihood Estimation Review

- Data: $X_1, \ldots, X_n \overset{i.i.d.}{\sim} p_\theta(x)$
- Likelihood: $\mathcal{L}(\theta) = \prod_{i=1}^{n} p_\theta(x_i)$
- Log-likelihood: $\ell(\theta) = \log \mathcal{L}(\theta)$
- MLE: $$\hat{\theta} = \arg\max_\theta \mathcal{L}(\theta) = \arg\max_\theta \ell(\theta)$$



- **Intuition**: Find the value of θ under which we would be least surprised to see a sample like the observed one.
- **Optimization problem**: take derivative, set equal to zero, solve for parameter.

EM Overview

Motivation 1: “Hard” Maximum Likelihood Estimation Problems

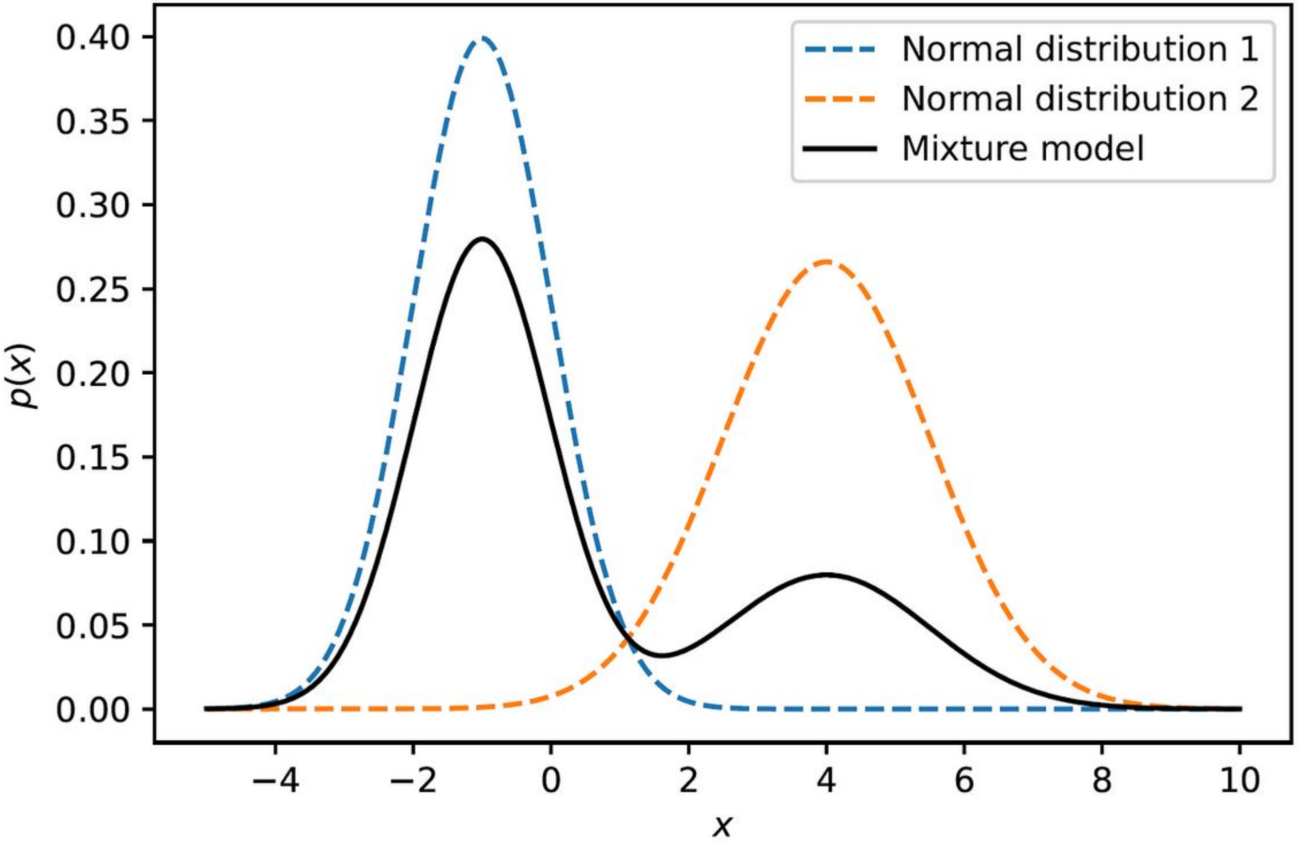
Say we have the following mixture of Gaussians problem:

$$X_1, \ldots, X_n \overset{i.i.d.}{\sim} \pi_1 N(\mu_1, \sigma_1^2) + \pi_2 N(\mu_2, \sigma_2^2),\ \pi_1 + \pi_2 = 1$$

- Because of the sum of normals, the log-likelihood isn’t as helpful as before and taking derivatives w.r.t $\theta = (\pi_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$ and setting equal to zero, etc., doesn’t lead to closed form solutions.
- Instead, we can introduce **latent variables** which tell us which of the two Gaussians each observation comes from: $Z_i \overset{i.i.d.}{\sim} \text{Bernoulli}(1 - \pi_1) + 1$
- If we know the latent allocations then the problem simply becomes two easy MLE exercises. $X_i | Z_i = 1 \overset{i.i.d.}{\sim} N(\mu_1, \sigma_1^2)$

  $X_i | Z_i = 2 \overset{i.i.d.}{\sim} N(\mu_2, \sigma_2^2)$

EM Overview



Normal distribution 1
Normal distribution 2
Mixture model
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
$p(x)$
−4
−2
0
2
4
6
8
10
x

EM Overview

Motivation 2: Clustering

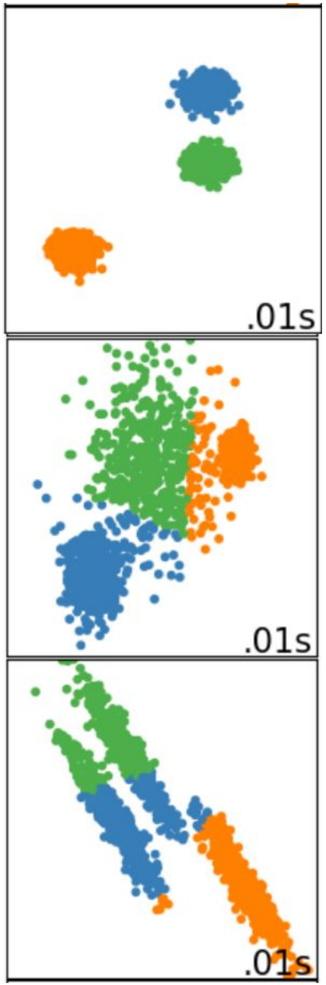
Since EM helps with finding solutions to mixture problems, it lends itself naturally to clustering.

Recall:

- K-means performs well when clusters are homogeneous.
- But if fails miserably when faced with elongated or irregular shapes.

The EM generalizes K-means:

- Still performs great where K-means does.



EM Overview

Motivation 2: Clustering

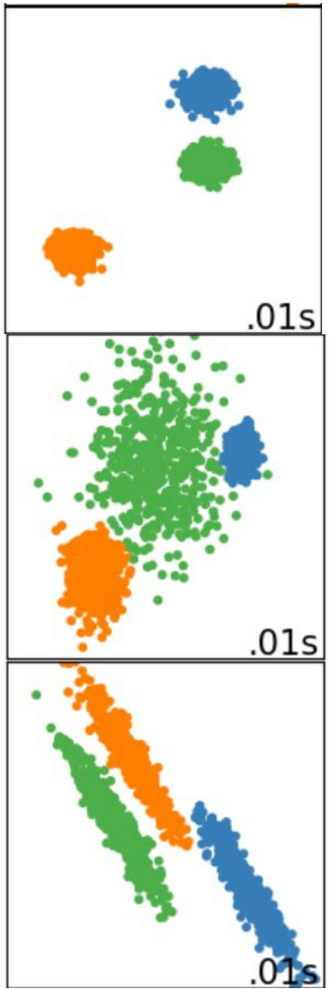
Since EM helps with finding solutions to mixture problems, it lends itself naturally to clustering.

Recall:

- K-means performs well when clusters are homogeneous.
- But if fails miserably when faced with elongated or irregular shapes.

The EM generalizes K-means:

- Still performs great where K-means does.
- But it can also handle differences in spread and symmetry.
- EM is a “soft” clustering algorithm.

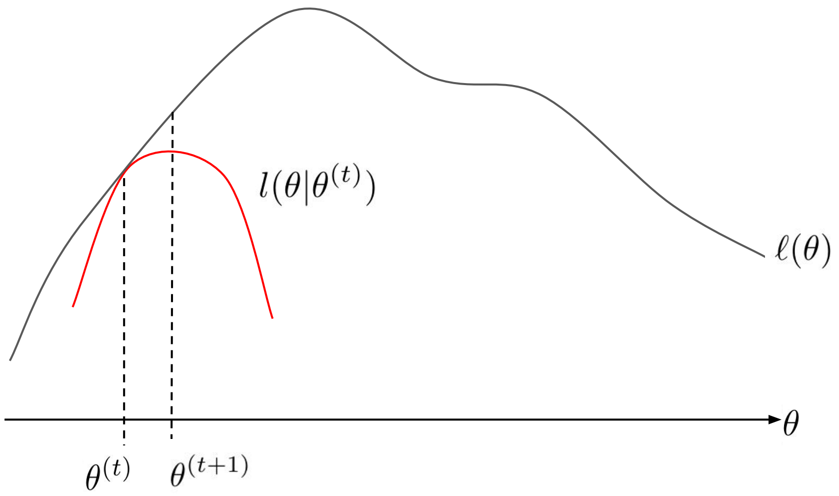


EM algorithm intuition

Say we make a guess $\theta^{(t)}$

The insight of the EM algorithm is that we can find a function $l(\theta|\theta^{(t)})$ such that

- $l(\theta|\theta^{(t)}) \leq \ell(\theta)$
- $l(\theta^{(t)}|\theta^{(t)}) = \ell(\theta^{(t)})$



So any θ that increases $l(\theta|\theta^{(t)})$ also increases $\ell(\theta)$

EM algorithm steps

It turns out that maximizing $l(\theta|\theta^{(t)})$ is equivalent* to maximizing the expectation $Q(\theta|\theta^{(t)}) = \mathbb{E}[\ell(\theta; X, Z)|\theta^{(t)}, X]$

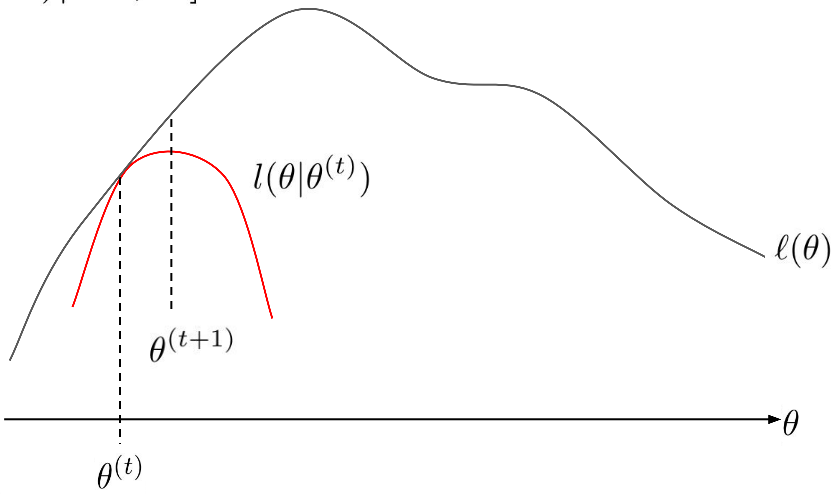
The EM algorithm involves two steps that are repeated until convergence:

1. **E:** Calculate the expectation

$$Q(\theta|\theta^{(t)}) = \mathbb{E}[\ell(\theta; X, Z)|\theta^{(t)}, X]$$

1. **M:** Maximize $Q(\theta|\theta^{(t)})$ w.r.t. θ

Note: we can initialize with a random guess $\theta^{(0)}$



* see the Borman tutorial for the derivation

EM: Gaussian Mixture Example

Same setup from before:

$$X_1, \ldots, X_n \overset{i.i.d.}{\sim} \pi_1 N(\mu_1, \sigma_1^2) + \pi_2 N(\mu_2, \sigma_2^2),\ \pi_1 + \pi_2 = 1$$

$$X_i | Z_i = 1 \overset{i.i.d.}{\sim} N(\mu_1, \sigma_1^2) \qquad X_i | Z_i = 2 \overset{i.i.d.}{\sim} N(\mu_2, \sigma_2^2)$$

$$Z_i \overset{i.i.d.}{\sim} \text{Bernoulli}(1 - \pi_1) + 1$$

Likelihood: $p_\theta(x_i, z_i) = p_\theta(x_i | z_i) p_\theta(z_i) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left\{-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}\right\} \pi_1, & \text{if } Z_i = 1 \\ \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left\{-\frac{(x_i - \mu_2)^2}{2\sigma_2^2}\right\} \pi_2, & \text{if } Z_i = 2 \end{cases}$

$$\log p_\theta(x_i, z_i) = \begin{cases} -\frac{1}{2} \log 2\pi - \log \sigma_1 - \frac{(x_i - \mu_1)^2}{2\sigma_1^2} + \log \pi_1, & \text{if } Z_i = 1 \\ -\frac{1}{2} \log 2\pi - \log \sigma_2 - \frac{(x_i - \mu_2)^2}{2\sigma_2^2} + \log \pi_2, & \text{if } Z_i = 2 \end{cases}$$

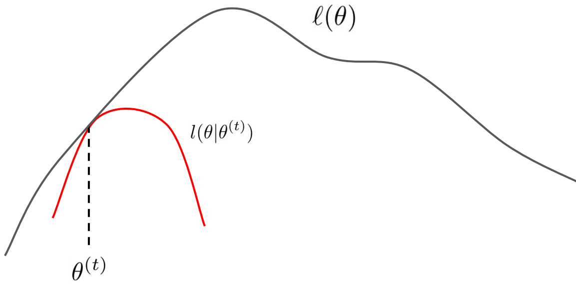
E-Step: Compute $Q(\theta|\theta^{(t)}) = \mathbb{E}[\ell(\theta; X, Z)|\theta^{(t)}, X]$

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{n} \mathbb{E}[\ell(\theta; X_i, Z_i)|\theta^{(t)}, X]$$

$$= \sum_{i=1}^{n} \Big\{ \mathbb{E}[\ell(\theta; X_i, Z_i)|\theta^{(t)}, X, Z_i = 1]\mathbb{P}(Z_i = 1|\theta^{(t)}, X) +$$

$$\mathbb{E}[\ell(\theta; X_i, Z_i)|\theta^{(t)}, X, Z_i = 2]\mathbb{P}(Z_i = 2|\theta^{(t)}, X) \Big\}$$

(law of total expectation)



$$= \sum_{i=1}^{n} \Big\{ \Big(\log \pi_1 - \frac{1}{2}\log 2\pi - \log \sigma_1 - \frac{(x_i - \mu_1)^2}{2\sigma_1^2}\Big) Z_{i,1}^{(t)} +$$

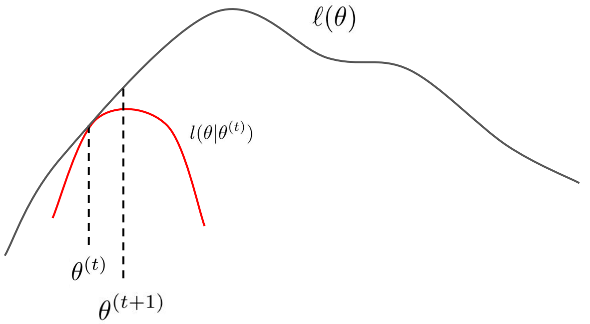
$$\Big(\log \pi_2 - \frac{1}{2}\log 2\pi - \log \sigma_2 - \frac{(x_i - \mu_2)^2}{2\sigma_2^2}\Big) Z_{i,2}^{(t)} \Big\}$$

Use Bayes rule

$$\boxed{\begin{aligned} Z_{i,j}^{(t)} &= \mathbb{P}(Z_i = j|\theta^{(t)}, X) \\ &= \frac{\pi_j^{(t)} \phi(\frac{x_i - \mu_j}{\sigma_j})/\sigma_j}{\sum_{k=1}^{2} \pi_k^{(t)} \phi(\frac{x_i - \mu_k}{\sigma_k})/\sigma_k} \end{aligned}}$$

Standard normal pdf

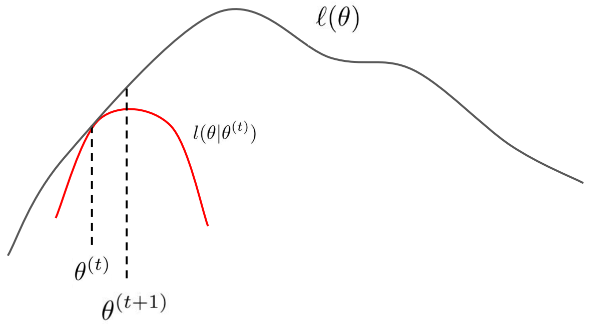
M-Step: Maximize $Q(\theta|\theta^{(t)})$ w.r.t. $\theta = (\pi_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$



- π_1 : $$\frac{\partial Q}{\partial \pi_1} = \frac{\partial}{\partial \pi_1} \sum_{i=1}^{n} \left(\log \pi_1 Z_{i,1}^{(t)} + \log(1-\pi_1) Z_{i,2}^{(t)} \right)$$
$$= \sum_{i=1}^{n} \frac{Z_{i,1}^{(t)}}{\pi_1} + \sum_{i=1}^{n} \frac{Z_{i,2}^{(t)}}{1-\pi_1}$$
$$\overset{set}{=} 0 \quad \Longrightarrow \boxed{\pi_1^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)}}{\sum_i Z_{i,1}^{(t)} + Z_{i,2}^{(t)}} = \frac{1}{n} \sum_i Z_{i,1}^{(t)}}$$

- μ_1 : $$\frac{\partial Q}{\partial \mu_1} = \frac{\partial}{\partial \mu_1} \sum_{i=1}^{n} \left(-\frac{(x_i - \mu_1)^2}{2\sigma_1^2} \right) Z_{i,1}^{(t)} \quad \Longrightarrow \boxed{\mu_1^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)} X_i}{\sum_i Z_{i,1}^{(t)}}}$$
$$\overset{set}{=} 0$$

M-Step: Maximize $Q(\theta|\theta^{(t)})$ w.r.t. $\theta = (\pi_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$



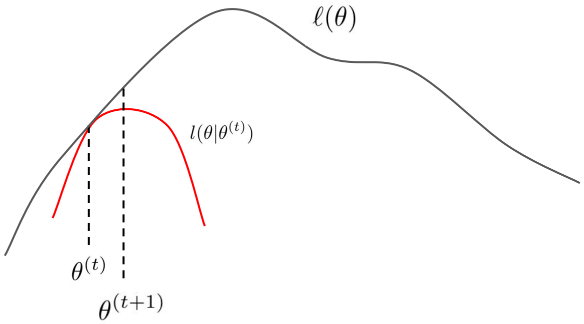
- $$\boxed{\pi_1^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)}}{\sum_i Z_{i,1}^{(t)} + Z_{i,2}^{(t)}} = \frac{1}{n}\sum_i Z_{i,1}^{(t)}}$$

- $$\boxed{\mu_1^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)} X_i}{\sum_i Z_{i,1}^{(t)}}}$$

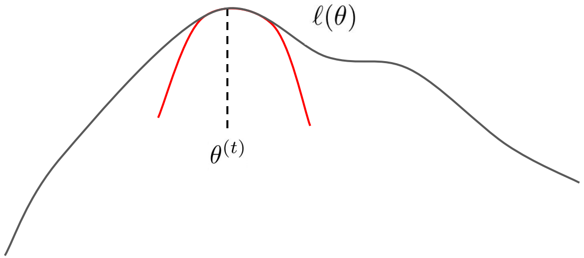
- $\sigma_1:$ $$\frac{\partial Q}{\partial \sigma_1} = \frac{\partial}{\partial \sigma_1}\sum_{i=1}^{n}\left(-\log\sigma_1 - \frac{(x_i-\mu_1)^2}{2\sigma_1^2}\right) Z_{i,1}^{(t)}$$
$$= \sum_{i=1}^{n}\left(-\frac{1}{\sigma_1} + \frac{(x_i-\mu_1)^2}{\sigma_1^3}\right) Z_{i,1}^{(t)}$$
$$\overset{set}{=} 0$$
$$\Longrightarrow \boxed{(\sigma_1^2)^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)}(X_i - \mu_1^{(t+1)})^2}{\sum_i Z_{i,1}^{(t)}}}$$

M-Step: Maximize $Q(\theta|\theta^{(t)})$ w.r.t. $\theta = (\pi_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$

- $$\boxed{\pi_1^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)}}{\sum_i Z_{i,1}^{(t)} + Z_{i,2}^{(t)}} = \frac{1}{n}\sum_i Z_{i,1}^{(t)}}$$
- $$\boxed{\mu_1^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)} X_i}{\sum_i Z_{i,1}^{(t)}}}$$
- $$\boxed{(\sigma_1^2)^{(t+1)} = \frac{\sum_i Z_{i,1}^{(t)} (X_i - \mu_1^{(t+1)})^2}{\sum_i Z_{i,1}^{(t)}}}$$
- Similar process for μ_2 & σ_2



Repeat this process until we find a (local) maximum



`stat-214-gsi/discussion/week7/em.ipynb`

SLURM on Bridges2

Write a shell script to run your job

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -p GPU-shared
#SBATCH -t 5:00:00
#SBATCH --gpus=v100-32:1

#type 'man sbatch' for more information and options
#this job will ask for 1 V100 GPU on a v100-32 node in GPU-shared for 5 hours
#this job would potentially charge 5 GPU SUs

#echo commands to stdout
set -x

# move to working directory
# this job assumes:
# - all input data is stored in this directory
# - all output should be stored in this directory
# - please note that groupname should be replaced by your groupname
# - PSC-username should be replaced by your PSC username
# - path-to-directory should be replaced by the path to your directory where the executable is
module load anaconda3
conda activate env_214
cd /ocean/projects/groupname/PSC-username/path-to-directory

#run Python script which is already in your project space

python train.py
```

Submit your job

To submit:

`sbatch shell_example.sh`

To cancel:

`scancel 12345` (replacing 12345 with the id of your job)

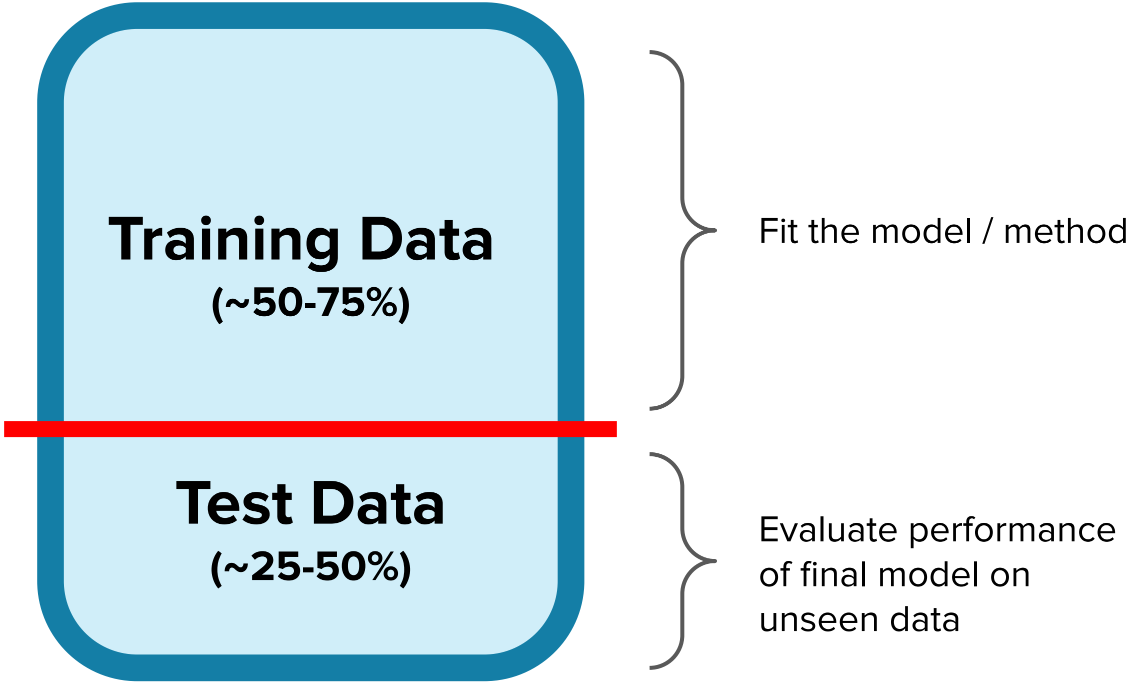
To see my running jobs:

`squeue -u austin.zane`

Splitting your data

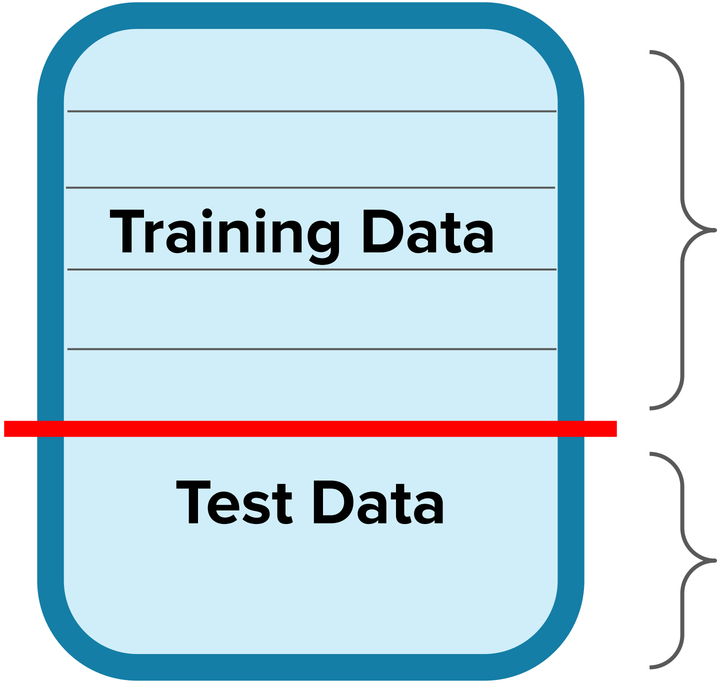
Data splitting: a way to assess generalizability

Simplest case:



Data splitting: a way to assess generalizability

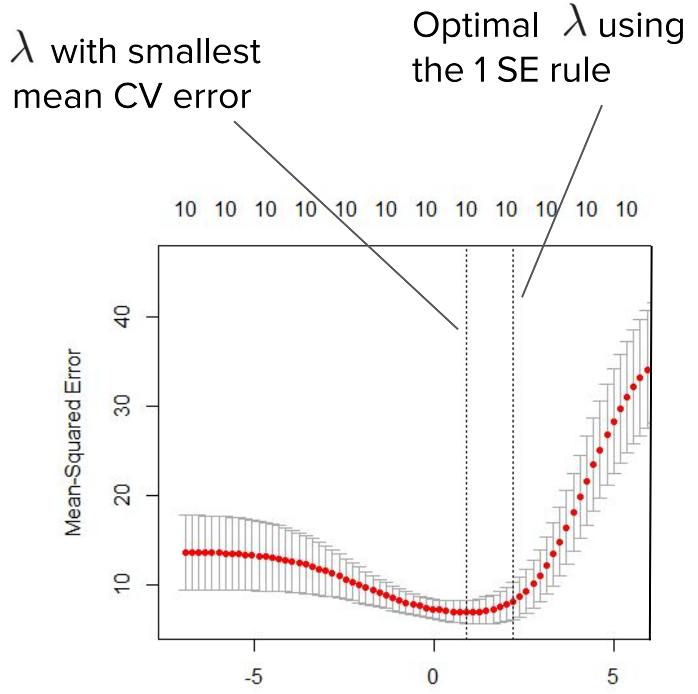
With cross validation:



Do 5-fold or 10-fold CV to choose hyperparameters and fit tuned model

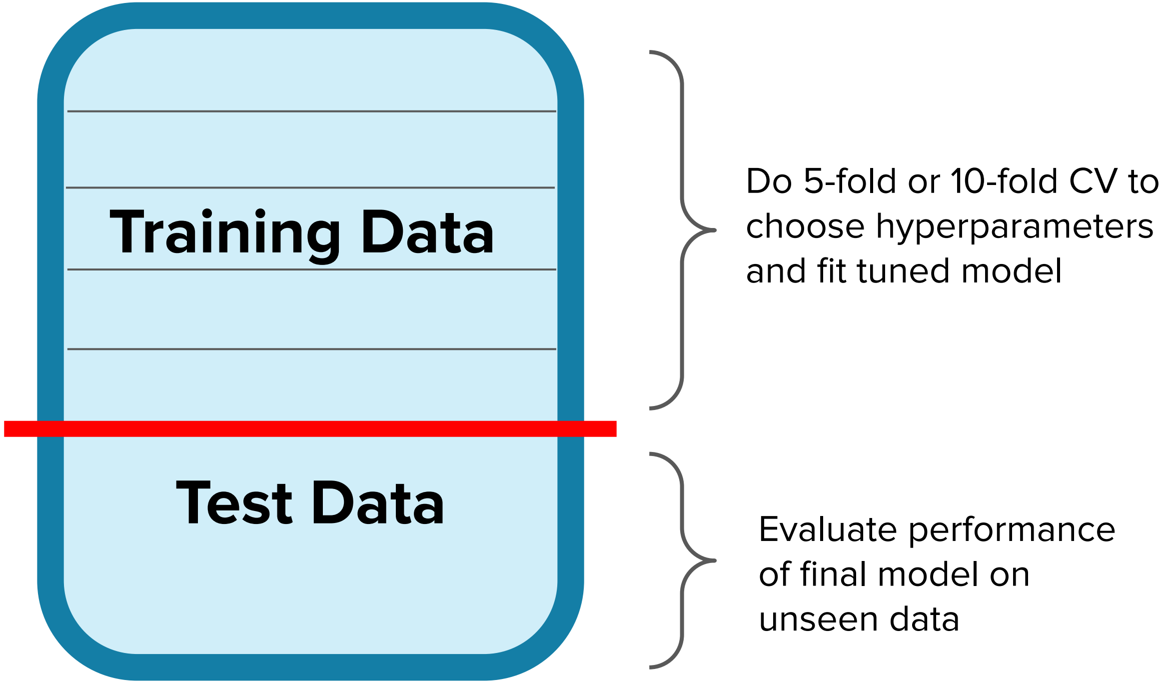
Evaluate performance of final model on unseen data

K-fold cross validation for choosing hyperparameters

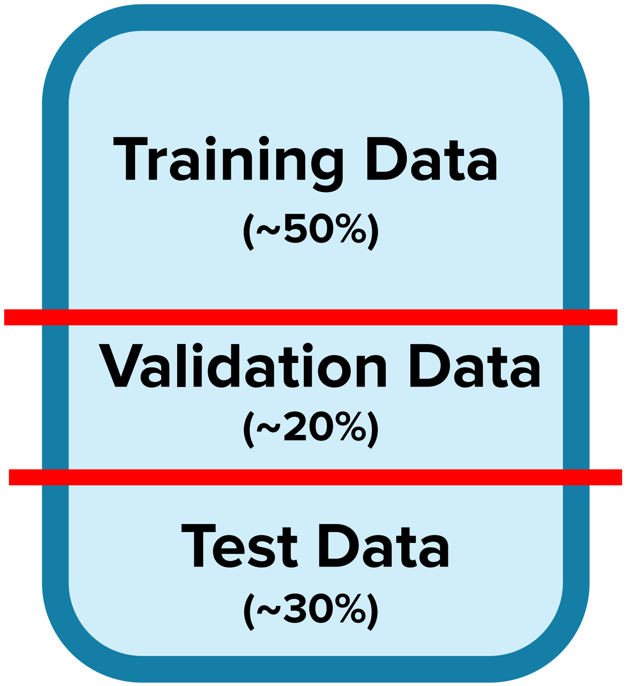


Data splitting: a way to assess generalizability

With cross validation:



Data splitting + tuning hyperparameters + multiple methods



E.g., use CV on this split to pick best λ for both lasso and ridge

Evaluate trained lasso and ridge models and pick the model with lower error

Evaluate performance of final model on unseen data

How much data to hold out for testing?

- Some say “30%” or give an arbitrary number.

- Realistically it depends on the problem.
    - If you have a billion observations, you might only need a several thousand in the test set to get a good idea of accuracy (assuming they are a simple random sample)

- Think about how low you need the standard error of your accuracy estimate to be.