Look at numpy from assignments. ✓

KNN ✓

Decision Trees

Stats

Linear Algebra.

Vectorization ✓

Use chatgpt to generate practice questions.

Look at SLIDES and ASSIGNMENTS

Made with Goodnotes

KNN:

$\sqrt{1^2+1^2}=\sqrt{2}$

$\sqrt{2}$ 2 2

training data → Data used to train the model.
testing data → Data used to test the model.

Concept: Find k (usually odd number) nearest neighbors to test sample, pick majority class.

If k is even and split is equal, use weighted voting (closer = larger weight). Random?

Euclidean distance min dist $\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}$

Possible edge cases in data:

- Ties in class labels.
- Noisy data
- High-dimensionality → curse of dimensionality.
- Imbalance in number of samples for a given class.
- Identical points.

overfits, all points appear equally close, distance metrics lose effectiveness high cost etc.

Training set = Test final model performance. Model doesn't see this while being trained.

Validation set = Use during training to tune hyper parameters

Tune hyper parameters using a validation set

Use test set at the very end to see generalization performance of the final configuration.

Choosing K

Small k
- Good at capturing patterns. (local structure)
- May overfit (be sensitive to noisy data)

Large k
- Stable predictions as it averages over lots of data.
- May under fit → As it smoothes, it fails to capture local patterns and ignores relevant/closer neighbors. Fail to capture the underlying complexity.

Generally $k < \sqrt{N}$. This helps maintain a good balance between capturing local structures in the data while avoiding oversmoothing. Balance between bias and variance.

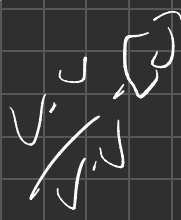
Evaluating Classifier:

- What's the accuracy?
- How badly did it get the wrong predictions.
- Which ones did it get wrong?

We can't use test sets for tuning meta-parameters as:
- Overfit (will be biased on test set it's tuned on)
- Biased Performance
- Test set should remain unseen.

Generalization Error: How well does the classifier do with data it's never seen?

Cross Validation: Choose best k by utilizing all test data rather than a single train-test split.



USE WHEN

Euclidean
+ Normally distributed/continuous
+ Emphasize larger differences (squares, weights larger differences)

Manhattan
- Contains outliers. that can skew results
- Working in high-dimensional space.

Made with Goodnotes

Stats:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

$$P(+) = P(+,H) + P(+,DH)$$
$$= P(H) \times P(+|H) + P(DH) \times P(+|DH)$$

$$P(a,b) = P(b|a) \times P(a) = P(a|b) \times P(b)$$

$$P(a,b,c) = P(a|b,c) \times P(b,c) = P(a|b,c) \times P(b|c) \times P(c)$$

A and B are independent if $P(A,B) = P(A) \times P(B)$

A and B are conditionally independent given C if $P(A,B|C) = P(A|C) \times P(B|C)$

BAYES:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad //$$

EXPECTED VALUE: Average value given probabilities.

$$E[X] = \sum P(X=a) \times a$$

$$E[X+Y] = E[X] + E[Y]$$

$$E[cX] = cE[X]$$

VARIANCE: Measure the spread of a set around their mean. Shows average of how spread the data is compared to its mean.

$$Var(X) = E[(X-\mu)^2]$$

$$Var(cX) = c^2 var(X)$$

Ex For a dice

$E[X] = 3.5$

$Var(X) = \frac{1}{6} \times (1-3.5)^2 + \frac{1}{6} \cdot (2-3.5)^2 \ldots\ldots$

If X and Y are independent:

$$E[XY] = E[X]E[Y]$$

$$Var(X+Y) = Var(X) + Var(Y)$$

Made with Goodnotes

Vectorization:

- Convert operations performed element by element into operations performed on entire arrays or vectors by taking advantage of modern CPU architectures that support parallel processing.

- Reduce overhead and processing time, increase performance.

```
array = np.array([1,2,3,4,5])

res = array + 5,,  # add 5 to each element.
```

np.sum(), np.mean()

Broadcasting = Perform operations on arrays of different shapes.

Made with Goodnotes

Info THEORY:

$E[\text{Surprise}] = -\sum P(x) \cdot \log(P(x))$

$H(x) = 1 \rightarrow$ Maximum uncertainty
$H(x) = 0 \rightarrow$ Maximum certainty.

PROPERTIES:

$H(x,y) = H(x|y) + H(y) = H(y|x) + H(x)$

If x and y are independent.

$H(x|y) = H(x)$

$H(x|x) = 0$

$H(x|y) \leq H(x)$

$IG(x|y) = H(x) - H(x|y)$

if y doesnt help x at all:

$IG(x|y) = 0$

If y is completely informative about x

$IG(x|y) = H(y)$

Ex R, C = 0.24
R, NC = 0.01
NR, C = 0.25
NR, NC = 0.5

$IG(C|R) = H(C) - H(C|R)$

$0.49 \times \log(0.49) - 0.51 \times \log(0.51)$
$-\left(\frac{1}{4} \times H(\text{cloudy} | \text{is rain}) + \frac{3}{4} \times H(\text{cloudy} | NR\right)$

$IG(y|x) = H(y) - H(y|x)$

Decision Tree:	KNN
Fast	Slow
Interpretable	Harder to interpret.
Can handle missing data	Can't handle missing data.
Robust to scaling	Not robust to scaling.
Lot of hyperparams	Few hyperparams.

Made with Goodnotes

LINEAR ALGEBRA:

$$U = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad V = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad B^T = \begin{bmatrix} 7 & 9 \\ 8 & 10 \end{bmatrix}$$

$$U^T A = \begin{bmatrix} 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1\times1+2\times3 & 1\times2+2\times4 \end{bmatrix} = \begin{bmatrix} 7, & 10 \end{bmatrix}$$

Made with Goodnotes