Recitation 4
Classification and Regression

10-301/10-601: Introduction to Machine Learning

09/25/2020

1 k-NN

1.1 Warm Up - A Classification Example

Using the figure below, what would you categorize the green circle as with $k = 3$? $k = 5$?

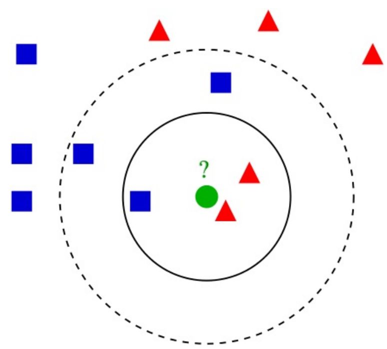


Figure 1: From wiki

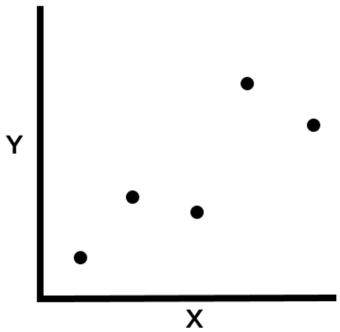
Example of k-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles.

If k = 3 (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle.

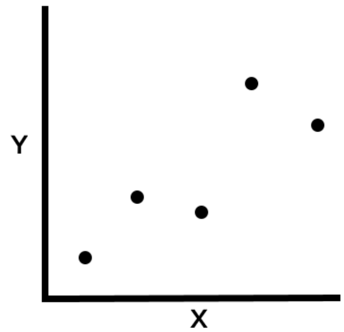
If k = 5 (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

1.2 k-NN for Regression

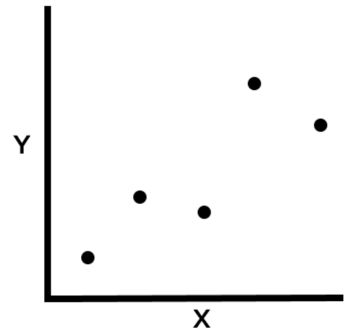
You want to predict a continuous variable Y with a continuous variable X. Having just learned k-NN, you are super eager to try it out for regression. Given the data below, draw the regression lines (what k-NN would predict Y to be for every X value if it was trained for the given data) for k-NN regression with $k = 1$, weighted $k = 2$, and unweighted $k = 2$. For weighted $k = 2$, take the weighted average of the two nearest points. For unweighted $k = 2$, take the unweighted average of the two nearest points.



(a) $k = 1$

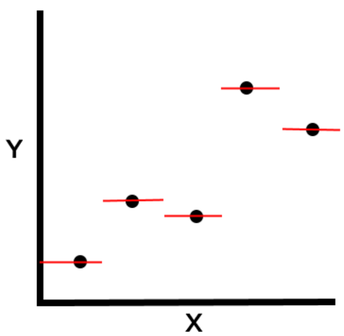


(b) weighted $k = 2$

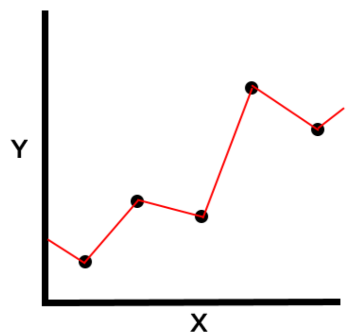


(c) unweighted $k = 2$

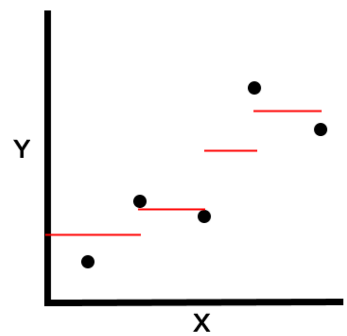
SOLUTION:



(a) $k = 1$



(b) weighted $k = 2$



(c) unweighted $k = 2$

2 Linear Regression

2.1 Defining the Objective Function

1. What does an objective function $J(\theta)$ do ? A function to measure how "good" the linear model is

2. What are some properties of this function?

   - Should be differentiable
   - Preferably convex

3. What are some examples?

   – Mean Squared Error $\frac{1}{N}\sum_{i=1}^{N} e_i^2$

   – Mean Absolute Error: $\frac{1}{N}\sum_{i=1}^{N} |e_i|$

2.2 Deriving the Closed-form Solution

We are given the following data where x is the input and y is the output:

x	1.0	2.0	3.0	4.0	5.0
y	2.0	4.0	7.0	8.0	11.0

Based on our inductive bias, we think that the linear hypothesis with no intercept should be used here. We also want to use the Mean Squared Error as our objective function: $\frac{1}{5}\sum_{i=1}^{5}(y^{(i)} - wx^{(i)})^2$, where $y^{(i)}$ is our i^{th} data point and w is our weight. Using the closed-form method, find w.

1. What is the closed-form formula for w?

$$J(w) = \frac{1}{5}\sum_{i=1}^{5}(y^{(i)} - wx^{(i)})^2$$

$$0 = \frac{dJ(w)}{dw} = \frac{1}{5}\sum_{i=1}^{5} -2x^{(i)}(y^{(i)} - wx^{(i)})$$

$$\sum_{i=1}^{5} x^{(i)}(y^{(i)} - wx^{(i)}) = 0$$
$$\sum_{i=1}^{5} x^{(i)}y^{(i)} - \sum_{i=1}^{5} w(x^{(i)})^2 = 0$$
$$w\sum_{i=1}^{5}(x^{(i)})^2 = \sum_{i=1}^{5} x^{(i)}y^{(i)}$$
$$\therefore w = \frac{\sum_{i=1}^{5} x^{(i)}y^{(i)}}{\sum_{i=1}^{5}(x^{(i)})^2}$$

2. What is the value of w?

$$\sum_{i=1}^{5} x^{(i)}y^{(i)} = 118$$
$$\sum_{i=1}^{5}(x^{(i)})^2 = 55$$
$$w = \frac{\sum_{i=1}^{5} x^{(i)}y^{(i)}}{\sum_{i=1}^{5}(x^{(i)})^2} = \frac{118}{55} = 2.15$$

We now extend the data set to include more features, $\mathbf{x} \in \mathbb{R}$:

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$
x_1	1.0	2.0	3.0	4.0	5.0
x_2	-2.0	-5.0	-6.0	-8.0	-11.0
x_3	3.0	8.0	9.0	12.0	14.0
y	2.0	4.0	7.0	8.0	11.0

We again think that the linear hypothesis with no bias should be used here. We also want to use the Mean Squared Error as our objective function:

$$\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2,$$

where $\mathbf{w} = [w_1, w_2, w_3]^T$, $\mathbf{x}^{(i)}$ is the i^{th} datapoint and $y^{(i)}$ is the i^{th} $y-$value.

1. What is the closed-form formula for w_1?

   We can re-express:

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T x^{(i)})^2$$

as

$$J(w_1, w_2, w_3) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - w_1 x_1^{(i)} - w_2 x_2^{(i)} - w_3 x_3^{(i)})^2$$

   We now take the partial derivative w.r.t w_1

$$\begin{aligned} 0 = \frac{\partial J(w_1, w_2, w_3)}{\partial w_1} &= \frac{2}{N}\sum_{i=1}^{N} -x_1^{(i)}(y^{(i)} - w_1 x_1^{(i)} - w_2 x_2^{(i)} - w_3 x_3^{(i)}) \\ &= \sum_{i=1}^{N}(x_1^{(i)} y^{(i)} - w_2 x_2^{(i)} x_1^{(i)} - w_3 x_3^{(i)} x_1^{(i)}) - \sum_{i=1}^{N} w_1 (x_1^{(i)})^2 \end{aligned}$$

$$\begin{aligned} w_1 \sum_{i=1}^{N}(x_1^{(i)})^2 &= \sum_{i=1}^{N}(x_1^{(i)} y^{(i)} - w_2 x_2^{(i)} x_1^{(i)} - w_3 x_3^{(i)} x_1^{(i)}) \\ \therefore w_1 &= \frac{\sum_{i=1}^{N}(x_1^{(i)} y^{(i)} - w_2 x_2^{(i)} x_1^{(i)} - w_3 x_3^{(i)} x_1^{(i)})}{\sum_{i=1}^{N}(x_1^{(i)})^2} \end{aligned}$$

   Notice that to solve for w_1, we need w_2 and w_3 and if you observe the equation, to solve for w_3, we need w_1 and w_2 etc. We can actually these 3 equations and 3 unknowns as a series of simultaneous equations.

2. What is the closed-form matrix solution for $\mathbf{w}$?

   We will not go over the derivation now, but there is a convenient matrix solution for $\mathbf{w}$:

$$\begin{aligned} \hat{\mathbf{w}} &= \text{argmin } J(\mathbf{w}) \\ &= (X^T X)^{-1} X^T Y \end{aligned}$$

   The design matrix X is given by:

$$\begin{bmatrix} 1 & -2 & 3 \\ 2 & -5 & 8 \\ 3 & -6 & 9 \\ 4 & -8 & 12 \\ 5 & -11 & 14 \end{bmatrix}$$

and

$$y = \begin{bmatrix} 2 \\ 4 \\ 7 \\ 8 \\ 11 \\ \end{bmatrix}$$

Using the closed-form formula in class, we get

$$\mathbf{w} = \begin{bmatrix} 2.36 \\ -0.205 \\ -0.218 \end{bmatrix}$$

3 Gradient Descent

3.1 Solving Linear Regression using Gradient Descent

We use the same data set from last section. However, we want to implement the gradient descent method.

Assuming that $\alpha = 0.1$ and $\mathbf{w}$ has been initialized to $[0, 0, 0]^T$, perform one iteration of gradient descent:

1. What is the gradient of the objective function , $J(\mathbf{w})$, w.r.t $\mathbf{w}$: $\nabla_{\mathbf{w}} J(\mathbf{w})$

$$\frac{dJ(\mathbf{w})}{dw_k} = \frac{1}{5}\sum_{i=1}^{5} -2x_k^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)})$$

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \begin{pmatrix} \frac{dJ(\mathbf{w})}{dw_1} \\ \frac{dJ(\mathbf{w})}{dw_2} \\ \frac{dJ(\mathbf{w})}{dw_3} \end{pmatrix} = \begin{pmatrix} \frac{1}{5}\sum_{i=1}^{5} -2x_1^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)}) \\ \frac{1}{5}\sum_{i=1}^{5} -2x_2^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)}) \\ \frac{1}{5}\sum_{i=1}^{5} -2x_3^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)}) \end{pmatrix}$$

2. How do we carry out the update rule?

   We initialize:

$$\mathbf{w} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

   Follow the update rule:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \underbrace{\alpha}_{\text{“Cross-validated”}} \nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}^{(k)}} J(\mathbf{w})$$

   , where $k = 0$ here

$$\frac{1}{5}\sum_{i=1}^{5} -2x^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)}) = \frac{-2}{5}\cdot(2+8+21+32+55)$$
$$= -47.2$$
$$\frac{1}{5}\sum_{i=1}^{5} -2x^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)}) = \frac{-2}{5}\cdot(-4-20-42-64-121)$$
$$= 100.4$$
$$\frac{1}{5}\sum_{i=1}^{5} -2x^{(i)}(y^{(i)} - \sum_{j=1}^{3} w_j x_j^{(i)}) = \frac{-2}{5}\cdot(6+32+63+96+154)$$
$$= -140.4$$

$$\therefore \mathbf{w}^{(1)} = \mathbf{w}^{(0)} + \lambda \nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}^{(0)}} J(\mathbf{w})$$
$$= \begin{pmatrix}0\\0\\0\end{pmatrix} + 0.1\begin{pmatrix}-47.2\\100.4\\-140.4\end{pmatrix}$$
$$= \begin{pmatrix}-4.72\\10.4\\-14.4\end{pmatrix}$$

***Convexity of objective function ensures that the local min(max) of the function is the global min(max).**

4 Decision Trees and Beyond

1. **Decision Tree Classification with Continuous Attributes**
   Given the dataset $\mathcal{D}_1 = \{\mathbf{x}^{(i)}, y\}_{i=1}^{N}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^2, y \in \{\text{Yellow}, \text{Purple}, \text{Green}\}$ as shown in Fig. 4, we wish to learn a decision tree for classifying such points. Provided with a possible tree structure in Fig. 4, what values of α, β and leaf node predictions could we use to perfectly classify the points? Now, draw the associated decision boundaries on the scatter plot.

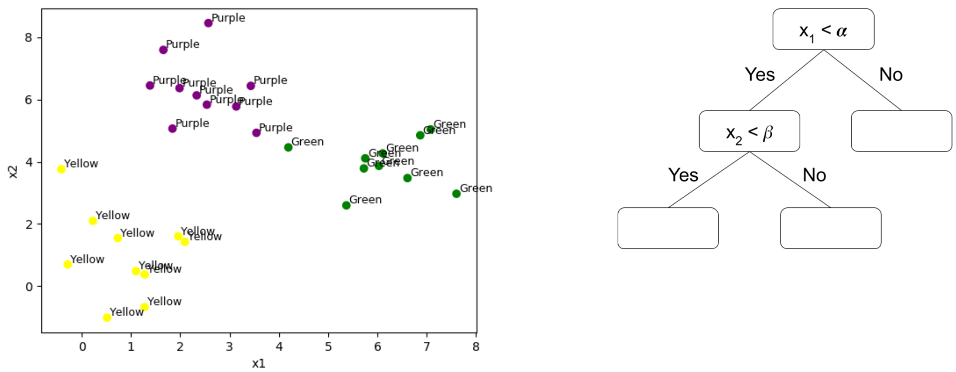
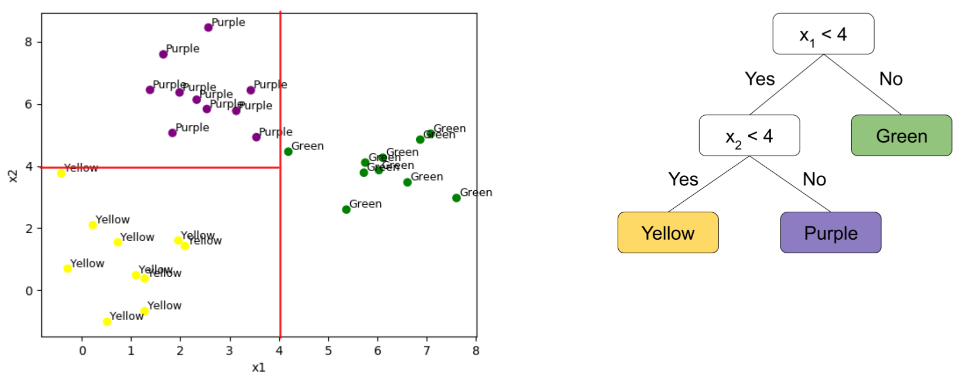


Figure 4: Classification of 2D points, with Decision Tree to fill in

Solution:



Note how our decision tree actually creates partitions in the 2D space of points, and each partition is associated with one predicted class. If we had trees of larger maximum depth, we gain the ability to create even more fine-grained partitions of the feature space, resulting in greater flexibility of predictions.

Decision Tree Regression with Continuous Attributes

Now instead if we had dataset $\mathcal{D}_2 = \{\mathbf{x}^{(i)}, y\}_{i=1}^{N}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^2, y \in \mathbb{R}$ as shown in Fig. 5, we wish to learn a decision tree for regression on such points. Using the same tree structure and values of α, β as before, what values should each leaf node predict to minimize the training Mean Squared Error (MSE) of our regression? Assume each leaf node just predicts a constant.

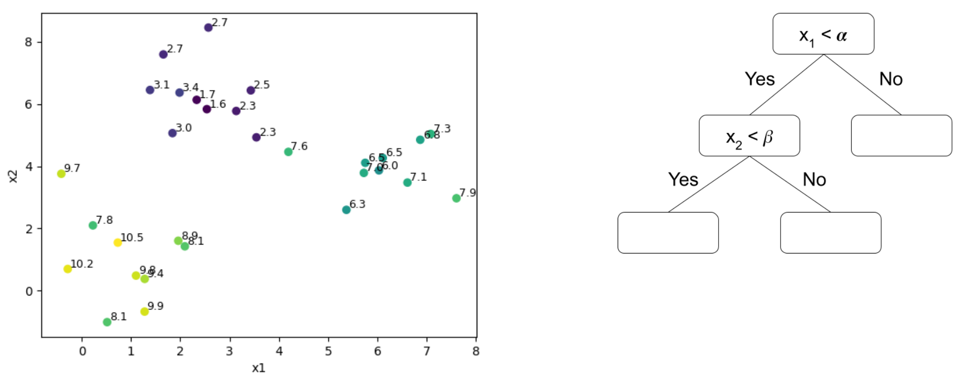
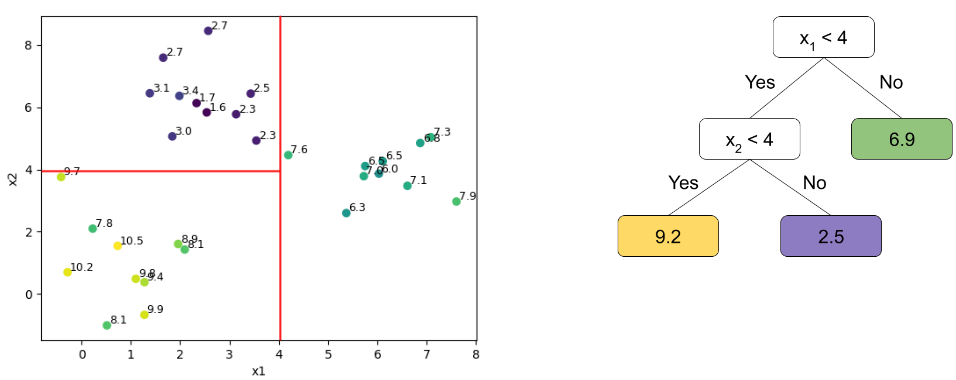


Figure 5: Regression on 2D points, with Decision Tree to fill in

Solution:



In this example we see how decision trees can be used for regressions too. Since we already know the splits, we partition up the feature space in the same way as before where each partition yields a single constant as a prediction. Instead of predicting a class, we want to predict a real number for each partition that will minimize our metric, MSE. The mean value of y in each partition will be the prediction that minimizes MSE.

2. **Feature engineering - A good model isn't everything.** Given a two-dimensional dataset shown in Fig.6, where red circles are positive examples and blue circles are negative examples. Which of the following model would you choose to classify the samples using the two coordinates as features?

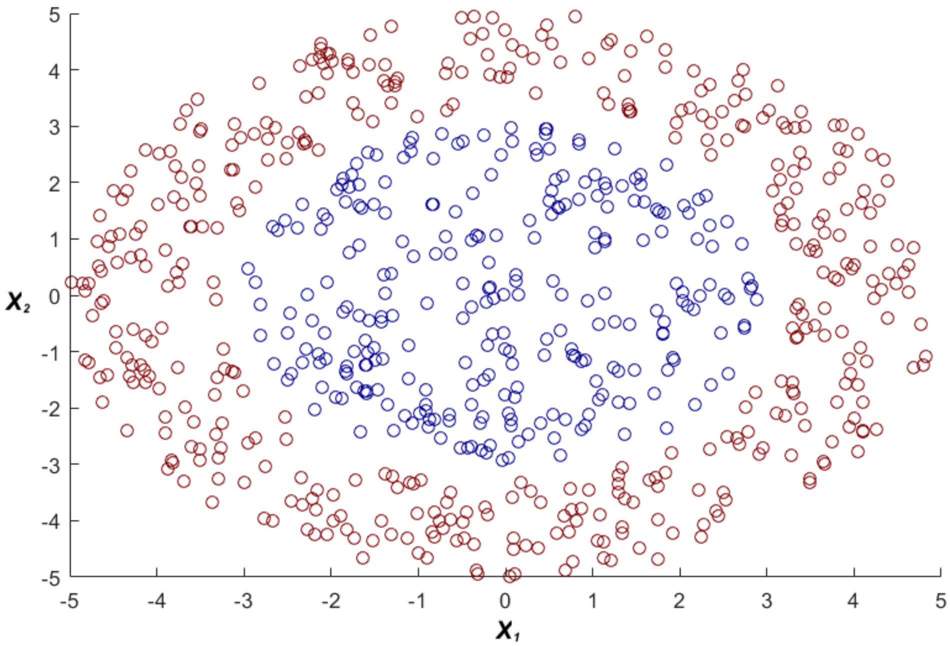


Figure 6

Which model would you choose for this classification task? (There is no best answer here)

☐ Linear regression on $\mathbf{x}$ and y

☐ Decision tree using attribute $\mathbf{x}$ and label y trained with ID3 algorithm

☐ kNN with $k = 5$ (any distance metric)

Solution:
Linear regresssion is clearly incapable of this task, other models can be used to perform reasonably well
But the problem itself has a perfect classification solution if one simply transform the features like $r = \sqrt{x^2 + y^2}$, then almost every model here can do a perfect job.
The idea of this question is not to arrive at a correct answer, but to inspire people to understand the importance of good features.

5 Summary

5.1 k-NN

Pros	Cons	Inductive bias	When to use
• Simple, minimal assumptions made about data distribution • No training of parameters • Can apply to multi-class problems and use different metrics	• Becomes slow as dataset grows • Requires homogeneous features • Selection of k is tricky • Imbalanced data can lead to misleading results • Sensitive to outliers	• Similar (i.e. nearby) points should have similar labels • All label dimensions are created equal	• Small dataset • Small dimensionality • Data is clean (no missing data) • Inductive bias is strong for dataset

5.2 Linear regression

Pros	Cons	Inductive bias	When to use
• Easy to understand and train • Closed form solution	• Sensitive to noise (other than zero-mean Gaussian noise)	• The relationship between the inputs x and output y is linear. i.e. hypothesis space is Linear Functions	• Most cases (can be extended by adding non-linear feature transformations)

5.3 Decision Tree

Pros	Cons	Inductive bias	When to use
• Easy to understand and interpret • Very fast for inference	• Tree may grow very large and tend to overfit. • Greedy behaviour may be sub-optimal	• Prefer the smallest tree consistent w/ the training data (i.e. 0 error rate)	• Most cases. Random forests are widely used in industry.