Decision Theory

Alípio Jorge

March 2021

Function approximation > Statistical Decision Theory

Recap

- We can see machine learning as **function approximation**.
- There is an underlying unknown function $f(\mathbf{X})$ and we want to discover a function $\hat{f}(\mathbf{X})$ that approximates it.
  - The discovery process is done by **learning** from examples, i.e., observed **data**.
  - This can be done in many **different ways**
    - classes of functions, quality criteria, learning algorithms
- **How do we define what is the best approximation to look for?**

Statistical Decision Theory

- To learn an approximated $f(\mathbf{X})$ (now denoted only by f for simplicity) we need to measure how good the approximation is.
  - A **loss function** penalizes bad predictions.

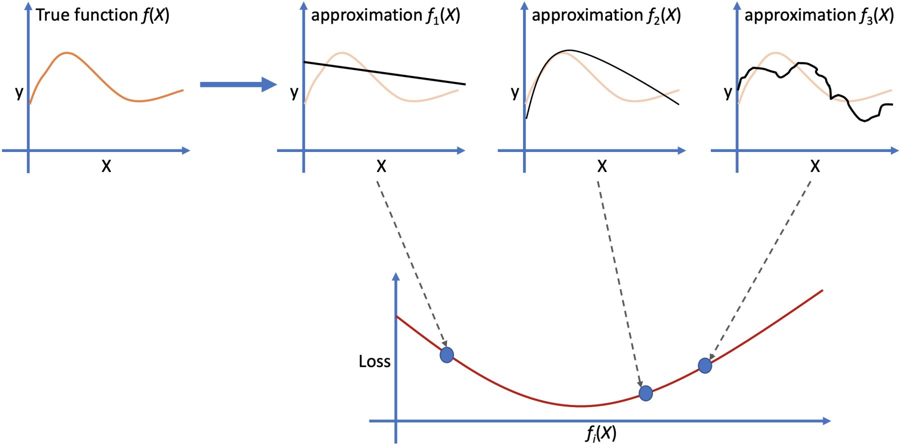
$$L(Y, f(X))$$

- Such as the **Squared Error Loss** (also called L_2)

$$L(Y, f(X)) = (Y - f(X))^2$$

- We want the $f(\mathbf{X})$ that minimizes **loss**

Statistical Decision Theory



Statistical Decision Theory

Examples of loss functions

- A credit decision:
  - true classes are $y =< good, bad, bad, good >$
  - predictions are $\hat{y} =< good, good, bad, good >$
  - **Loss** can be:
    - the number of errors: 1
    - the proportion of errors: 0.25
- How many days before discharge?
  - true values are $y =< 10, 8, 2, 6 >$
  - predictions are $\hat{y} =< 7, 6, 4, 6 >$
  - **Loss** can be:
    - RSS (Residual Sum of Squares): 17
    - RMSE (Root Mean Squared Error): 2.062

Statistical Decision Theory

- The loss function gives a **criterion** for choosing f
  - we want to minimize the **Expected Prediction Error**

$$EPE(f) = E(L(Y, f(X)))$$

- If Y and X are continuous, by the definition of **Expected value**

$$EPE(f) = \int L(y, f(x)) \Pr(dx, dy)$$

- In the case we use squared error loss

$$EPE(\hat{f}) = E(Y - f(X))^2 = \int [y - f(x)]^2 \Pr(dx, dy)$$

Statistical Decision Theory

- We know that $P(X, Y) = P(Y|X).P(X)$
- So:

$$EPE(\hat{f}) = \int [y - f(x)]^2 \Pr(dx) \Pr(dy|dx)$$

$$= \int_x \int_{y|x} [y - f(x)]^2 \Pr(dy|dx) \Pr(dx)$$

$$= E_X E_Y([Y - f(X)]^2|X)$$

- E_X ranges over on the **universe of possible cases**
  - we can abstract that away by focussing on each point x

Statistical Decision Theory

- How do we **minimize** *EPE*?
  - $f(x)$ is the value c that minimizes the squared error, given x

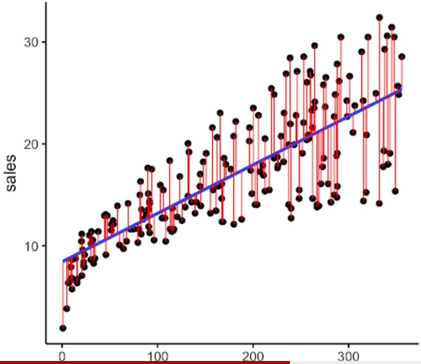
$$f(x) = \arg\min_c E([Y - c]^2 | X = x)$$

$$f(x) = E(Y | X = x)$$

- The **best prediction** of Y at any point $X = x$ is the **conditional mean**
  - when best is **measured by average squared error**

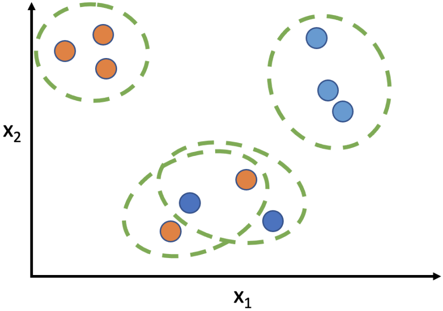
Statistical Decision Theory

- Does **linear regression** find the best prediction?
  - LR uses the **least squares** method (LS)
  - LS minimizes $([y - f(x)]^2)$ over X which minimizes EPE
  - As long as **we assume** that the best f is a linear function
- Given a data set, we find $f(x)$ by relying on the **training data**
  - We minimize the **average** loss over the training points



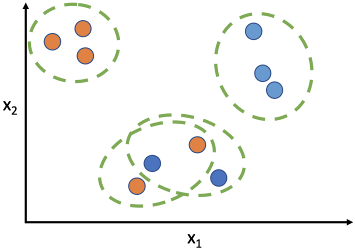
Statistical Decision Theory

- Does **Nearest Neighbors** find the best prediction in regression?
  - Given $X = x$, NN **averages** the $f(x')$ where $x' \in Neighbours(x)$
  - So, NN estimates $E(Y|X = x)$ by reasoning **locally**
- The success of **NN** depends on the robustness of this estimate
  - Training points need to be **sufficiently dense**



Statistical Decision Theory

- Does **Nearest Neighbors** find the best prediction in regression?
- It can be shown that if:
  - $N, k \to \infty$
  - $k/N \to 0$
  - Then $kNN(x) \to E(Y|X=x)$
- **Machine learning is solved**! Why do we need to look further?



Statistical Decision Theory

- Why **kNN** is not a **universal approximator**
    - Samples are **not so large**
    - Especially if the **number of dimensions** p **is high**
    - The estimate of $E(Y|X=x)$ gets harder as p increases
    - kNN converges to an optimal solution but the **rate of convergence** can be very slow
    - Think of the problem of estimating the **price of an apartment**
- Typically **kNN** approximations tend not to be stable

Statistical Decision Theory

In summary

- The Nearest Neighbour reasons **locally** and calculates the average of the neighborhood values of *y*
  - expectation is approximated by **averaging over sample data**
  - estimation at a point is relaxed to estimation on some **region close to the target point**.
  - This tends to work with a **sufficient** number of examples
- Linear regression assumes a **specific form of function**
  - it is **model-based**
  - this is a **global** function that works for any region of the input space
  - these assumptions lead to the **least squares formulation**

Machine Learning is a hard problem

- To reason **locally** we need a lot of data
  - We may not have it
  - Even a lot of data can be little data (e.g.: images of people doing things)
- If we reason **globally** we need to assume a model
  - The model may be too simplisitc
- More **complicated models**?
  - Always require some kind of **assumptions**
  - Increasing computational cost
  - **Sub-optimal** solutions
- **Hybrid** local/global search?
  - Finding **optimal** regions is **unfeasable**
  - We use **sub-optimal** approaches

Statistical Decision Theory

What happens if we replace the L_2 loss function with

$$L_1 = E(|Y - f(X)|)$$

- Then
  - $f(x) = median(Y \mid X = x)$
- Advantage of L_1
  - Estimates are more robust than the mean (e.g.: different sub-samples)
- This can be used but
  - L_2 is more convenient analytically (we can more easily prove **properties**)
    - In particular L_2 is more ameanable to derivation
  - L_2 is more popular

Statistical Decision Theory

- What if the output is a **categorical** variable G?
  - we need a different loss function
  - we can use a **cost matrix**
  - below is the **0-1 loss function**

c.a.	C_1	C_2
C_1	0	1
C_2	1	0

$$L_{0/1}(\hat{C}, C) = \nVdash_{\hat{C}=C}$$

- The $0/1$ loss for a set of examples is the sum of the losses
  - The proportion can also be used

Statistical Decision Theory

- What if the **loss** is different for different classes?
    - Other cost matrices can be used (e.g. diagnosis)

c.a.	*sick*	*healthy*
sick	0	5000
healthy	50	0

- Example:
    - $y =< sick, sick, healthy >$
    - $\hat{y} =< sick, healthy, sick >$

Statistical Decision Theory

- For a **generic loss function**, what is the Expected Prediction Error (EPE)?
  - K are the k classes and the **classifier function** to be learned is $f(x)$

$$EPE = E[L(G, f(X))]$$

- We factor the joint densities

$$EPE = E_X E_Y(L(G, f(X)) \mid X)$$

- Because G is categorical E_Y is calculated with a **sum**
  - ($\mathcal{G}$ is the set of classes)

$$EPE = E_X \sum_{k \in K} L(\mathcal{G}_k, f(X)) \ \Pr(\mathcal{G}_k | X = x)$$

Statistical Decision Theory

- So the **approximated** classification function is

$$f(x) = argmin_{g \in \mathcal{G}} \sum_{k=1}^{K} L(\mathcal{G}_k, g) \Pr(\mathcal{G}_k | X = x)$$

- In the case of the **zero-one loss** function

$$f(x) = max_{g \in \mathcal{G}} \Pr(g | X = x)$$

- Which is known as the **Bayes classifier**

Statistical Decision Theory

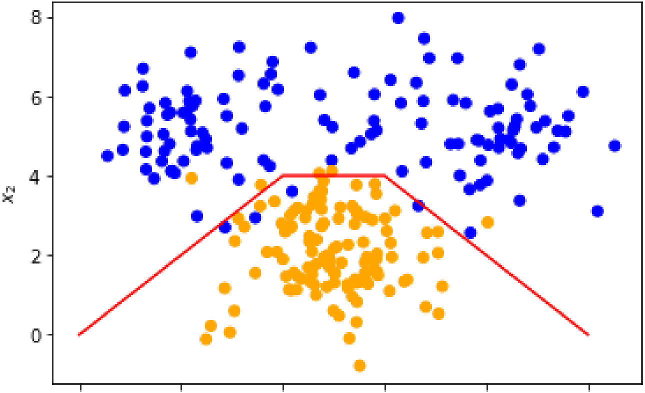
The Bayes Classifier

$$f(x) = max_{g \in \mathcal{G}} \Pr(g|X = x)$$

- **BC** says the best class is the most **probable** one, given the observation x
- The error rate of **BC** is the **Bayes rate**
- How can we obtain a **BC**?
  - kNN classifier **approximates** the **BC**.
    - the **majority vote** estimates the conditional probability
  - There are different ways of **estimating** $\Pr(g|X = x)$
    - Naive Bayes, Decistion Trees, Neural Networks

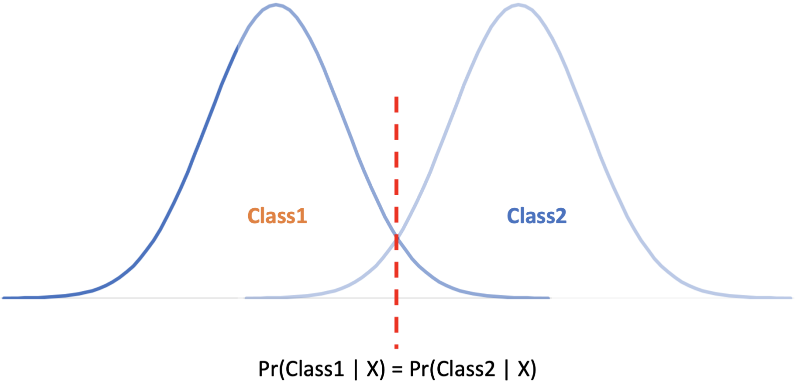
Bayes decision boundary

- The decision boundary defined by the optimal **BC** is the **Bayes Decision Boundary**
  - The **BDB** is optimal (from the Bayesian Decision Theory point of view)
  - It is usually not possible to determine, unless we know the densities behind



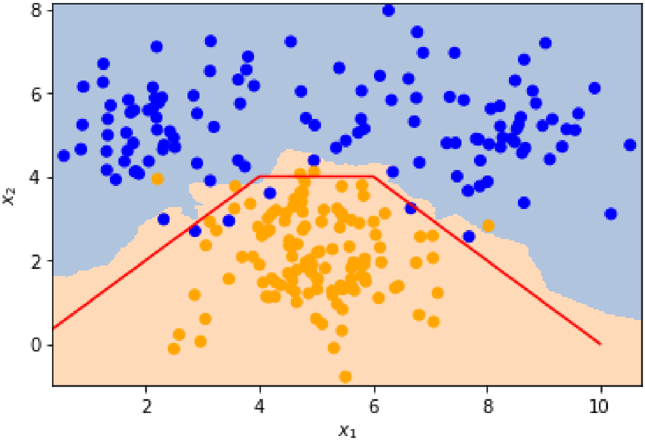
Bayes decision boundary

- We can determine the Bayes Decision Boundary if we know the densities behind
  - In the example, we have 4 bivariate normals with the same standard deviation



Bayes decision boundary

- kNN can approximate the Bayes Decision Boundary



Summary

- How to do ML?
  - ML can be done by **function approximation**
  - The quality of an approximation can be defined by a **loss function**
- How do we **minimize** loss?
  - We minimize **Expected Prediction Error** using Statistical Decision Theory
  - Minimizing loss amounts to **robustly estimating** $P(y|X=x)$

Summary

- How do we estimate $P(y|X = x)$?
  - Linear Regression assumes a **model shape** and uses least squares
    - Learning becomes **parameter estimation**
  - k Nearest Neighbor estimates **locally** by averaging y in a vicinity of x
    - In Classification it uses **majority voting**
  - Other ML methods will have **other approaches**
- Bayes Classifier assigns to x the most probable class, conditioned to $X = x$
- Bayes Decision Boundary is the boundary of the optimal Bayes Classifier

Bibliography

- Hastie, T., Tibshirani, R.,, Friedman, J. (2008). The Elements of Statistical Learning, Second Edition. New York, NY, USA: Springer New York Inc. (Chapter 2)