COMPSCI 762 Tutorial Week 10

Supplementary material for kNN and SVM

Luke Chang

May 2022

The University of Auckland

K-Nearest Neighbour (kNN) Model

K-Nearest Neighbour (kNN) Model

The k-nearest neighbour fits for $\hat{Y}$ is defined as follows:

$$\hat{Y}(\mathrm{x}) = \frac{1}{k} \sum_{\mathrm{x} \in N_k(\mathrm{x})} y_i$$

where $N_k(\mathrm{x})$ is the neighbourhood of x defeined by the k closest points x in the training sample.

What does kNN do during training?

K-Nearest Neighbour (kNN) Model

The k-nearest neighbour fits for $\hat{Y}$ is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x \in N_k(x)} y_i$$

where $N_k(x)$ is the neighbourhood of x defeined by the k closest points x in the training sample.

What does kNN do during training?

- Saving all training instances
- Algorithms used to compute the nearest neighbors:
  - Brute-force search
  - **KD Tree:** Splits from *median* on every feature; works well in lower dimensional data
  - **Ball Tree:** Also a binary tree which partitions data from N-dimensional hyper-sphere; the preferred method for high dimensional data

K-Nearest Neighbour (kNN) Model

What should you use for the distance metric?

K-Nearest Neighbour (kNN) Model

What should you use for the distance metric?

- Euclidean Distance: L_2-norm
- Manhattan Distance: L_1-norm, works better in higher dimensional data
- Mahalanobis Distance, Chebyshev Distance (L_∞-norm) and others

How do you choose the value of k?

K-Nearest Neighbour (kNN) Model

What should you use for the distance metric?

- Euclidean Distance: L_2-norm
- Manhattan Distance: L_1-norm, works better in higher dimensional data
- Mahalanobis Distance, Chebyshev Distance (L_∞-norm) and others

How do you choose the value of k?

- Apply cross-validation on the training data.
- Don't forget fit the model with the full training data after the optimal k is selected.

What are the limitations?

K-Nearest Neighbour (kNN) Model

What should you use for the distance metric?

- Euclidean Distance: L_2-norm
- Manhattan Distance: L_1-norm, works better in higher dimensional data
- Mahalanobis Distance, Chebyshev Distance (L_∞-norm) and others

How do you choose the value of k?

- Apply cross-validation on the training data.
- Don't forget fit the model with the full training data after the optimal k is selected.

What are the limitations?

- Sensitive to noise
- Computational expensive at inference time (Scale by the size of training data)
- Does not scale well with larger datasets

Support Vector Machine (SVM)

Support Vector Machine (SVM)

Explain briefly how an SVM is trained.

[1] Hastie, Tibshirani, and Friedman, *The elements of statistical learning: data mining, inference, and prediction*.

Support Vector Machine (SVM)

Explain briefly how an SVM is trained.

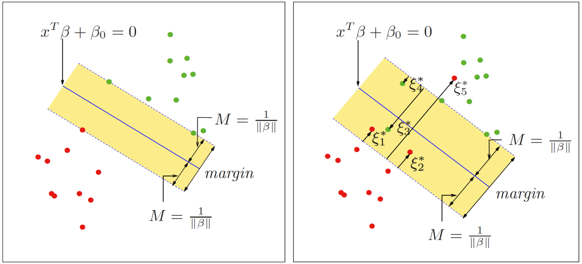


Figure 1: SVM. Left: A separable case; Right: A non-separable case. The vectors ξ_j^* are the support vectors[1].

[1] Hastie, Tibshirani, and Friedman, *The elements of statistical learning: data mining, inference, and prediction.*

Support Vector Machine (SVM)

Explain briefly how an SVM is trained.

- A technique for constructing an optimal separating hyperplane between two classes
- The margin M is $\frac{1}{||\beta||}$; Minimize $||\beta||$ (Maximize margin)
- **Hard-margin:** the training data is linearly separable
- **Soft-margin:** the data are not linearly separable, minimize the observations on the wrong side by minimizing the hinge loss using Lagrangian multiplier.
- **Kernel function** is used for the non-linear boundaries. e.g.: 2-degree polynomial $\phi(x) = x^2$

Multi-class Classification

What strategies are SVM use when the data have more than two classes?

Multi-class Classification

What strategies are SVM use when the data have more than two classes?

- **One-Vs-Rest (OVR):** Example: Three output classes: A, B, C. Solve 3 binary classified problem:
  1. A vs. (B, C)
  2. B vs. (A, C)
  3. C vs. (A, B)
- **One-Vs-One (OVO):** Train N choose 2 classifiers, $\binom{N}{2} = \frac{N \cdot (N-1)}{2}$
  1. A vs. B
  2. A vs. C
  3. B vs. C

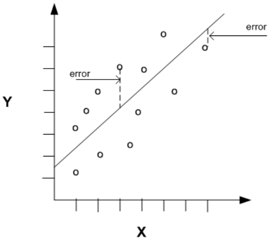
Support Vector Machine

What the difference between SVM and logistic regression?

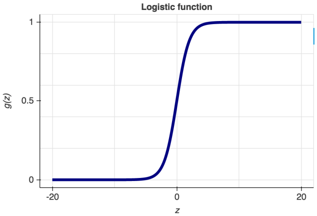
Support Vector Machine

What the difference between SVM and logistic regression?

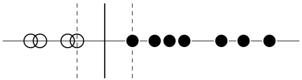
- SVM maximizes the margin between the closest support vectors
- Logistic regression maximize the posterior class probability (Different loss function)
- SVM is deterministic and LR is probabilistic
- SVM can be used for both classification and regression



(a) Linear Regression



(b) Logistic Regression



(c) SVM

How do SVMs compare to simple instance-based learning approaches such as k-Nearest Neighbour?

Support Vector Machine

How do SVMs compare to simple instance-based learning approaches such as k-Nearest Neighbour?

- Both can be thought of as instance based learners
- SVM doesn't need to store all training samples
- SVM outperforms kNN in high dimensional data

Parameters in SVM

Which hyper-parameters should you tune?

Parameters in SVM

Which hyper-parameters should you tune?

- SVM is NOT scale invariant. Before training, normalize your data.
- **Complexity parameter:** The penalty parameter c of the error term. In *sklearn*, the default value is 1. If the data is noisy, decrease it (Less penalty for misclassification.). If the data is highly non-linear, increase it. c can take value larger than 1, e.g.: $c \in [0.1, 100]$
- **kernel**: Linear, polynomial, sigmoid, Radial Basis Function (RBF)
- For non-linear kernels, γ is the kernel coefficient. The default value is $\frac{1}{\text{n\_features}}$. If γ is small, the model prefers linear-like decision boundary. Large value may lead to overfitting.