Performance measures & kNN

Introduction

k-nearest neighbors (kNN)

- Simple(st) non-parametric approach to estimate $f(x)$
- Assumes that $f(x)$ can be approximated by a **locally constant** function
- Memory-based method without training phase

Distances and prediction

kNN setup: Compute **distances** between training and test observations (in $\mathbf{x}$)

Euclidean Distance: $||\boldsymbol{x}_a - \boldsymbol{x}_b||_2$

$$\sqrt{\sum_{j=1}^{P}(x_{aj} - x_{bj})^2}$$

Minkowski Distance:

$$\left(\sum_{j=1}^{P}|x_{aj} - x_{bj}|^q\right)^{\frac{1}{q}}$$

Manhattan Distance: $||\boldsymbol{x}_a - \boldsymbol{x}_b||_1$

Other metrics available: Mahalanobis, Chebyshev...

Distances and prediction

Prediction based on K cases in the training data closest to test example x_0 ($\mathcal{N}_0$)

kNN Regression

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i$$

$\rightarrow$ Estimated y equals average outcome of training cases in $\mathcal{N}_0$

kNN Classifier

$$P(Y = j | X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

$\rightarrow$ $P(Y = j)$ equals fraction of $I(y_i = j)$ in neighborhood $\mathcal{N}_0$

Distances and prediction

Prediction based on K cases in the training data closest to test example x_0 ($\mathcal{N}_0$)

kNN Regression

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i$$

$\rightarrow$ Estimated y equals average outcome of training cases in $\mathcal{N}_0$

kNN Classifier

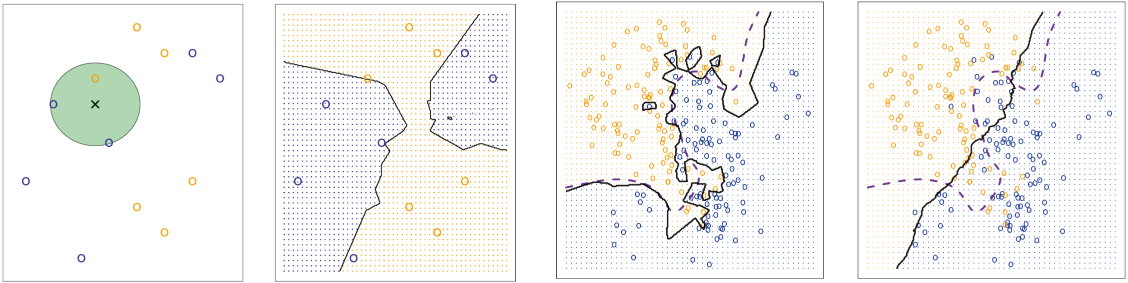
$$P(Y = j | X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

$\rightarrow$ $P(Y = j)$ equals fraction of $I(y_i = j)$ in neighborhood $\mathcal{N}_0$

Distances and prediction

Figure: kNN classifier

(a) Example, k = 3 (b) Decision boundary, k = 3 (c) k = 1 (d) k = 100



James et al. 2013

Tuning K

The number of neighbors K

- K is a **tuning parameter** and chosen using e.g. CV
- Small K provides flexible fit but high variance
- Large K leads to less variable fit which may cause bias

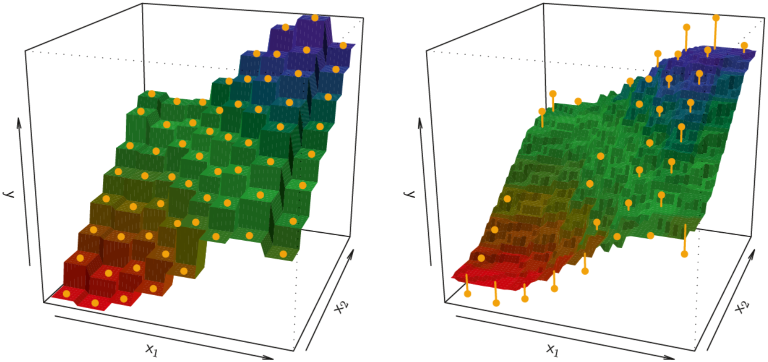
$\rightarrow$ Bias-Variance Trade-Off with kNN

$$\begin{aligned}\mathrm{Err}(x_0) &= \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) + \mathrm{Var}(\varepsilon) \\ &= \left[f(x_0) - \frac{1}{k}\sum_{\ell=1}^{k} f(x_{(\ell)})\right]^2 + \frac{\sigma_\varepsilon^2}{k} + \sigma_\varepsilon^2\end{aligned}$$

Tuning K

Figure: kNN regression

(a) K = 1 (b) K = 9



James et al. 2013

Limitations

Drawbacks of kNN

- Distances dependent on scaling of predictor variables
  - Standardizing of features
- "Curse of dimensionality"
  - No nearby neighbor in sparse data with many features
- Sensitive to irrelevant features
- Computational costs increase with sample size

Extensions

- Weighted kNN
  - Weight neighbors by distances

References

Hossin, M., Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2): 1–11.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York, NY: Springer.

Sokolova, M., Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4): 427–437.