

Arbeitsbericht

Projekt: LDA und Svm

Autor : Benedikt Lauer

Ein großer Vorteil bei unserem Projekt war, dass unser Team schon vorher ein Freundeskreis war. Wir kamen alle sehr gut miteinander aus und hatten eine sehr gute Teamarbeit. Nachdem die Themen verteilt waren, haben wir uns noch im Dezember zusammengesetzt, um unser weiteres Vorgehen zu planen. Wir hatten vor uns alle ein wenig in unser Thema einzulesen, und uns im neuen Jahr noch in der vorlesungsfreien Zeit zu treffen, um unser weiteres Vorgehen zu besprechen. Bei diesem treffen haben wir dann die einzelnen Themen verlost.

Ich habe SVM gezogen. Das erste Problem lag darin, sich in die Thematik einzulesen. Ich habe mehrere Wochen gebraucht, um genau zu verstehen, was ich machen muss. Leider wurde in der Quelle teilweise auch Vorwissen aus den Vorherigen Kapiteln vorausgesetzt.

Zum Berechnen der Variable alpha war das Lösen eines quadratischen Optimierungsproblems nötig. Dafür durften wir ein Paket einbinden. Nach einigem Suchen nach einem geeigneten Paket habe ich mich für das Paket Nlcoptim entschieden, welches mein Problem anscheinend gut löst, wobei manchmal Fehlermeldungen ausgegeben werden, die wahrscheinlich mit dem Startpunkt der Optimierung zusammenhängen, wobei es womöglich noch andere Gründe geben kann. Um möglichst keine Fehlermeldung auszugeben, wird die Fehlermeldung einige Male abgefangen und der Startwert verändert. Wird weiterhin ein Fehler ausgegeben und wenn ein Kernel verwendet wurde, wird noch einmal versucht, eine Klassifikationsfunktion ohne Kernel zu berechnen und wenn das auch nicht geklappt hat eine Fehlermeldung ausgegeben.

Die SVM – Klassifikationsfunktion, die in der Quelle beschrieben wird, funktioniert leider nur für zwei Klassen. Deswegen habe ich die Klassifikationsfunktion so erweitert, dass sie auch auf mehrere Klassen anwendbar ist. Dafür gibt meine Decision-Funktion `svm_classify_list()` eine Liste von Funktionen zurück, die jeweils zwei Klassen des Datensatzes miteinander vergleichen. Meine Klassifikationsfunktion, `svm_classify()` die als Eingabeparameter eine Observation nimmt und die zugehörige Klasse zurückgibt vergleicht jeweils zwei Klassen und ordnet eine Observation der Klasse zu, die im direkten Vergleich am besten ist. Das war fast der Hauptaufwand der eigentlichen Implementierung des Algorithmus, der eventuell auch als Bonus-Task angesehen werden kann.

Anschließend wurden von mir noch die Kernels für den SVM-Algorithmus implementiert und die Tests für unser Paket geschrieben.

Es war eine sehr gute Erfahrung mit motivierten Teammitgliedern zu arbeiten, die auch gut programmieren können, sodass ich viel von ihnen lernen konnte und sie mir häufig bei Problemen helfen konnten. Außerdem haben die Teammitglieder, die einen leichteren Algorithmus implementieren mussten und dadurch schneller waren viele zusätzliche Funktionen geschrieben und ein R6 Objekt erstellt, die ich auch nutzen konnte, was eine große Hilfe war. Wir haben gemeinsam an der Präsentation gearbeitet und alle die gleiche Präsentation gehalten. Die Arbeit mit Git und Bitbucket war teilweise kompliziert, da es häufiger Merge-Konflikte gab und teilweise alte Dateien neu gepusht wurden. Eine genauere Einführung in die Nutzung von Git in der Vorlesung wäre möglicherweise hilfreich gewesen.

Als Zusatzaufgaben haben wir eine Shiny-App und eine Educational Vignettes erstellt. In der Shiny-App kann ein vorhandener Datensatz eingelesen, oder ein neuer erstellt und anschließend der Plot einer der von uns implementierten Klassifikationsfunktionen angezeigt werden. Außerdem wird der Fehler, den unser Algorithmus auf dem Datensatz macht berechnet und in einer Grafik angezeigt.

Die beiden Grafiken kann man bei bedarf auch in seinen Dateien abspeichern.