Problem 1

Indicate in the figure below what the linear maximum margin (SVM) classifier for the binary problem triangle vs. dot is.

Draw three lines:
- the two boundaries of the maximum margin
- the maximum margin hyperplane

Which of the vectors are support vectors?
You can solve this problem visually by drawing your solution into the figure.
Problem 1

Recap: SVM

- large margin classifiers
- for vector space classification
- binary classification
- aim: find a decision boundary that is maximally far from any point in the training data
Recap: SVM

Why do we want to maximize the margin?
Problem 1

Recap: SVM

Why do we want to maximize the margin?

- classification safety margin with respect to errors and random variation
- better generalize to test data
- unique solution for decision boundary
Recap: SVM

Terminology:

- maximum margin: the “board” we use to separate our classes
- maximum margin hyperplane: the decision boundary (middle of the two boundaries of the maximum margin)
- support vectors: the vectors on the boundaries of the max. margin
Problem 1

Indicate in the figure below what the linear maximum margin (SVM) classifier for the binary problem triangle vs. dot is.

Draw three lines:
- the two boundaries of the maximum margin
- the maximum margin hyperplane

Which of the vectors are support vectors?
You can solve this problem visually by drawing your solution into the figure.
Problem 1

Indicate in the figure below what the linear maximum margin (SVM) classifier for the binary problem triangle vs. dot is.
Problem 2

(i) Perform a 3-means clustering for the points below. If a tie occurs during an assignment step, you can freely choose any of the possible assignments.

(ii) Give an example of a clustering that 3-means can converge to that is different from the one in (i)
Recap: K-means

- clustering algorithm
- works in vector space with Euclidean distance
- idea: represent each cluster by its centroid
- goal: minimize the average squared difference from the centroid
- iterative algorithm
Recap: K-means: Algorithm

- initialize centroids
  (e.g. with random points (seeds) from the training data)
- while != stop:
  - assign each vector to its closest centroid
  - update centroids given assigned vectors
Problem 2

Solution to (i):

Initialization:

```
1 2 3
1
2
3
```

Iteration 1:

- Re-assignment:

```
1 2 3
1
2
3
```

- Re-computation:

```
=> converged
```

Iteration 1:

- Re-assignment:

```
1 2 3
1
2
3
```

- Re-computation:

```
=> converged
```
Problem 2

Solution to (ii):

\[ 12 / 22 \]
For this web graph, compute PageRank for each of the three pages. Assume that the PageRank teleport probability is 0.1.
Recap: Page Rank

- **idea: web-graph:**
  - nodes: web pages
  - edges: links between pages
- user clicks through web pages randomly
  \(\Rightarrow\) random walker walks through web graph
- each link is used equiprobably!
- long-term visit rate of a page = PageRank of the page
Problem 3

Recap: Page Rank

- PageRank is only well-defined if web-graph is an ergodic Markov chain (esp.: no dead-ends in graph!)
- make web-graph ergodic: include teleportation!
- teleportation with rate $r$:
  - at a dead end:
    - jump to random page with probability $\frac{1}{\text{num\_pages}}$
  - at a non dead-end:
    - if page $i$ has no link to page $j$:
      - set probability of going from $i$ to $j$ to $r \cdot \frac{1}{\text{num\_pages}}$
    - adjust the probabilities for link connections so that sum of probabilities stays 1
Recap: Page Rank: Computation

If our current probability vector is \( x \),
then it will be \( x \cdot P \) after one step
and \( x \cdot P^2 \) after two steps
and \( x \cdot P^i \) after \( i \) steps.

\((P: \text{transition probability matrix with teleportation})\)

This converges. Hence, for the PageRank vector \( \pi: \pi = \pi \cdot P \)
\(\Rightarrow \pi \) is the left eigenvector for the eigenvalue 1.

Power method:
start with any distribution \( x \) and multiply \( P \) until the result converges.
Problem 3

For this web graph, compute PageRank for each of the three pages. Assume that the PageRank teleport probability is 0.1.
Problem 3

Link-matrix:
\[
\begin{pmatrix}
0 & 1 & 1 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}
\]

Probability transition matrix:
\[
\begin{pmatrix}
0 & 0.5 & 0.5 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}
\]

Teleported matrix:
\[
P = \begin{pmatrix}
1 & \frac{29}{30} & \frac{29}{30} \\
\frac{1}{30} & \frac{60}{1} & \frac{60}{14} \\
\frac{1}{30} & \frac{30}{14} & \frac{15}{1} \\
\frac{30}{15} & \frac{15}{30}
\end{pmatrix}
\]
Problem 3

- Initialize $x$ randomly: $x = \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)$
- $x \cdot P = \left( \frac{1}{30}, \frac{29}{60}, \frac{29}{60} \right)$
- $x \cdot P^2 = \left( \frac{1}{30}, \frac{29}{60}, \frac{29}{60} \right)$

$\Rightarrow$ Convergence $\Rightarrow \pi = \left( \frac{1}{30}, \frac{29}{60}, \frac{29}{60} \right)$
For this web graph, compute PageRank for each of the three pages. Assume that the PageRank teleport probability is 0.1.

**Hint:** Using symmetries to simplify and solving with linear equations might be easier than using iterative methods.
Solution 2 (using symmetries):

- in-degree of $d_1$: 0
  \[ \Rightarrow \text{PageRank}(d_1) = 0,1 \cdot \frac{1}{3} = \frac{1}{30} \text{ (teleport)} \]
- by symmetry: PageRank($d_2$) = PageRank($d_3$)
  \[ \Rightarrow \text{PageRank}(d_2) = \text{PageRank}(d_3) = \frac{1 - \frac{1}{30}}{2} = \frac{29}{60} \]
The end

Thank you for your attention!

Do you have any questions?