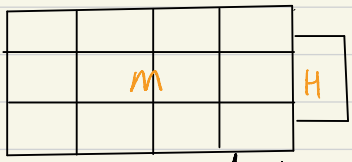
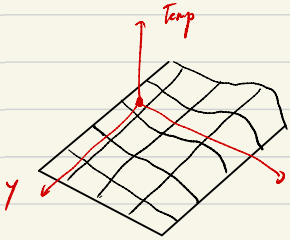
Note for Lop 8 Data Science.



(x, y) record of a temp



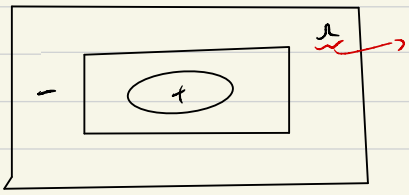
Find a function $f(x, y, t) \rightarrow$ temp

↓

found by testing known information, and explain hidsights or predicted future.

Positive examples: Expect positive results under hypothesis.

Negative ----- ; ~~~ negative - - - — — — —.



U = set of all possible examples.

The venn diagram shows the benefit of negative examples.

None for Sep 13th.

Minimal progress loss → Backup work + commit to avoid loss

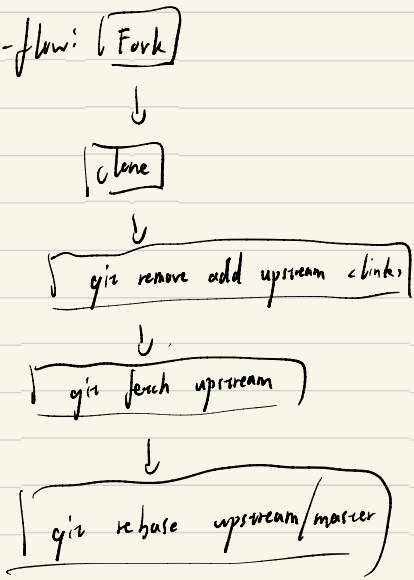
Collab process : - each collaborator creates a new branch.
- provide process for merging their branch into master → PR (pull request)

key step: ① Fork CS506 public repo
② Make all changes to your repo
③ request via pull request.

★ keep copy up-to-date
↓
method: Using git upstream

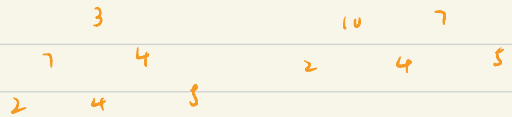
0. git checkout master
2. git fetch upstream master
3. git rebase upstream/master

Work-flow:



Note for Sep 15th.

max path sum



12 -> max(10+4, 7+4)

Data Representations

- Records (n dimensional vector as tuple array, holds different attributes)
- Graphs (nodes connected with edges)
- image (matrix of RGB pixes).
- Text (list of words).
- String (list of characters)
- Time series (list of data at specific intervals of time).

Types of Learning
- supervised learning -> create a model allow for predictions
  ex. Regression, classification.
- unsupervised learning -> more about the structure of the data
  ex. Clustering.

Note for Sep 20th

Feature Space - set of all possible data points

cosine similarity $s(x, y) = \cos\theta$ where θ is angle b/w x, y

↓

using dissimilarity function: $d(x,y) = \frac{1}{s(x,y)}$

↓

As $d(x,y) = k - s(x,y)$ for some k.

↓

$\frac{1}{d(x,y)} = 1 - s(x,y)$.

☆ We use cosine over euclidean distance when direction matters more than magnitude.

Jaccard similarity: ex. take text and each word is an attribute, and values are whether it is in the given doc.

$$J_{sim}(x,y) = \frac{|x \cap y|}{|x \cup y|} \qquad J_{DIS}(x,y) = 1 - \frac{|x \cap y|}{|x \cup y|}$$

Note Sep 22th:

Clustering → create groupings / assignment of objects s.t. 1. similar to each other
2. dissimilar to other clusters.

Clustering problem: • similar data points in same clusters
• dissimilar data points in different clusters.

Types: ① Partitional
② Hierarchical
③ Density-based
④ soft clustering

K-means

$\sum_i \sum_{x \in C_i} \|x - \mu_i\|_2^2$ => Find k point that minimize the loss function

Note: easy if k=1 or k=n.

Steps: 1. Randomly pick k centers
2. Assign each point in dataset to its closest center
3. compare the new centers as means of each clusters
4. Repeat 2. 3 until converge

Limitations: ① splits large number
② dislike non-globular cluster shape
③ dislike varying density

Optimal k selection: Elbow method:

Note Sep 27th

Hierarchical clustering :

- output: tree of clusters showing closest nested points (We can use as a special threshold to generate different clusters)

Agglomerative Clustering Algorithm: ① each point in dataset be in its own cluster
② Compute distance b/w all pairs of clusters
③ merge 2 closest clusters
④ Repeat 2. 3 until converge.

Single link distance: $D_{SL}(C_1, C_2) = \max\{d(p_1, p_2) \mid p_1 \in C_1,\ p_2 \in C_2\}$.

Avg Link distance: $D_{AL}(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \cdot \{d(p_1, p_2) \mid p \in C_1,\ p_2 \in C_2\}$

Centroid Distance: $D_C(C_1, C_2) = d(\mu_1, \mu_2)$

Wards distance: $D_{WD}(C_1, C_2) = \sum_{p \in C_2} d(p, \mu_{12}) - \sum_{p_1 \in C_1} d(p_1, \mu_1) - \sum_{p_2 \in C_2} d(p_2, \mu_2)$

DBscan Algorithms: ① Find ε neighbourhood of each point
② label core if $\geq$ min-point
③ Not core, but in neighborhood
④ else noise
⑤ For each core, assign to same cluster
⑥ Assign border points to nearby clusters.