Statistische Geheimhaltung - Cell Key Methode

Joshua Simon

Otto-Friedrich-University Bamberg

joshua-guenter.simon@stud.uni-bamberg.de

May 24, 2022

Overview

1. Einführung
   - Veröffentlichungen in der amtlichen Statistik
   - Warum ist Geheimhaltung notwendig?
2. Etablierte Geheimhaltungsverfahren
   - Posttabulare Verfahren
   - Pretabulare Verfahren
3. Cell Key Methode
   - Methodik
   - Beispiel Implementierung
   - Anwendung in der Hochschulstatistik
4. Fazit

Linearly separable data classes

First, let's consider a given data set $\mathcal{X}$ of labeled points (inputs) with individual labels $y_i \in \{-1, 1\}$, e.g. $(x_1, y_1), ..., (x_m, y_m) \in \mathcal{X} \times \{-1, 1\}$.

Our goal is to implement a classification method, which is able to classify new and unlabeld data points with the right or 'best' label.

Linearly separable data classes

In machine learning, a well established classification method are the so called **Support Vector Machines** (SVM). Developed by Vladimir Vapnik and his coworkers in the 1990s, SVMs are still a relevent topic and an even more powerful tool for **classification** and **regression**.

Hyperplane classifiers

The underlying learning algorithm of SVMs yields to find a hyperplane in some dot product space $\mathcal{H}$, which separates the data. A hyperplane of the form

$$\langle w, x \rangle + b = 0 \tag{1}$$

where $w \in \mathcal{H}, b \in \mathbb{R}$ shall be considered [Schölkopf, 2002] (p. 11). Futhermore decision functions

$$f(x) = sgn\left(\langle w, x \rangle + b\right) \tag{2}$$

can be assigned.

Hyperplane classifiers - A constrained optimization problem

The **optimal hyperplane** can be calculated by finding the normal vector w that leads to the largest margin. Thus we need to solve the optimization problem

$$\begin{aligned} \min_{w \in \mathcal{H}, b \in \mathbb{R}} \quad & \tau(w) = \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i\left(\langle w, x\rangle + b\right) \geq 1 \ \forall i = 1, \ldots, m. \end{aligned} \tag{3}$$

The constraints in (3) ensure that $f(x_i)$ will be $+1$ for $y_i = +1$ and -1 for $y_i = -1$. The ≥ 1 on the right hand side of the constraints effectively fixes the scaling of w. This leads to the maximum margin hyperplane. A detailed explanation can be found in [Schölkopf, 2002](Chap 7).

The kernel trick

To extend the introduced SVM algorithm, we can substitute (**??**) by applying a kernel of the form

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \tag{4}$$

where

$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{H} \\ (x) &\mapsto \Phi(x) \end{aligned} \tag{5}$$

is a function that maps an input from $\mathcal{X}$ into a dot product space $\mathcal{H}$. This is referred to as the **kernel trick**.

A suitable kernel

Going back to our problem of non linearly separable data, we can use a kernel function of the form

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right), \tag{6}$$

a so called **Gaussian radial basis function** (GRBF or RBF kernels) with $\sigma > 0$.

More kernel applications

Some interessting kernel applications:

- Image recognition/classification (with SVMs) for example in
  - Handwriting recognition
  - Tumor detection
- Computer vision and computer graphics, 3D reconstruction
- Kernel principal component analysis

References

Schölkopf, Bernhard, Alexander J. Smola
Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT press, 2002.

Liesen, Jörg, Volker Mehrmann
Lineare Algebra. Wiesbaden, Germany: Springer, 2015.

Jarre, Florian, Josef Stoer
Optimierung: Einführung in mathematische Theorie und Methoden. Springer-Verlag, 2019.

Reinhardt, Rüdiger, Armin Hoffmann, Tobias Gerlach
Nichtlineare Optimierung: Theorie, Numerik und Experimente. Springer-Verlag, 2012.

Bronstein, Ilja N., et al.
Taschenbuch der Mathematik. 11. Auflage, Springer-Verlag, 2020.

Chang, Chih-Chung, Chih-Jen Lin
LIBSVM : A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

Time for your questions!

Follow our development on GitHub

https://github.com/JoshuaSimon/Cell-Key-Method