Non Parametric Methods

COMP9417, 23T1

1. Non Parametric Methods
2. Decision Trees
3. k-NN
4. Linear Smoothing

Section 1

Non Parametric Methods

Non Parametric Methods

Parametric modelling

We make assumptions on the type of function which our data takes.

- Linear regression
- Perceptron
- Logistic regression

Non parametric modelling

We make no assumptions on the underlying function and purely use our datapoints as guides for pattern inference.

- k-Nearest neighbours
- Local regression
- Decision Trees

Section 2

Decision Trees

Decision Trees

A tree-like model used for both **regression** and **classification**.

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

- Interpretable
- Useful when used in ensemble learning (we'll come back to this notion)

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

- Interpretable
- Useful when used in ensemble learning (we'll come back to this notion)

Disadvantages:

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

- Interpretable
- Useful when used in ensemble learning (we'll come back to this notion)

Disadvantages:

- Tend to overfit data
- Often innacurate in their most basic form

Entropy

Entropy essentially measures the *uncertainty* or *surprise* of a random variable.

We define the entropy for a set S,

$$H(S) = \sum_{x \in X} -p(x) \log p(x)$$

where $p(x)$ represents the *proportion* of x in S.

Entropy

Entropy essentially measures the *uncertainty* or *surprise* of a random variable.

We define the entropy for a set S,

$$H(S) = \sum_{x \in X} -p(x) \log p(x)$$

where $p(x)$ represents the *proportion* of x in S.

Say we have a random variable $X \sim \text{Bernoulli}(p)$. We can define the entropy of X:

Entropy

Entropy essentially measures the *uncertainty* or *surprise* of a random variable.

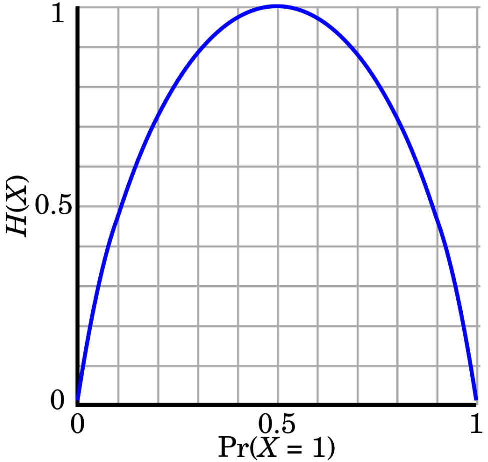
We define the entropy for a set S,

$$H(S) = \sum_{x \in X} -p(x) \log p(x)$$

where $p(x)$ represents the *proportion* of x in S.

Say we have a random variable $X \sim \text{Bernoulli}(p)$. We can define the entropy of X:

$$H(x) = -(1-p)\log(1-p) - p\log p$$


1
0.5
0
$H(X)$
0
0.5
1
$\Pr(X = 1)$

Gain

To measure the *information* we gain by splitting on an attribute A for a dataset S, we define:

Gain

To measure the *information* we gain by splitting on an attribute A for a dataset S, we define:

$$\text{Gain}(S, A) = \text{Current entropy} - \text{Entropy if we split on A}$$

Gain

To measure the *information* we gain by splitting on an attribute A for a dataset S, we define:

$$\text{Gain}(S, A) = \text{Current entropy} - \text{Entropy if we split on A}$$

If we have a dataset S with a feature A,

$$\text{Gain}(S, A) = H(S) - \sum_{v \in V_A} \frac{|S_v|}{|S|} H(S_v)$$

Basic Example

Say we have a dataset as follows: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

Basic Example

Say we have a dataset as follows: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$
\begin{aligned}
H(S) &= \sum_{x \in X} -p(x) \log p(x) \\
&= -\frac{29}{29+35} \log\left(\frac{29}{29+35}\right) - \frac{35}{29+35} \log\left(\frac{35}{29+35}\right) \\
&= 0.9936
\end{aligned}
$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$

$$H(S) = 0.9936$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$

$$H(S) = 0.9936$$

$$\begin{aligned} H(S_{A_{1,T}}) &= -\frac{21}{26}\log\left(\frac{21}{26}\right) - \frac{5}{26}\log\left(\frac{5}{26}\right) \\ &= 0.7063 \end{aligned}$$

Dataset: $[29+, 35-]$:

- A1 ~ T: $[21+, 5-]$ F: $[8+, 30-]$

$$H(S) = 0.9936$$

$$\begin{aligned} H(S_{A_{1,T}}) &= -\frac{21}{26}\log\left(\frac{21}{26}\right) - \frac{5}{26}\log\left(\frac{5}{26}\right) \\ &= 0.7063 \end{aligned}$$

$$\begin{aligned} H(S_{A_{1,F}}) &= -\frac{8}{38}\log\left(\frac{8}{38}\right) - \frac{30}{38}\log\left(\frac{30}{38}\right) \\ &= 0.7425 \end{aligned}$$

Dataset: $[29+, 35-]$:

- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$H(S) = 0.9936$$
$$H(S_{A_{1,T}}) = 0.7063$$
$$H(S_{A_{1,F}}) = 0.7425$$

Dataset: $[29+, 35-]$:

- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$H(S) = 0.9936$$
$$H(S_{A_{1,T}}) = 0.7063$$
$$H(S_{A_{1,F}}) = 0.7425$$

$$\begin{aligned} H(S_{A_{2,T}}) &= -\frac{18}{51}\log\left(\frac{18}{51}\right) - \frac{33}{51}\log\left(\frac{33}{51}\right) \\ &= 0.9366 \end{aligned}$$

Dataset: $[29+, 35-]$:

- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$H(S) = 0.9936$$
$$H(S_{A_{1,T}}) = 0.7063$$
$$H(S_{A_{1,F}}) = 0.7425$$

$$\begin{aligned} H(S_{A_{2,T}}) &= -\frac{18}{51}\log\left(\frac{18}{51}\right) - \frac{33}{51}\log\left(\frac{33}{51}\right) \\ &= 0.9366 \end{aligned}$$

$$\begin{aligned} H(S_{A_{2,F}}) &= -\frac{11}{13}\log\left(\frac{11}{13}\right) - \frac{2}{13}\log\left(\frac{2}{13}\right) \\ &= 0.4674 \end{aligned}$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$H(S) = 0.9936$$
$$H(S_{A_{1,T}}) = 0.7063$$
$$H(S_{A_{1,F}}) = 0.7425$$
$$H(S_{A_{2,T}}) = 0.9366$$
$$H(S_{A_{2,F}}) = 0.4674$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$H(S) = 0.9936$$
$$H(S_{A_{1,T}}) = 0.7063$$
$$H(S_{A_{1,F}}) = 0.7425$$
$$H(S_{A_{2,T}}) = 0.9366$$
$$H(S_{A_{2,F}}) = 0.4674$$

$$\text{Gain}(S, A_1) = H(S) - \sum_{v \in \{T,F\}} \frac{|A_{1,v}|}{|S|} H(A_{1,v})$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$
\begin{aligned}
H(S) &= 0.9936 \\
H(S_{A_{1,T}}) &= 0.7063 \\
H(S_{A_{1,F}}) &= 0.7425 \\
H(S_{A_{2,T}}) &= 0.9366 \\
H(S_{A_{2,F}}) &= 0.4674
\end{aligned}
$$

$$
\text{Gain}(S, A_1) = H(S) - \sum_{v \in \{T,F\}} \frac{|A_{1,v}|}{|S|} H(A_{1,v})
$$

$$
\begin{aligned}
&= H(S) - \frac{26}{64} H(A_{1,T}) - \frac{38}{64} H(A_{1,F}) \\
&= 0.2658
\end{aligned}
$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$\text{Gain}(S, A_1) = 0.2658$$

$$H(S) = 0.9936$$
$$H(S_{A_{1,T}}) = 0.7063$$
$$H(S_{A_{1,F}}) = 0.7425$$
$$H(S_{A_{2,T}}) = 0.9366$$
$$H(S_{A_{2,F}}) = 0.4674$$

Dataset: $[29+, 35-]$:

- A1 $\sim$ T: $[21+, 5-]$ F: $[8+, 30-]$
- A2 $\sim$ T: $[18+, 33-]$ F: $[11+, 2-]$

$$
\begin{aligned}
H(S) &= 0.9936 \\
H(S_{A_{1,T}}) &= 0.7063 \\
H(S_{A_{1,F}}) &= 0.7425 \\
H(S_{A_{2,T}}) &= 0.9366 \\
H(S_{A_{2,F}}) &= 0.4674
\end{aligned}
$$

$$\text{Gain}(S, A_1) = 0.2658$$

$$
\begin{aligned}
\text{Gain}(S, A_2) &= H(S) - \frac{51}{64} H(A_{2,T}) - \frac{13}{64} H(A_{2,F}) \\
&= 0.1643
\end{aligned}
$$

ID3 Algorithm

Basically what we just did:

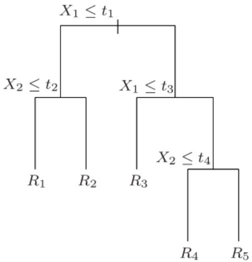
- Calculate the entropy for each attribute $a \in A$.
- Split on the attribute with the maximum Gain. This means creating a decision tree node using that attribute.
- Recurse on this new subset of the data.

Regression Trees

Regression trees split the dataset up into regions and fit separate models to each region.

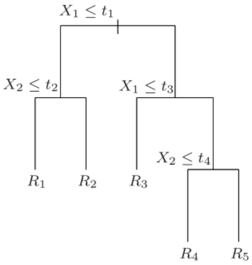
Regression Trees

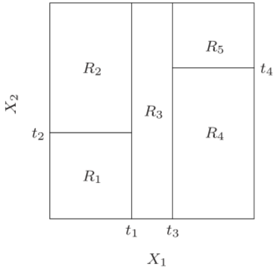
Regression trees split the dataset up into regions and fit separate models to each region.



Regression Trees

Regression trees split the dataset up into regions and fit separate models to each region.





If we define our two regions as $R_1(j,s) = \{X | X_j \leq s\}$ and $R_2(j,s) = \{X | X_j > s\}$.
We can find optimal regions with the formula:

If we define our two regions as $R_1(j,s) = \{X|X_j \leq s\}$ and $R_2(j,s) = \{X|X_j > s\}$. We can find optimal regions with the formula:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i - c_1)^2 + \sum_{x_i\in R_2(j,s)}\min_{c_2}(y_1 - c_2)^2\right]$$

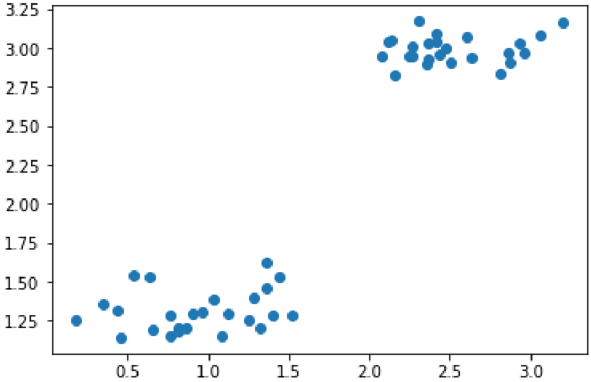
Where $\hat{c_1} = \text{ave}(y_i|x_i \in R_1), \quad \hat{c_2} = \text{ave}(y_i|x_i \in R_2)$.

If we define our two regions as $R_1(j,s) = \{X|X_j \leq s\}$ and $R_2(j,s) = \{X|X_j > s\}$. We can find optimal regions with the formula:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i - c_1)^2 + \sum_{x_i\in R_2(j,s)}\min_{c_2}(y_1 - c_2)^2\right]$$

Where $\hat{c}_1 = \text{ave}(y_i|x_i \in R_1), \quad \hat{c}_2 = \text{ave}(y_i|x_i \in R_2)$.

This essentially finds regions (R_1 and R_2) with the minimum variance.



$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\sum_{x_i\in R_2(j,s)}\min_{c_2}(y_1-c_2)^2\right]$$

Section 3

k-NN

k-NN

We predict $\hat{y}_i$ for a point x_i to be the average of the k-nearest points.

k-NN

We predict $\hat{y}_i$ for a point x_i to be the average of the k-nearest points.

Regression If we define the set K as the k-nearest neighbours of a point X_i, then our k-NN estimate is:

$$\hat{y}_i = \frac{1}{k} \sum_{i=1}^{n} \mathbf{1}\{X_i \in K\} y_i$$

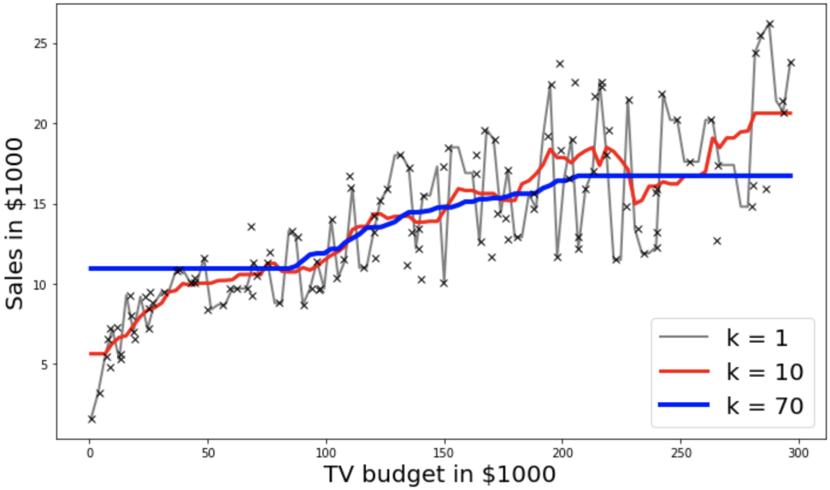
k-NN

We predict $\hat{y}_i$ for a point x_i to be the average of the k-nearest points.

Regression If we define the set K as the k-nearest neighbours of a point X_i, then our k-NN estimate is:

$$\hat{y}_i = \frac{1}{k}\sum_{i=1}^{n} \mathbf{1}\{X_i \in K\} y_i$$

Classification we assign X_i the majority class in K.


Sales in $1000
TV budget in $1000
k = 1
k = 10
k = 70
0
50
100
150
200
250
300
5
10
15
20
25

An obvious limitation:

An obvious limitation:

We need lots of relevant data for accurate predictions.

An obvious limitation:

We need lots of relevant data for accurate predictions.

A not-so obvious limitation:

An obvious limitation:

We need lots of relevant data for accurate predictions.

A not-so obvious limitation:

Curse of dimensionality.

Curse of Dimensionality

Certain phenomena occur when we increase the number of dimensions (i.e features) in our problem.

The most common are:

Curse of Dimensionality

Certain phenomena occur when we increase the number of dimensions (i.e features) in our problem.

The most common are:

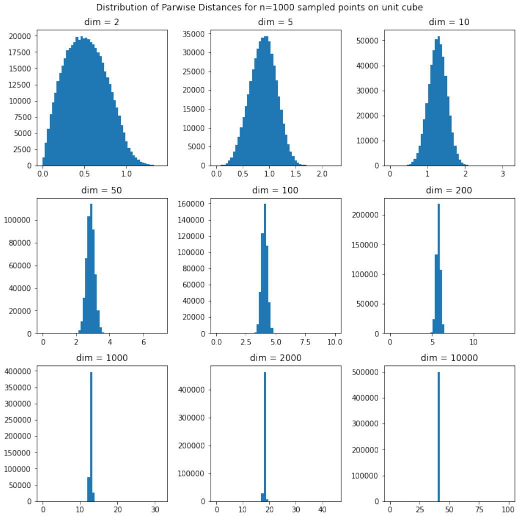
- Distances between points breaking down

Curse of Dimensionality

Certain phenomena occur when we increase the number of dimensions (i.e features) in our problem.

The most common are:

- Distances between points breaking down
- The need for even *more* data


Distribution of Parwise Distances for n=1000 sampled points on unit cube
dim = 2
dim = 5
dim = 10
dim = 50
dim = 100
dim = 200
dim = 1000
dim = 2000
dim = 10000

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

- *1 binary variable*

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

- *1 binary variable* - 2 unique combinations

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

- *1 binary variable* - 2 unique combinations - 20 samples

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

- *1 binary variable* - 2 unique combinations - 20 samples
- *2 binary variables* - 4 unique combinations - 40 samples

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

- *1 binary variable* - 2 unique combinations - 20 samples
- *2 binary variables* - 4 unique combinations - 40 samples
- *k binary variables* - 2^k unique combinations - 10×2^k samples

Say we have a classification task and we want 10 samples per unique combination of variables for a comprehensive data set.

- *1 binary variable* - 2 unique combinations - 20 samples
- *2 binary variables* - 4 unique combinations - 40 samples
- *k binary variables* - 2^k unique combinations - 10×2^k samples

So for 20 features, we need $10 \times 2^{20} = 10485760$ data points!

Section 4

Linear Smoothing

Linear Smoothing

k-NN regression typically fits a choppy model to our data. Linear smoothing tries to smooth out the fit by incorporating a *kernel* to weight the influence nearest neighbours by distance.

Linear Smoothing

k-NN regression typically fits a choppy model to our data. Linear smoothing tries to smooth out the fit by incorporating a *kernel* to weight the influence nearest neighbours by distance.

If we define h as the smoothing parameter and K as the kernel, the Linear Smoothing estimate is:

$$\hat{y}_i = \frac{\sum_{j=1}^{n} K\left(\frac{\|x_i - x_j\|}{h}\right) y_i}{\sum_{j=1}^{n} K\left(\frac{\|x_i - x_j\|}{h}\right)}$$

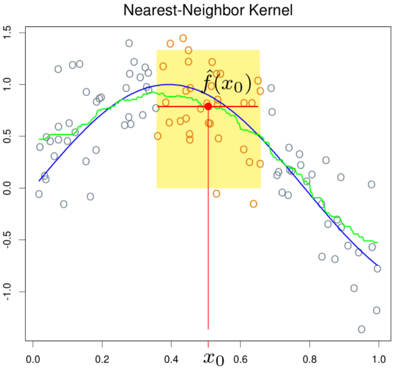
Linear Smoothing

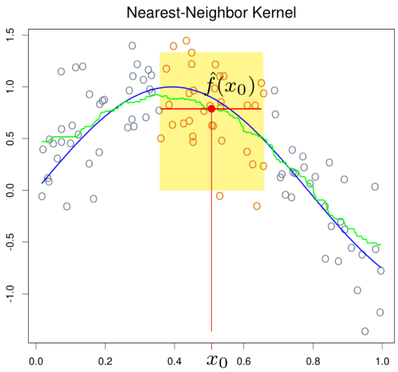
k-NN regression typically fits a choppy model to our data. Linear smoothing tries to smooth out the fit by incorporating a *kernel* to weight the influence nearest neighbours by distance.

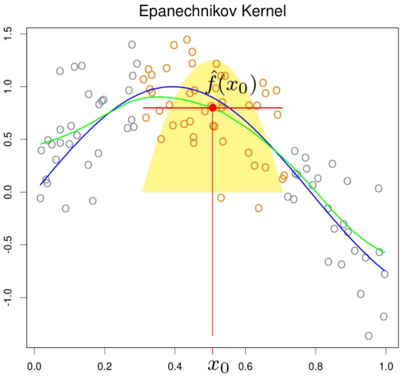
If we define h as the smoothing parameter and K as the kernel, the Linear Smoothing estimate is:

$$\hat{y}_i = \frac{\sum_{j=1}^{n} K\left(\frac{\|x_i - x_j\|}{h}\right) y_i}{\sum_{j=1}^{n} K\left(\frac{\|x_i - x_j\|}{h}\right)}$$

As $h \to 0$ our distances have a higher variance. If $h \to \infty$ have a lower variance, and our model is in turn smoother.


Nearest-Neighbor Kernel
$\hat{f}(x_0)$
x_0
0.0
0.2
0.4
0.6
0.8
1.0
-1.0
-0.5
0.0
0.5
1.0
1.5


Nearest-Neighbor Kernel
$\hat{f}(x_0)$
x_0
0.0
0.2
0.4
0.6
0.8
1.0
-1.0
-0.5
0.0
0.5
1.0
1.5


Epanechnikov Kernel
$\hat{f}(x_0)$
x_0
0.0
0.2
0.4
0.6
0.8
1.0
-1.0
-0.5
0.0
0.5
1.0
1.5