KNN-Imbalanced

Machine Learning - Lecture 6

Agenda :

1. Weighted KNN
2. Distances
3. Bias Variance Tradeoff (Read only)
4. Outliers
5. Interpretability
6. Practical Use Case of KNN
7. Case study : KNN
8. KNN Implementations (Sklr)

---

* Train time → $O(1)$ → very important (fit training)
test time → $O(n \cdot d \cdot \log k)$

Space Complexity → entire training data → $O(nd)$

log log linear/ log time complexity → $\frac{\log n}{n} \cdot \frac{d}{d}$
$\Rightarrow O(d)$

* Weighted KNN

$w_i = \frac{1}{d}$

$\frac{1}{0.1} = 10$

$w^{(1)} = \frac{1}{d^{(1)}}$

$w^{(1)} = 10$

$w^{(2)} = 2$

$(1,0,0)$ $d^{(1)} = 0.1$

$(0,1,0)$ $d^{(2)}, d^{(3)} = 0.5$

$w^{(2)}, w^{(3)} = 2$

Total weight (1,0,0,0,0) $\Rightarrow 10+10 \Rightarrow \boxed{20}$ $\Rightarrow$ 1-vs

total weight (1,1,1,0,0) $\Rightarrow 2+2+2 \Rightarrow \boxed{6}$ $\Rightarrow$ 1-vs

KNN → 1-vs

KNN → Learning Prototypes

* Prototypes (KNN)

→ Homogeneous neighborhood

* Distance.

Hyperparameter

euc. dist $(x_i, x_j) = \left[\sum_{j=1}^{d} (x_{ij} - x_{jj})^2\right]^{1/2}$

Point wise value of d large.
(curse of dimension) → Metric.

Manhattan Dist.

$d = 4$
↳ Manhattan dist.

Lorentzian → 2D

$d = \left[\sum_{j=1}^{d} |x_{ij} - x_{jj}|^1\right]^{1/1}$

Lorentzian.

$\sum_{j=1}^{d} \log(1 + |x_{ij} - x_{jj}|)$

Manhattan dist

$d(x_i, x_j) = \left[\sum_{j=1}^{d} |x_{ij} - x_{jj}|^q\right]^{1/q}$

generalization
can be between dist

if $p = 2$ → euclidean
if $p = 1$ → Manhattan.
(L1 dist)

$A(1,2)$ $B(3,5)$

$\frac{1}{1+4} = \frac{1}{5}$ $\frac{1}{1+9} = \frac{1}{10}$

$\frac{1}{4} > \frac{1}{9}$

* Bias Variance tradeoff.

Overfit
Underfit

$\boxed{k = 1}$ (Zero)

Fitting to noise/ outlier

overfit

$\boxed{k = n}$

overfit → high variance.

total datapoints → 100

$\frac{90}{100}$ → +ve

$\frac{10}{100}$ → -ve

$\boxed{K = 90}$

$\Rightarrow y_q$

$\Rightarrow$ +ve

Always predict (+ve)
↳ underfitting.

Underfit → high bias

Conclusion :

K ↓↓ → overfit
K ↑↑ → underfit

Train
Test

Cross Fold validation ?

Best.

overfit
underfit
Bias
Variance
Model complexity

* Multi-class classification ?
(Easy)

* Outliers

(Zero)
$\boxed{k = 1}$

Model complex
Due to outliers

Outliers
$\frac{1}{k} \to 0$
if great.

As we incr. (K↑) model gets data imposed due to outliers↑.

* Interpretability. (Sklr)

log reg → weights (w)

distance not interpretable.

Case : 1

y_q

Also DS : Interpretable

Using "neighborhood" for interpretation

* E. g. Categorical.

gender, Race, Marital Status.

$\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$

Hamming distance

Categorical → 0-1
Numerical → Euclidean.

$P(y=1 | \frac{M}{F})$
Count
$\frac{M}{F}$ → $\frac{y=1}{y=0}$
$\frac{M}{F}$ → $\frac{y=1}{y=0}$

Test → 0.82

Prob → $\frac{2}{3}$

Cosine similarity $(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$
$\frac{\text{Dot Prod}}{\text{Prod Norm}}$
Prop. Space.

$[-1, 1, 0, \frac{1}{2}]$

Used when d is large.

Conclusion
1. (Good) KNN Mean → Easy dim red
2. visualization → prototype/boundary
3. Cosine. Sim → when d↑ (Big Data)
4. (Bad) Cosine distance metric
5. Prone to Curse of dim.

* How we get Interpretability from KNN instead of Black ?

$\boxed{k = 3}$
$\frac{1}{3}$ $\frac{2}{3}$ $\frac{3}{3}$
Interpretable

Weighted KNN :

$\frac{1}{1.1}$ $\frac{1}{2.1}$ $\frac{1}{3.1}$

$w^{(1)}, w^{(2)}, w^{(3)}$

$P(y=1 | x_q) = \frac{\sum_{i=1}^{k} w^{(i)} \cdot y^{(i)}}{\sum_{i=1}^{k} w^{(i)}}$

$P(y=1 | x_q) = \frac{0.91}{0.91}$

$P(y=1 | x_q) = \frac{0.91}{0.91}$

→ Now gradient of KNN.

Bonus [Kernel Density Estimation]

Can KNN use Kernel ?
→ Yes Kernel

* Ball Tree/ Kd Tree

$\boxed{k = 3}$

Kd Tree

Ball Tree

Preprocessing
Sklr

Tree → Ball Tree
Kd Tree

"KNN"

* Weighted Voting / Gaussian Kernel

↳ Weighting Function

$\boxed{k = 3}$

→ Weighted
→ Gaussian Kernel
→ Weighted Voting.
→ Interpretable

$w_q(x^{(1)}, x^{(2)}, x^{(3)}) \; w^{(1)}, w^{(2)}, w^{(3)}$

* Outliers

$\frac{1}{3}$ → $\frac{1}{3}$

y_q

$\frac{1}{k}$ → 0 if great.

Weighted → 0.5

$\frac{1}{1.1}$ $\frac{1}{2.1}$ $\frac{1}{3.1}$

$w^{(1)}, w^{(2)}, w^{(3)}$

$w^{(1)} \to 0$

$w^{(2)} \to 0$

$w^{(3)} \to 0$

$\frac{1}{k}$ → 0 if great.

$\frac{1}{1.1}$ $\frac{1}{2.1}$ $\frac{1}{3.1}$

$w^{(1)}, w^{(2)}, w^{(3)}$

$w^{(1)} \to 0$

$w^{(2)} \to 0$

$w^{(3)} \to 0$

* Implementation

Brute

$\frac{1}{3}$ → $\frac{1}{3}$

Ball Tree

$\frac{1}{3}$ → $\frac{1}{3}$

Kd Tree

$\frac{1}{3}$ → $\frac{1}{3}$

Preprocessing
Sklr

Tree → Ball Tree
Kd Tree

$\boxed{k = 3}$

* Sample Tempo Search.

Hard ...

$\boxed{k = 3}$

$\frac{1}{3}$ → $\frac{1}{3}$

$\frac{1}{3}$ → $\frac{1}{3}$

$\frac{1}{3}$ → $\frac{1}{