

Variational Bayesian Hidden Markov Models

Hidden Markov Modelle (HMM) finden sich in vielen Anwendungen im Bereich vom maschinellen Lernen wieder. Der Standard Algorithmus für die Schätzung des Maximum-Likelihoods (ML) der Modellparameter ist der sogenannte Baum-Welch-Algorithmus. Eine Instanz vom durchwegs bekannten Expectation-Maximisation-Algorithmus (EM). Dieser ML-Ansatz vernachlässigt die Modellkomplexität und resultiert in einer gewöhnlichen Punktschätzung. Dadurch neigt der Ansatz zur Überanpassung der Modellparameter, was grundsätzlich unerwünscht ist. Geeigneter ist ein Bayesianischer Ansatz (VB); die damit verbundene Notwendigkeit über den gesamten Parameterraum zu marginalisieren, kann mittels Variations Algorithmen angenähert werden. Diese Arbeit zeigt die Herleitung und c++ Implementierung solcher Algorithmen.

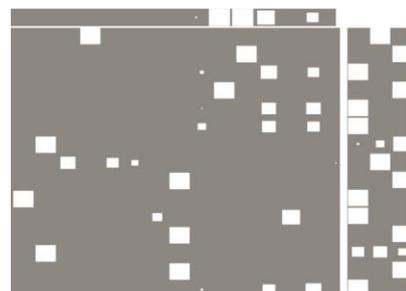
In einem ersten Experiment wurde das Problem der Überanpassung untersucht. Dafür wurden zwei HMMs mit synthetischen Daten trainiert. Für das eine HMM wurde der VB-Ansatz verwendet und für das andere der ML-Ansatz. Das Experiment war erfolgreich und zwei Hinton-Grafiken (siehe Abbildungen rechts) zeigen die Unterschiede auf. Beim VB-Ansatz sind die Parameter ohne Information gleichmässig uniform verteilt, während der ML-Ansatz die Parameter überanpasst.

In einem zweiten Experiment wurde der Einfluss der Prior-Stärke auf das Optimierungsverfahren untersucht. Dafür wurde ein HMM mit den Proteinsequenzen des Escherichia Coli Bakteriums trainiert. In einem Re-training, mit einer mutierten Proteinsequenz und verschiedenen Prior-Stärken, konnte deren Einfluss untersucht werden. Es hat sich gezeigt, dass mit einem starken Prior die Lernkurve steiler wird. Es kam die Vermutung auf, dass ein zu starker Prior das HMM zu stark stört und alle Modellparameter gleichmässig verteilt werden. Doch in diesem Zusammenhang sind weitere Untersuchungen notwendig.

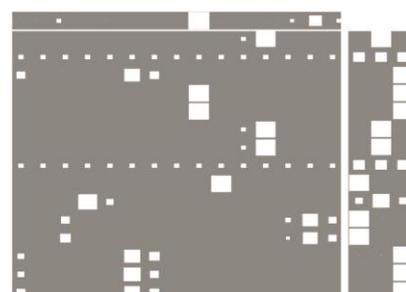


Diplomand
Manuel Alexander Hirzel

Dozent
Thomas Oskar Weinmann



ML-optimierte Modellparameter. Die weissen Quadrate repräsentieren das Gewicht eines Matrix-Eintrags. Je grösser das Quadrat, desto näher liegt der Wert bei 1.



VB-optimierte Modellparameter. Die Zeilen ohne Information sind uniform verteilt.