Supervised Learning

Application

New Data

Training Data

Learning Algorithm

Classifier

Predictions of Labels for new data

Our first learning algorithm

- How would you classify the next example?

kNN Algorithm (simple form)

- At “training time” do nothing.
- Store examples.

- When given new example:
  - find $k$ nearest neighbors
  - Predict $L$: majority vote of their labels

kNN Algorithm

- Theoretical basis + intuition:
  - “in the limit”, when the dataset is dense, this should pick up “all important regions”

- Very flexible classifier: no prior commitment to the shape, density, or distribution of regions

kNN: problems and extensions

- In some cases we have “noisy” labels in training data, or otherwise the label map is not smooth.

- How can we address this?
**kNN: problems and extensions**

- Expensive test time/application: because for every new example we must scan the entire dataset to find the neighbors.

- In many cases a Linear Time Scan is too expensive.

- How can we address this?

- *k* is a free parameter of the algorithm

- And different values of *k* are suitable to different datasets

- How can we choose *k* automatically?

- Completely dependent on the distance metric and representation

- E.g., Euclidean distance:
  - Different features may have different scale
  - Treats all dimensions equally
  - Sensitive to high dimension/irrelevant features

- How can we address these issues?

- So far we know how to predict one of a small number of categories – i.e., we are solving the classification problem.

- Can we adjust the algorithm to predict numerical real-valued labels? (i.e., solve the regression problem)

- Simple basic algorithm

- Has theoretical guarantees

- Adjustment of the basic scheme can make it robust and widely applicable

- Performs surprisingly well in many cases

**kNN Recap**