Research Seminar 1: Task 4 – Embedding und PCA

Markus Deutschl

Finden eines Embedding-Algorithmus

Ausgehend von der mir vorgeschlagenen Publikation (Slivkins 2006)[1] habe ich mir die Systematik von Embedding und weitere Arbeiten in diesem Themengebiet angesehen. Die algorithmischen Überlegungen sind sehr interessant, aber auch hochkomplex. Dies macht sich vor allem bei hohen Dimensionszahlen bemerkbar.

Viele der Algorithmen bauen auf Berechnungen mit Eigenwerten und Eigenvektoren auf, die mit einer Dimensionsanzahl größer drei nur schwierig zu berechnen sind. Des Weiteren besitzt auch nicht jede Matrix diese Werte und Vektoren.

Eine weitere Schwierigkeit ergab sich für mich bei der Implementierbarkeit der gefundenen Algorithmen. Diese wurden nämlich fast ausschließlich durch Fließtext und mathematische Formeln erklärt, was für mich als Nicht-Mathematiker schwer bis gar nicht verständlich war.

Schlussendlich bin ich dann auf die Dissertation von Blake Shaw (Shaw 2011)[2] gestoßen, worin der Algorithmus „Minimum Volume Embedding (MVE)“ vorgestellt wird. Dieser wird auch anhand von Pseudo-Code aufgeführt. Laut dem Autor reduziert MVE die Dimensionen in einer dünn besetzten Matrix effizienter und besser als kPCA (Kernel Principal Component Analysis). Damit fiel meine Wahl auf diesen Algorithmus, da dieser neben dem Embedding auch Dimensionsreduktion durchführt. Allerdings ergaben sich dadurch neue Probleme (siehe Mail vom 12.05.).

Fortschritt

Ich habe bereits begonnen, den MVE-Algorithmus mit Java zu implementieren. Dazu verwende ich ein Java Matrix Package namens JAMA[3], welches bereits einige Matrixoperationen zur Verfügung stellt, sowie den JDBC-Konnektor von Oracle[4], für die Datenbankabfragen, um die Distanzen zwischen den einzelnen Filmen zu ermitteln.

Der Algorithmus ist zum gegenwärtigen Zeitpunkt wegen den mathematischen Unklarheiten nur als Skelett vorhanden. Der Code dazu kann auf meinem persönlichen GitHub-Repository angesehen und heruntergeladen werden.

[1] http://research.microsoft.com/en-us/um/people/slivkins/thesis.pdf
[2] http://academiccommons.columbia.edu/catalog/ac:141634
[3] http://math.nist.gov/javanumerics/jama/
[4] http://dev.mysql.com/downloads/connector/j/