ML 2 Exam SS18, Session 1

xxx

September 3, 2018

Exercise 1: Multiple choice

(1) What does CCA do?

- Project two datapoints on the direction with highest covariance
- Project one datapoint on the direction with highest covariance
- ...

(2) What is the weakest for the kernel formulation such that one class SVMs are equivalent (circle/hyperplane)?

- $k(x_i, x_i) = 0$
- $k(x_i, x_i) = const$
- $k(x_i, x_j) = 0$
- $k(x_i, x_j) = const$

(3) Which equation is equal to the propability $P(S_i = q_5 | S_k = q_3)$

- ...

(4) Bioinformatic kernel ($k(x, x') = \sum_{m=1}^{d} \beta_m \sum_{n=1}^{N} I_{u_{m,n(x)}=u_{m,n(x')}}$) with d=1

- The kernel is more prone to overfitting than higher degree kernels
- The kernel has more hyperparameters than higher degree kernels(d¿1)
- ...

Exercise 2: Practical ML

(a) Gezeigt wird ein Github Repository. Man soll sich einen Kernel überlegen, um verschiedene Repos anhand von unstructured text und Zahlen(Anzahl commits usw.) zu vergleichen.

(b) Mit der gegebenen Gram Matrix soll nun das Ergebnis visualisiert werden. Welche Methode eignet sich und wie würde man vorgehen.

Exercise 3: Markov Chains

Gegeben ist die Übergangsmatrix A einer Markovkette mit 3 "States", die von 3 Variablen abhängen, e.g. $A_{11} = (1 - a)$

(a) Für die durch A gegebene Markovkette bedingte Wahrscheinlichkeiten ausrechnen

(b) Eine Folge von 10 "States" ist gegeben. Nun sollen mit der Maximum Likelihood Methode die optimalen Werte für die 3 Variablen ausgrechnet werden.

(c) Jetzt Hidden Markov: Die Ereignisse sollen zyklisch sein, also alle 9 Schritte wieder auf einen bestimmten State kommen. Wie sollte so ein Modell aussehen.

Exercise 4: One Class SVM

Gegeben ist die Lagrange-Funktion vom One Class SVM, und die Duale Version. Man soll beweisen, dass das Dual genau so aussieht.(Siehe VL7 Seite 17)

Exercise 5: Programming

Gegeben ist die Dokumentation von numpy.convolve und SVM.scipy:

(a) Man soll ausgehend von einem Datenvektor x die Gram-Matrix des convolution kernel $k(x, x') = ||x * x'||$ programmieren.

(b) Nun soll man prufen ob die Matrix psd ist (mit code)

(c) Eine SVM mit diesem Kernel implementieren