k Nearest Neighbor Classifier Homework

In the k-nearest neighbor folder on the class website, there are some data sets, Matlab m-files, and a Python example.

The Iris Dataset

The iris dataset consists of 150 samples, each with 4 measurements taken from different flowers. The "targets" are three different types of flowers (1, 2, or 3).

Iris Classifier-Python

Python has the iris data built-in as a sample from `sklearn`. In the example code below (available on the class website as `IrisExample.py`, we will:

- Load the data
- Split the data into training, testing sets.
- Use 5-fold cross validation to get an approximate error using 5 nearest neighbors (this is here so you can see the command).
- Repeat this process for varying number of neighbors, from 1 neighbot to 7 neighbors.
- Look at the accuracy/error obtained in each computation. Choose the number of nearest neighbors that gives the best result.
- Run the classifier with that value.
- Construct and display the confusion matrix.

Iris Classifier-Matlab

The data is given in `irisdata.mat`, and the helper files are all in the folder (StandardScaler, TrainTestSplit, fitknn, etc). The main code is in `knnapp2.m`.

In `knnapp2.m` we will:

- Load the data
- Split the data into training and testing sets.
- Split the data suitable for k-fold cross validation.
- Vary the number of nearest neighbors from 2 to 10, and use 5-fold cross validation to estimate the error each time.
- Plot the result of the training, and estimate the best number of nearest neighbors.
- Get the model output for that number
- Construct the confusion matrix.

Homework

Use the iris classifiers as templates, and construct a k-nearest neighbor classifier on wine data. For the wine data, there are 178 samples, each with 13 measurements. There are three classes for targets.

Load data in Python

```
from sklearn.datasets import load_wine

wine=load_wine()
X=wine.data
T=wine.target
```

Load data in Matlab

Be sure to download `winedata.mat` to the directory you're using. The variables X ($178x13$ matrix) and t ($178x1$ vector) will be automatically loaded when you type `load winedata`

Build the classifier (Python or Matlab)

For your classification, go through these steps:

- Load the data.
- Split the data into training and testing sets.
- Scale the data in matrix X using the standard scaler.
- Split the data suitable for k-fold cross validation.
- Vary the number of nearest neighbors from 1 to 7, and use 5-fold cross validation to estimate the error each time.
- Plot the result of the training, and estimate the best number of nearest neighbors.
- Get the model output for that number
- Construct the confusion matrix and display or print it.