Nearest Neighbor Methods

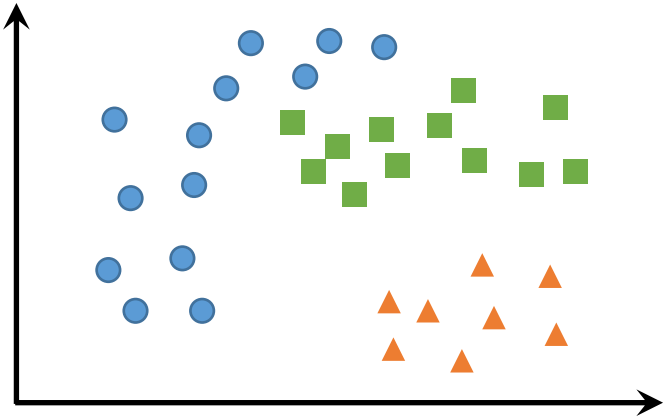
Shusen Wang

Nearest Neighbor Method for Multi-Class and Nonlinear Classification

Tasks | **Methods** | Algorithms

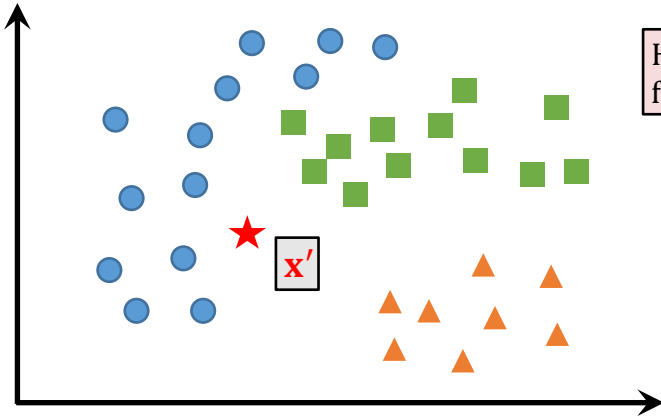
Nearest Neighbor Classifier

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



Nearest Neighbor Classifier

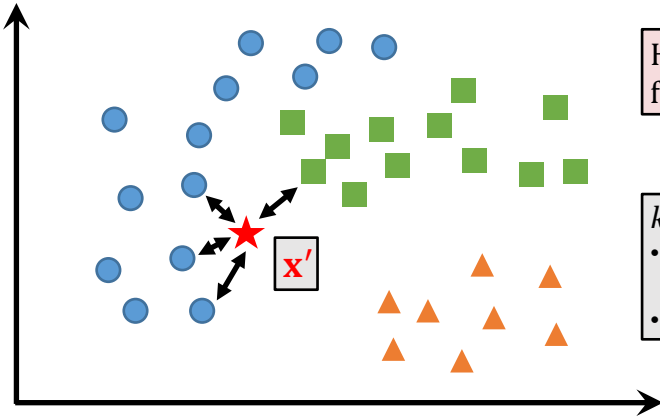
Input: feature vectors $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \cdots, y_n \in \mathbb{N}$.



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

Nearest Neighbor Classifier

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



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

k-Nearest Neighbor (KNN):

- Find the k nearest neighbors (NN) to $\mathbf{x}'$.
- Let the NNs vote.

Nearest Neighbor Classifier

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

k-Nearest Neighbor (KNN) classifier:

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

Nearest Neighbor Classifier

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

k-Nearest Neighbor (KNN) classifier:

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

How to define similarity? Examples:

- Cosine similarity: $\mathrm{sim}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{||\mathbf{x}||_2 ||\mathbf{x}'||_2}$.
- ℓ_2 similarity: $\mathrm{sim}(\mathbf{x}, \mathbf{x}') = -\left|\left|\mathbf{x} - \mathbf{x}'\right|\right|_2^2$.
- ℓ_1 similarity: $\mathrm{sim}(\mathbf{x}, \mathbf{x}') = -\left|\left|\mathbf{x} - \mathbf{x}'\right|\right|_1$.

Nearest Neighbor Classifier

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

k-Nearest Neighbor (KNN) classifier:

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

How to find the nearest neighbors?

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

Nearest Neighbor Classifier

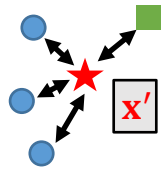
Input: feature vectors $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \cdots, y_n \in \mathbb{N}$.

k-Nearest Neighbor (KNN) classifier:

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

How to vote? Examples:

- Every neighbor has the same weight.



Nearest Neighbor Classifier

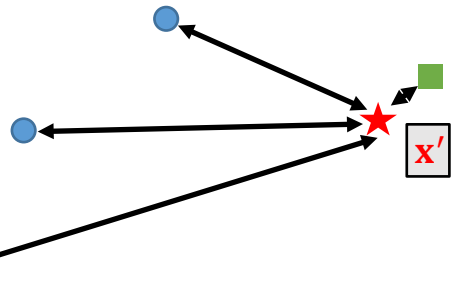
Input: feature vectors $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \cdots, y_n \in \mathbb{N}$.

k-Nearest Neighbor (KNN) classifier:

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

How to vote? Examples:

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



Nearest Neighbor Classifier

Tasks Methods **Algorithms**

KNN: Naïve Algorithm

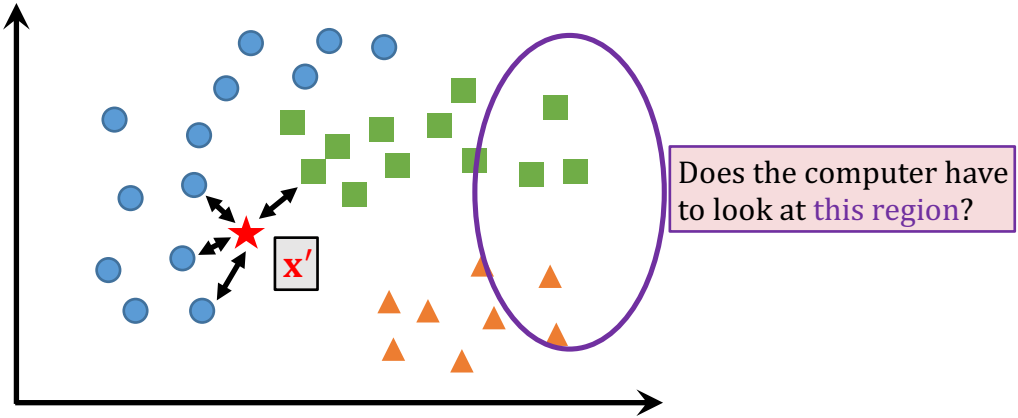
Input: feature vectors $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and labels $y_1, \cdots, y_n \in \mathbb{N}$.

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

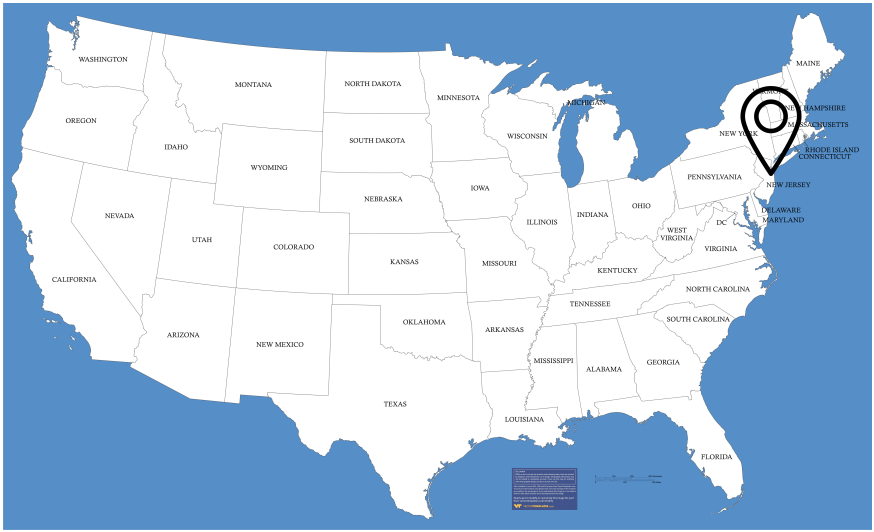
- Naïve algorithm
  - compute all the similarities $\text{sim}(\mathbf{x}_1, \mathbf{x}'), \cdots, \text{sim}(\mathbf{x}_n, \mathbf{x}')$ and find the top k.
- Training: no training at all.
- Test: for each query, $O(nd)$ time complexity

KNN: Efficient Algorithm

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



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



Vector Quantization for KNN

Training:

1. Vector quantization (build landmarks)
2. Assign each post office to one or several landmarks.

Vector Quantization for KNN

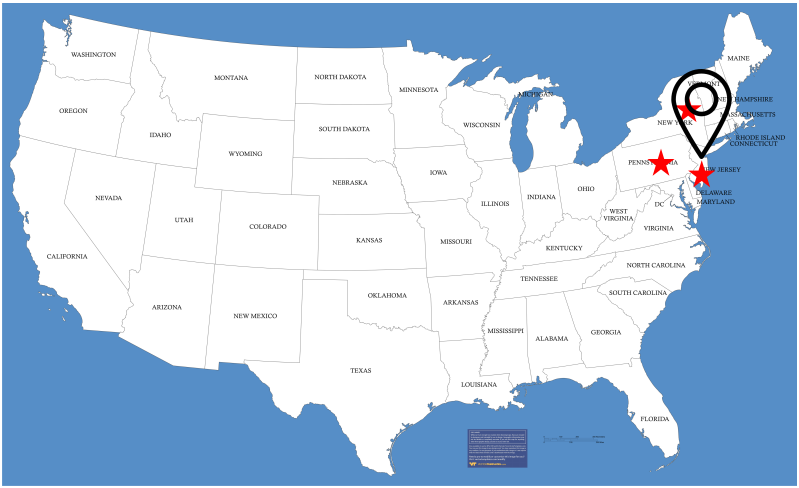
Training:

1. Vector quantization (build landmarks)
2. Assign each post office to one or several 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 one or several 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 one or several 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.

KNN: Efficient Algorithms

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

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

- Fast algorithms
  - Vector Quantization
  - KD-tree
  - Locality sensitive hashing

- More resources:
  - KNN Search (Wikipedia)