

Nearest Neighbor Methods

Shusen Wang

K-Nearest Neighbor (KNN)

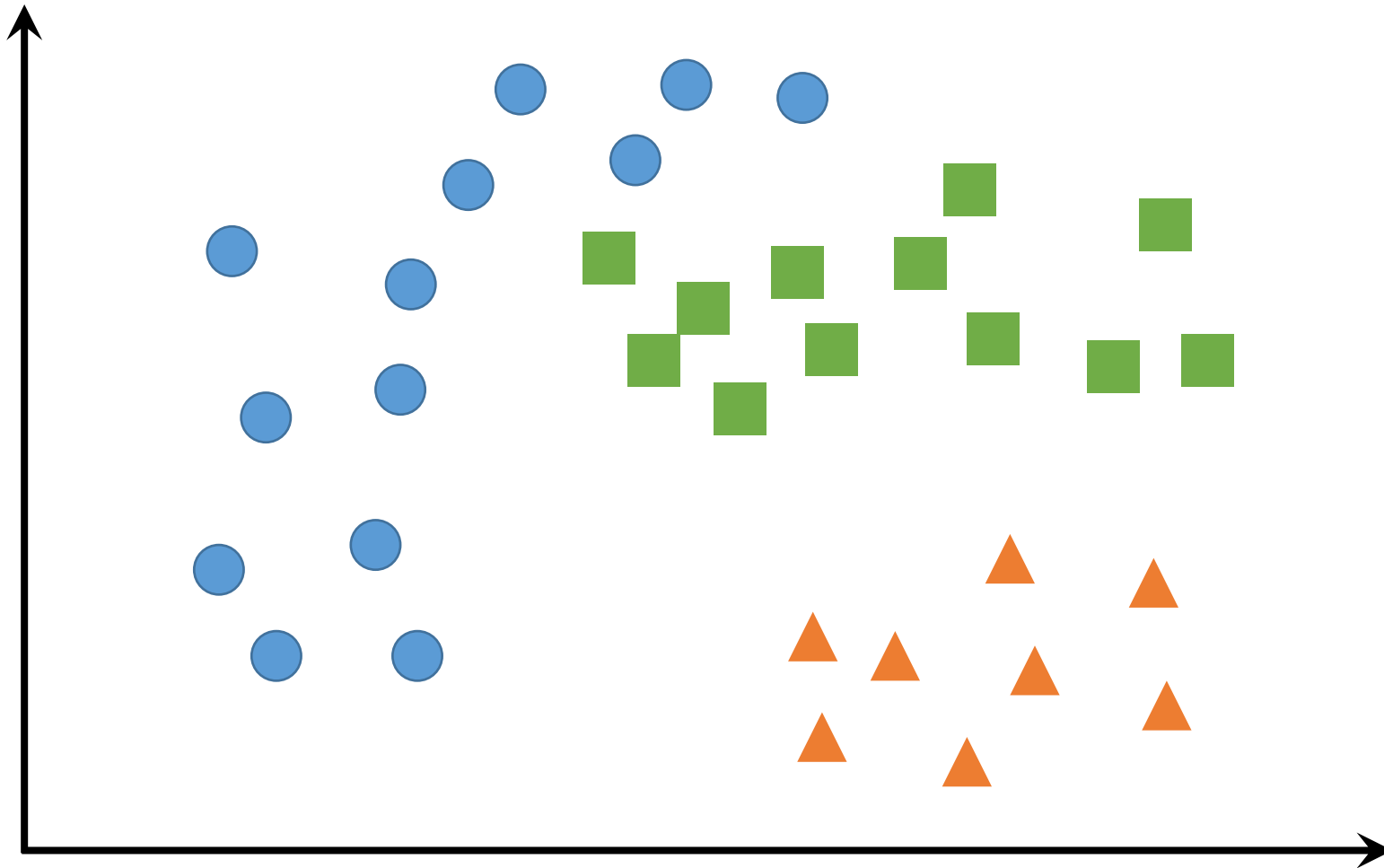
Tasks

Methods

Algorithms

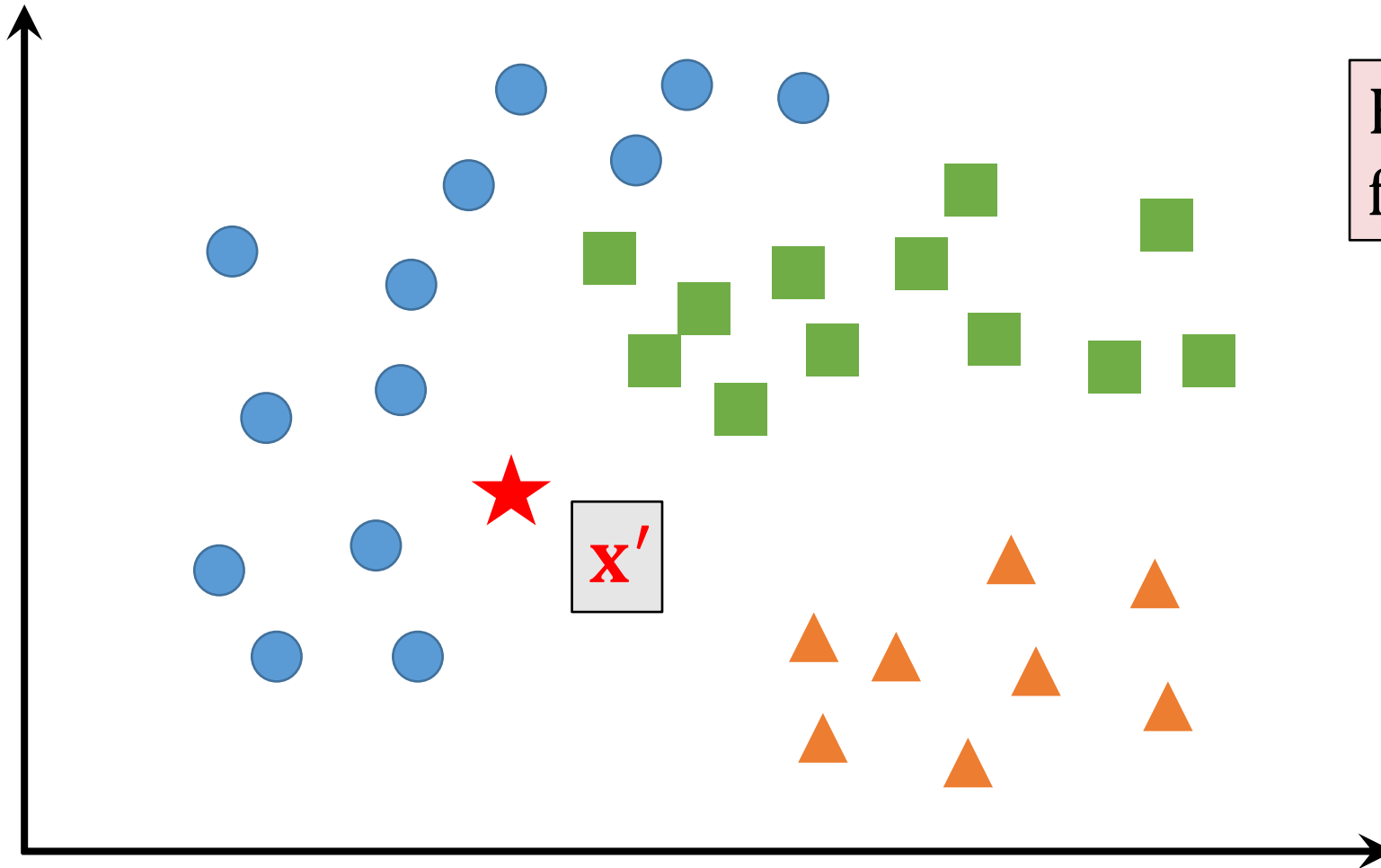
Nearest Neighbor Classifier

Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.



Nearest Neighbor Classifier

Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

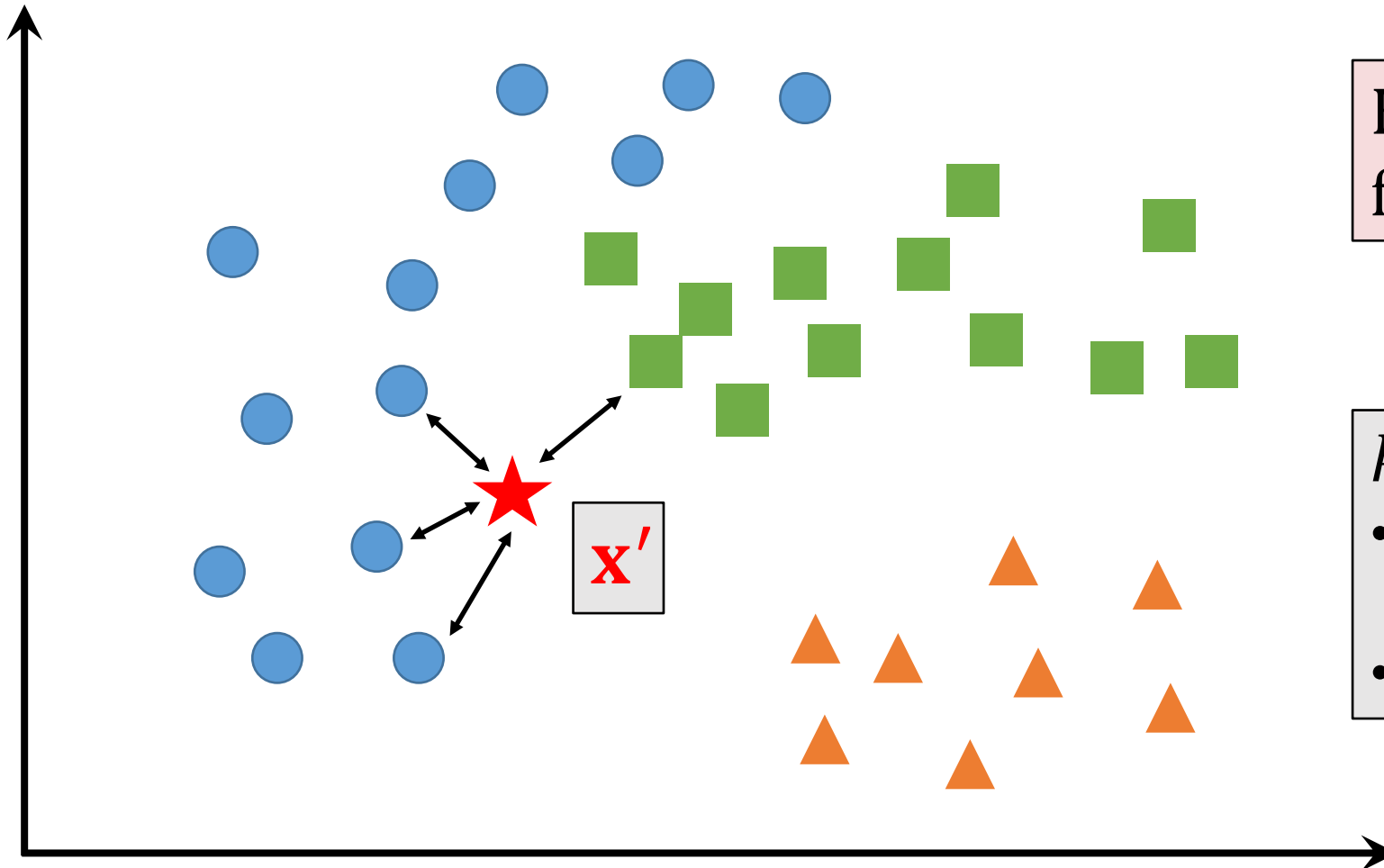


How to classify an test feature vector \mathbf{x}' ?

Nearest Neighbor Classifier

最近邻

Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.



How to classify an test feature vector \mathbf{x}' ?

- k -Nearest Neighbor (KNN):
- Find the k nearest neighbors (NN) of \mathbf{x}' .
 - Let the k NNs vote.

Nearest Neighbor Classifier

Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

k -Nearest Neighbor (KNN) classifier:

- Find the k nearest neighbors of \mathbf{x}' .
- Let the NNs vote.

Question: How to set k ?

- Treat k as hyper-parameter. K是一个超参数
- Tune k using cross-validation. 使用交叉验证来微调K

Nearest Neighbor Classifier

Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

k -Nearest Neighbor (KNN) classifier:

- Find the k nearest neighbors of \mathbf{x}' .
- Let the NNs vote.

Question: How to measure similarity? 如何测量相似性？

- 余弦相似度 • Cosine similarity: $\text{sim}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\|_2 \|\mathbf{x}'\|_2}$.
- 高斯算子 • Gaussian kernel: $\text{sim}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{\sigma^2} \|\mathbf{x} - \mathbf{x}'\|_2^2\right)$.
- 拉普拉斯算子 • Laplacian kernel: $\text{sim}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{\sigma} \|\mathbf{x} - \mathbf{x}'\|_1\right)$.

Nearest Neighbor Classifier

Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

k -Nearest Neighbor (KNN) classifier:

- Find the k nearest neighbors of \mathbf{x}' .
- Let the NNs vote.

Question: How to find the k nearest neighbors? KNN算法是一种非常特别的机器学习算法，因为它没有一般意义上的学习过程

- Naïve algorithm
 - compute all the similarities $\text{sim}(\mathbf{x}_1, \mathbf{x}'), \dots, \text{sim}(\mathbf{x}_n, \mathbf{x}')$
 - Sort the scores and find the top k .
 - $O(nd)$ time complexity (n : #samples, d : # features).
- Efficient algorithms (to be discussed later).

计算相似度

排序

时间复杂度

Nearest Neighbor Classifier

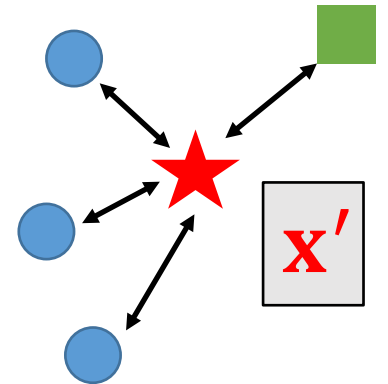
Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

k -Nearest Neighbor (KNN) classifier:

- Find the k nearest neighbors of \mathbf{x}' .
- Let the NNs **vote**.

Question: How to **vote**?

- Option 1: Every neighbor has the same weight.



Nearest Neighbor Classifier

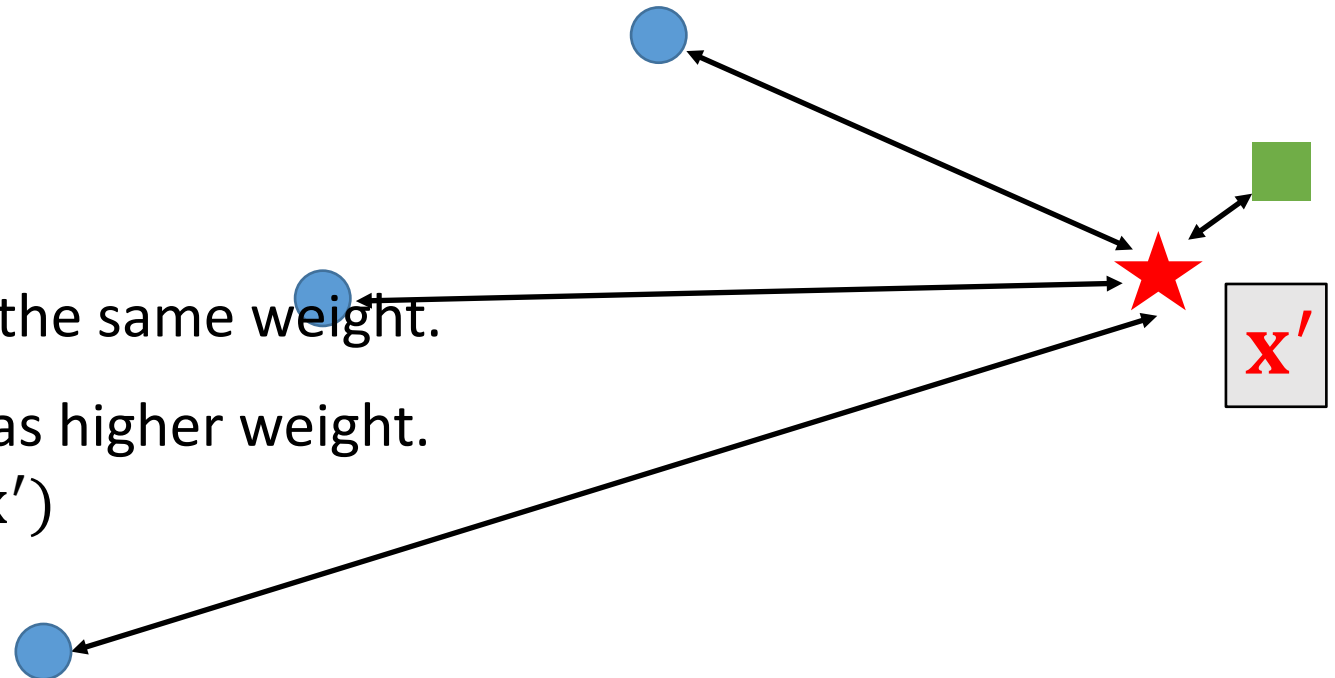
Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

k -Nearest Neighbor (KNN) classifier:

- Find the k nearest neighbors of \mathbf{x}' .
- Let the NNs **vote**.

Question: How to **vote**?

- Option 1: Every neighbor has the same weight.
- Option 2: Nearer neighbor has higher weight.
 - E.g., $\text{weight}_i = \text{sim}(\mathbf{x}_i, \mathbf{x}')$



Nearest Neighbor Classifier

Tasks

Methods

Algorithms

KNN: Naïve Algorithm

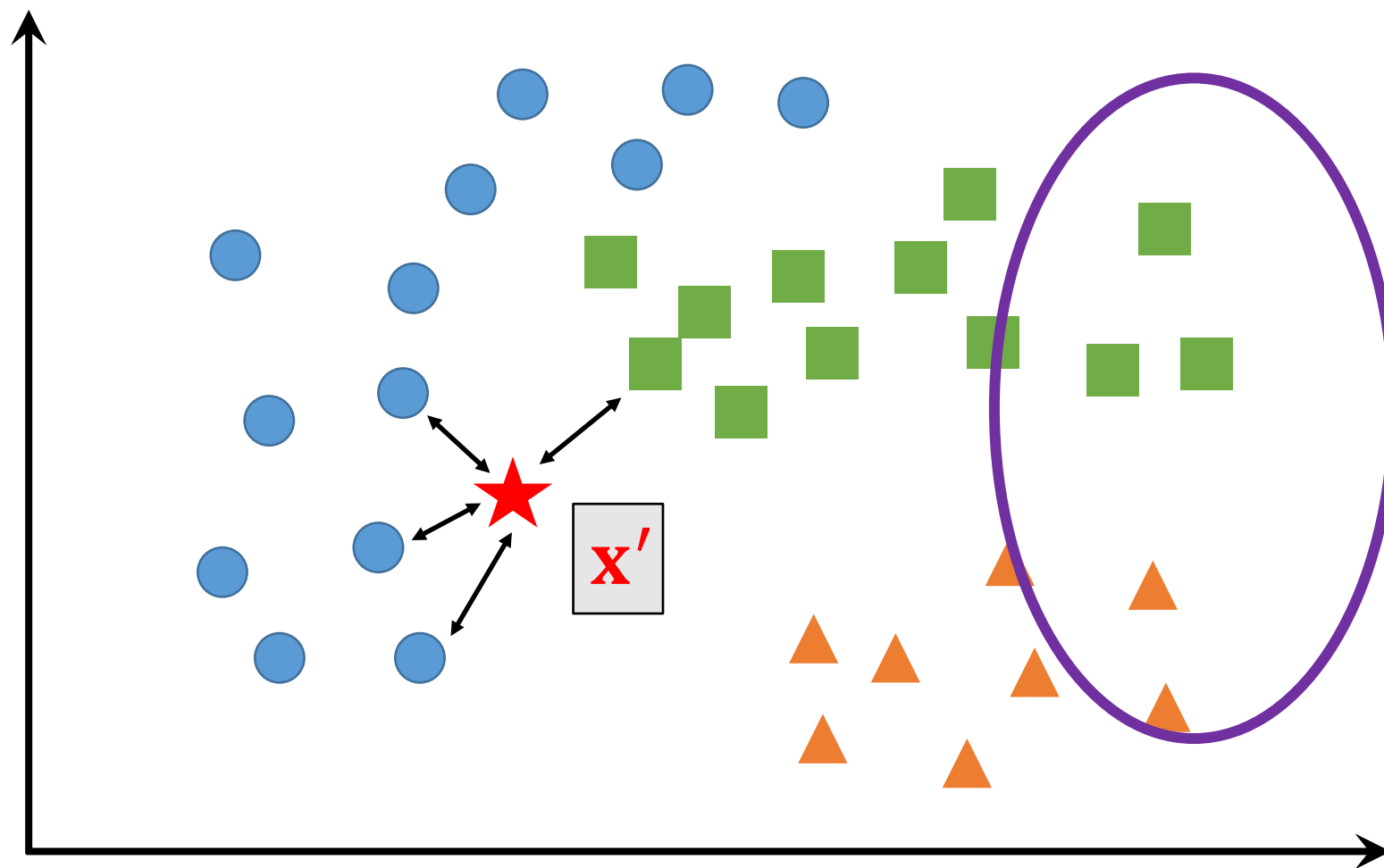
Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.

Algorithm: find the k nearest neighbors to \mathbf{x}' .

- Naïve algorithm
 - compute all the similarities $\text{sim}(\mathbf{x}_1, \mathbf{x}'), \dots, \text{sim}(\mathbf{x}_n, \mathbf{x}')$ and find the top k .
- Training: no training at all.
- Test: for each query, $O(nd)$ time complexity

KNN: Efficient Algorithm

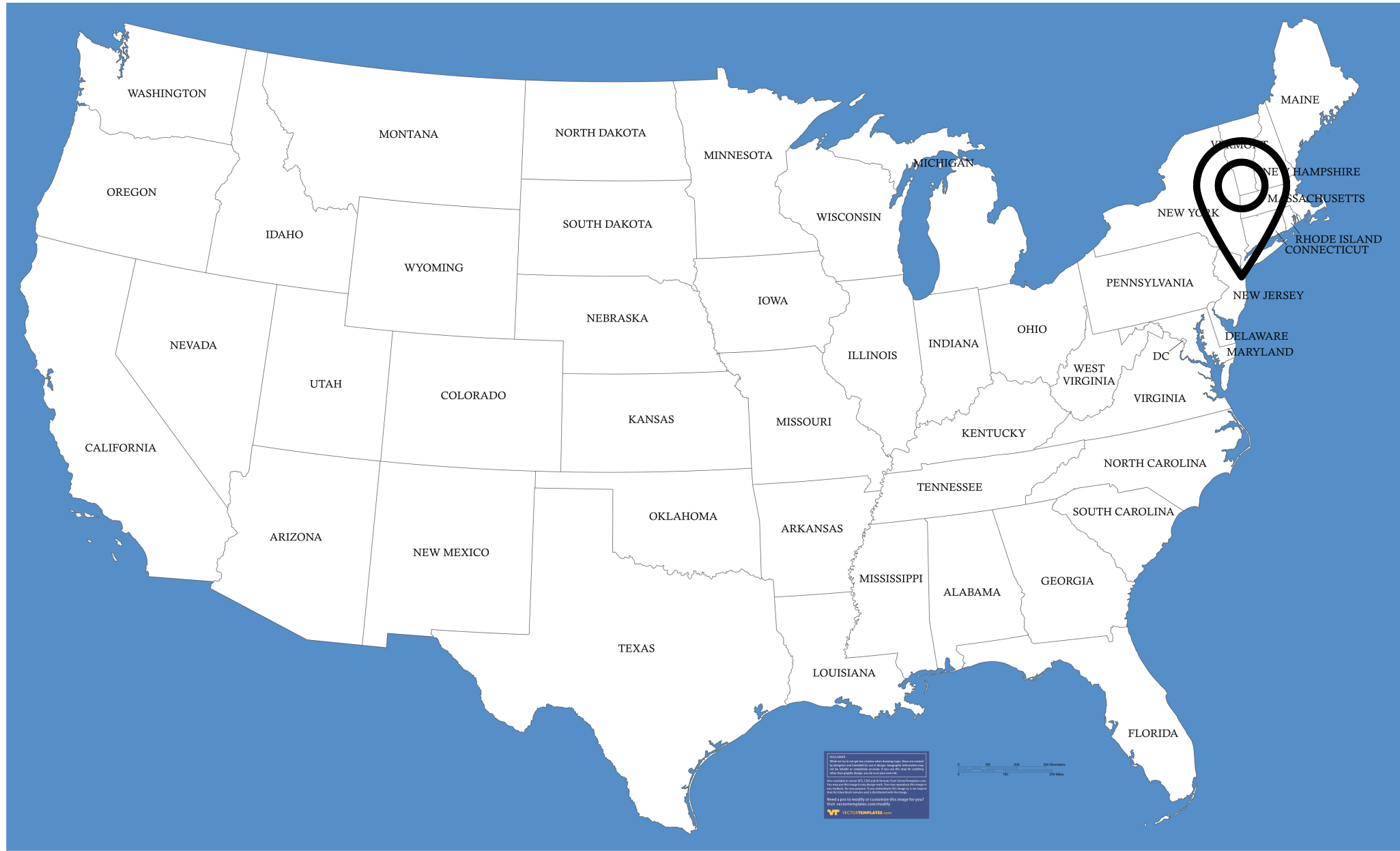
Input: feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \dots, y_n \in \mathbb{N}$.



Does the computer have to look at **this region**?

计算机一定要看这个区域吗？

Question: find your nearest post office (given longitude & latitude).



Question: find your nearest post office (given longitude & latitude).

数据总量 : 3w个 邮局的经纬度坐标

Data: $n = 30,000$ post offices' latitude and longitude:

- Post office 1: $(\text{lat}_1, \text{lon}_1)$
- Post office 2: $(\text{lat}_2, \text{lon}_2)$
- Post office 3: $(\text{lat}_3, \text{lon}_3)$
- Post office 4: $(\text{lat}_4, \text{lon}_4)$
- \vdots
- Post office n : $(\text{lat}_n, \text{lon}_n)$

Query: your own latitude and longitude:

- $(40.74627, -74.02431)$

Question: find your nearest post office (given longitude & latitude).

找到离你最近的邮局 (给定的经度和纬度)

Data: $n = 30,000$ post offices' latitude and longitude:

- Post office 1: $(\text{lat}_1, \text{lon}_1)$
- Post office 2: $(\text{lat}_2, \text{lon}_2)$
- Post office 3: $(\text{lat}_3, \text{lon}_3)$
- Post office 4: $(\text{lat}_4, \text{lon}_4)$
- \vdots
- Post office n : $(\text{lat}_n, \text{lon}_n)$

Query: your own latitude and longitude:

- $(40.74627, -74.02431)$

Question: Which is your nearest post office?

Vector Quantization for KNN

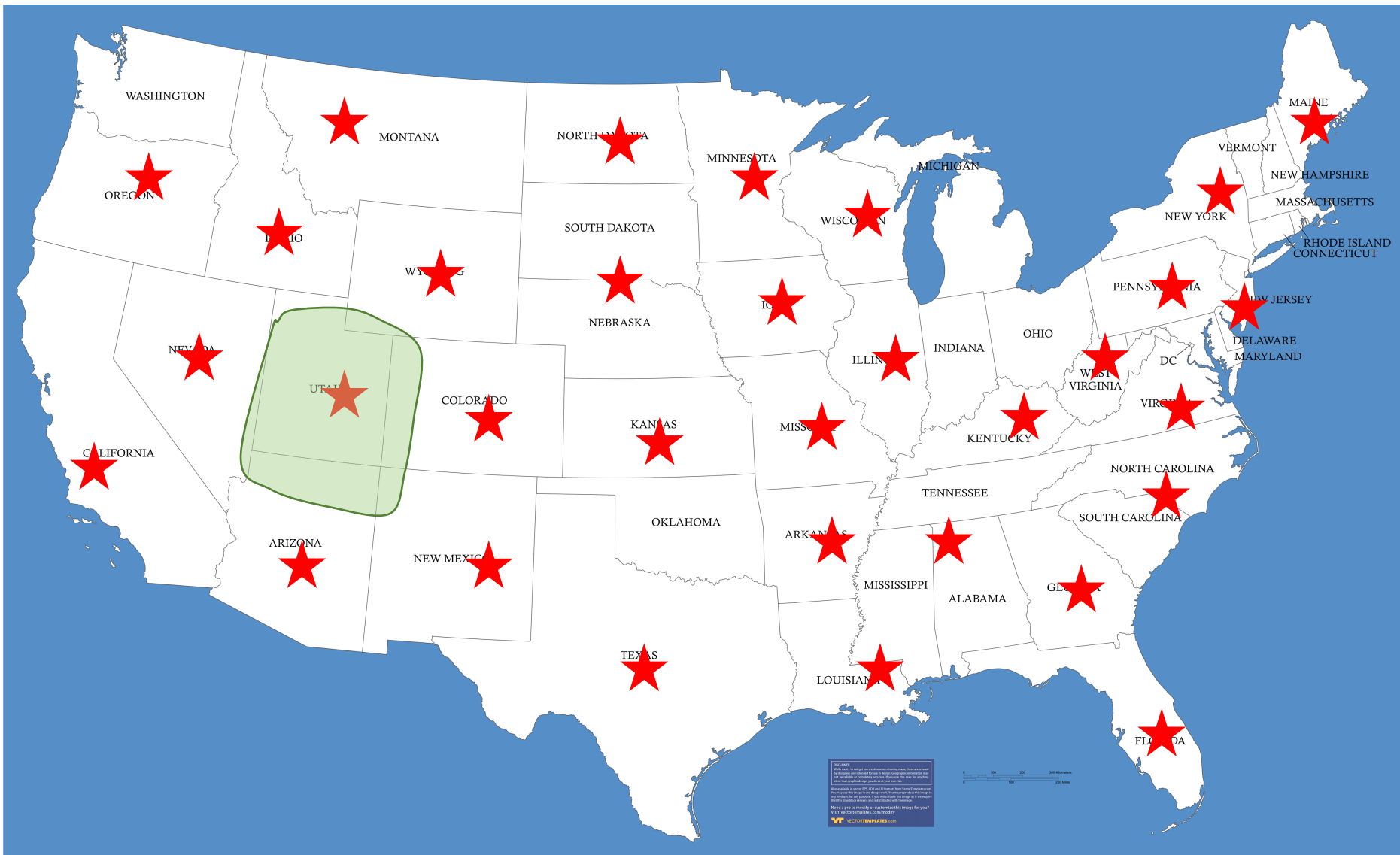


Training:

1. Vector quantization
(build **landmarks**)

1. 矢量量化 (构建地标)

Vector Quantization for KNN



Training:

1. Vector quantization (build **landmarks**)
2. Assign each post office to its nearest **landmarks**.

将每个邮局分配到其最近的地标性建筑上

Vector Quantization for KNN



Training:

1. Vector quantization (build **landmarks**)
2. Assign each post office to its nearest **landmarks**.

Test

1. Compare your location with all the **landmarks** and find the nearest **landmarks**.

比较你的位置与所有的地标，
并找到最近的地标。

Vector Quantization for KNN



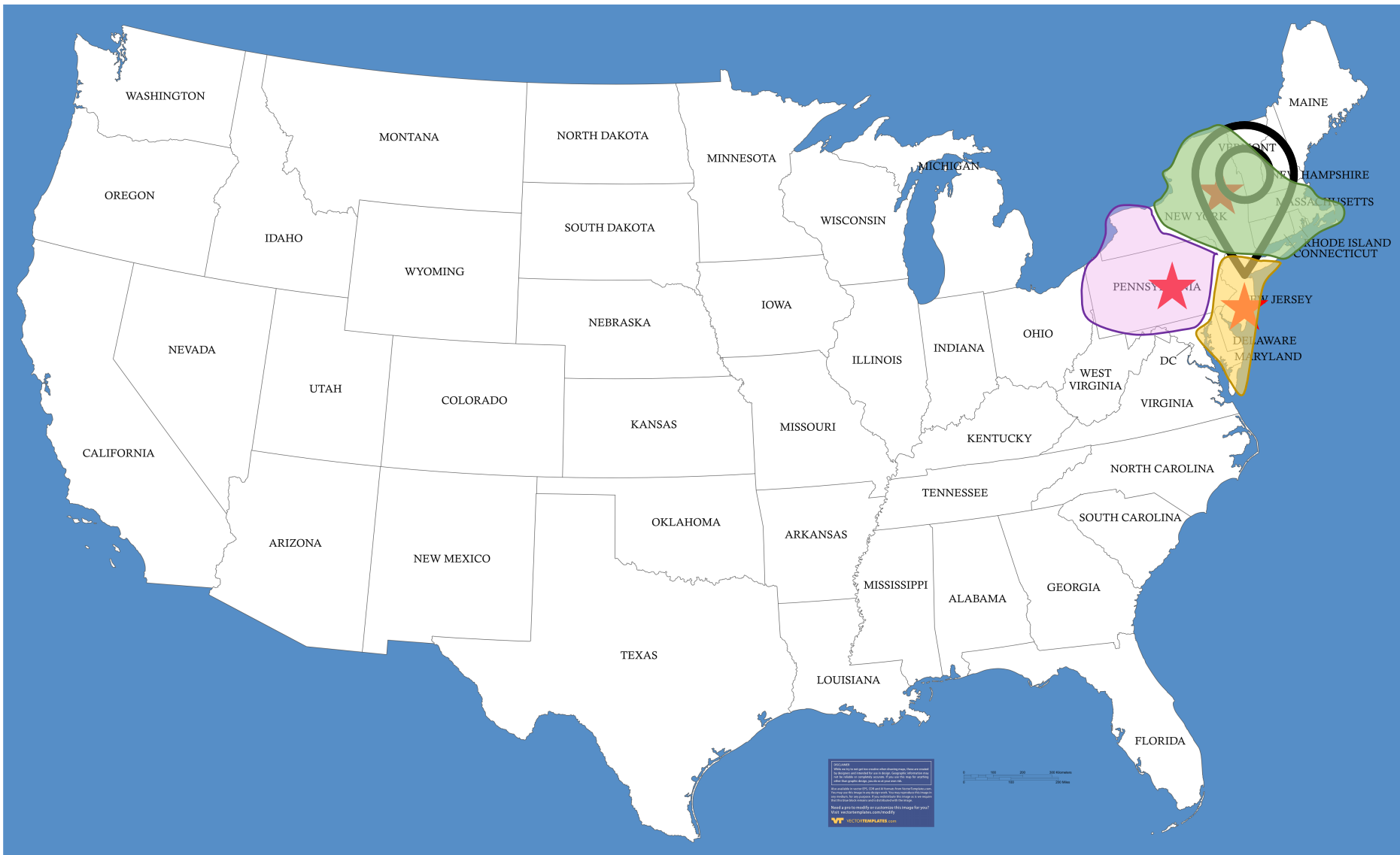
Training:

1. Vector quantization (build **landmarks**)
2. Assign each post office to its nearest **landmarks**.

Test

1. Compare your location with all the **landmarks** and find the nearest **landmarks**.

Vector Quantization for KNN



Training:

1. Vector quantization (build **landmarks**)
2. Assign each post office to its nearest **landmarks**.

Test

1. Compare your location with all the **landmarks** and find the nearest **landmarks**.
2. Compare with the postal offices assigned to the **landmarks**.

与分配给地标性的邮局进行比较。算loss

KNN: Efficient Algorithms

- Fast algorithms
 - Vector Quantization
 - KD-tree
 - Locality sensitive hashing
- More resources:
 - [KNN Search \(Wikipedia\)](#)

Summary

- KNN method for multi-class classification. KNN 适用于 多类分类
- KNN's advantage over Softmax classifier:
 - When #class is huge, Softmax classifier is expensive. 当#类别很大时, Softmax分类器是昂贵的
 - E.g., in the face recognition problem, #class can be millions.

Summary

- Training: partition the feature space to regions. 训练：将特征空间划分为区域。
- Prediction (for a test feature vector \mathbf{x}'):
 1. Find the nearest regions.
 2. Retrieve all the training feature vectors in the regions.
 3. Compare \mathbf{x}' with the retrieved feature vectors (using similarity score) and return the k nearest.
 4. Weighted/unweighted votes by the k nearest neighbors.