Studienarbeitsexposé

Repeatable Benchmarking Mahout

Entwicklung eines Lasttest-Rahmenwerkes für Apache Mahout

von:	Oliver Fischer
	Institut für Informatik
	Humbold-Universität zu Berlin
Matrikelnummer:	19 41 42
Fachgebiet:	Informatik
Lehrstuhl:	Wissensmanagement in der Bioinformatik
Betreuer:	Prof. Ulf Leser, Humboldt-Universität zu Berlin
	Isabel Drost, Neofonie GmbH

1 Motivation

Der Termin Maschinelles Lernen bezeichnde tden Teilbereich der Künstlichen Intelligenz, der sich mit der Bildung von Lernmodellen und deren computergestützten Umsetzung beschäftigt. In der praktischen Anwendung steht das Ziel im Vordergrund Programme zu entwickeln, die aufgrund eines gegeben Modells und bereits vorhandenes Wissens über die Modelldomäne zu neuen Schlußfolgerungen zu gelangen.

Die damit verbundenen Anwendungsfälle sind vielfältig und nicht nur von wissenschaftlichem Interesse. Anwendungsgebiete wie Fraud-Prevention und Detection im Bereich der Kreditwirtschaft, Börsenanalysen oder der Erkennung von Spam sind nur eine Beispiele.

Auch im Bereich der internetbasierten Dienste nimmt die Bedeutung des Maschinellen Lernens zu. Anfang der 1990ziger Jahre beruhte das gerade entstandene World-Wide-Web auf einer statischen Informationsbereitstellung, bei dem die einzelnen Informationen durch eine Hyperlink-Struktur miteinandern in Beziehung gesetzt wurden und setzte ein aktives Benutzerverhalten voraus, bei dem Benutzer selber entschieden welche Informationen sie abrufen wollen. Mit dem Aufkommen der ersten Suchmaschinen in den Folgejahren wurden die Internet-Benutzer in die Lage versetzt ihnen bis dahin unbekannte Information durch die Angabe von für das gesuchte Thema relevanten Stichworten aufzufinden. Die Suche über Schlüsselwörter ist auch heute noch der verbreitetste Weg zur Auffindung von Informationen.

Durch die Zunahme an Informationsangeboten und einer durch das kontinuierliche Erschließen neuer Anwendungsgebiete und damit Benutzergruppen, stößt die klassische Methode des aktiven Suchens durch den Benutzer an seine Grenzen. Neue Angebote im Internet setzen zunehmend auf eine Mischung aus klasssicher Suche über Schlüsslworte und Bereitstellung von Informationen, die aufgrund der bekannten Präferenzen des Benutzers als für ihn relevant ermittelt wurden. Diese Services beruhen auf einer praktischen Anwendung von Algorithmen aus dem Bereich des Maschinellen Lernens. Bekannte Beispiele dürfte der Produktempfehlungsservice von Amazon oder die Musikplattform last.fm sein. Die Verfügbarkeit von kostengünstigen und effizienten Lösungen für Maschinelles Lernen kann daher als Key-Success-Factor für die Entwicklung neuer und neuartiger Internet-gestützter Dienste angesehen werden.

Apache Mahout ist unter dem Dach der Apache Software Foundation entwickelte Bibliothek von Implementierungen von Algorithmen aus dem Bereich des Maschinellen Lernens. Bei der Entwicklung von Mahout wurde besonders Wert auf einen effizienten

Umgang mit großen Datenmengen und eine entsprechende Skalierbarkeit der Bibliothek an sich gelegt. Aufgrund der Lizenzierung unter der Apache 2.0-Lizenz, die eine weitgehende liberale Nutzung in jedem Umfeld erlaubt, kann Mahout zu einem wichtigen Infrastrukturbestandteil neuer Internet-Dienste entwickeln.

2 Zielsetzung

Der von Apache Mahout formulierte Anspruch eine effiziente und skalierbare Plattform für Maschinelles Lernen zu sein, jedoch existieren derzeit keine Benchmarks, die diesen Anspruch objektiv überprüfbar machen.

Im Rahmen der Studienarbeit soll daher eine Framework entwickelt werden, mit dessen Hilfe die Güte der von Mahout bereitgestellten Algorithmen aus den Bereichen Recommendations, Clustering und Classification überprüft werden kann. Das zu entwickelnde Framework soll zwei Aufgaben erfüllen. Zum einen soll eine kontinuierliche Beobachtung des Performanceverhaltens von Apache Mahout über den gesamten Zeitraum der aktiven Entwicklung ermöglichen. Durch diese Funktionalität werden die Mahout-Entwickler in die Lage versetzt negative als auch positive Veränderugen im Leistungsverhalten festellen zu können und so sowohl Architektur- als auch Implementierungsentscheidungen auf fundierter Basis hinsichtlich ihrer Auswirkungen auf das Leistungsverhalten der bereitgestellten Algorithmen bewerten zu können. Zum anderen soll das Framework Anwendern von Mahout ermöglichen eigene Benchmarks entsprechend ihren eigenen Anwendungsszenarien zu entwickeln und auszuführen. Anwender erhalten dadurch, analog zu den Mahout-Entwicklern, die Möglichkeit qualifizierte Systemarchitekturentscheidungen für den Einsatz von Mahout zu treffen.

Zur Erreichung der genannten Ziele muß das Framework über eine Plug-In-Architektur verfügen, die sowohl eine leichter Erweiterbarkeit der Testszenarien als auch weiterer Funktionen erlaubt. Für ein effizientes Performance-Monitoring wird das Framework eine kurfristige Persistierung lokaler Meßreihen als auch eine Langzeitarchivierung der Meßergebnisse erlauben. Für letzteres wird eine Export-Schnittstelle bereitgestellt, die den Aufbau einer Referenzsammlung ermöglicht.

Zur Auswertung der gesammelten Meßungen wird eine Reporting-Möglichkeit in das Framework integiert, die vorhandene Meßergebnisse visualisiert und Trends hervorhebt.

3 Vorgehensweise

Im Rahmen der Studienarbeit müssend die im folgenden aufgeführten Teilaufgaben umgesetzt werden. Diese können teilweise unabhängig von einander umgesetzt werden, teilweise bauen sie aufeinander auf. Daher stellt die Reihenfolge der Teilaufgaben nur bedingt deren Umsetzungsabfolge dar.

Folgende Teilaufgaben sind umzusetzen:

- Frameworkentwicklung
- Identifizierung von Testdatenbeständen
- Implementierung einer Referenztestsuite
- Aufbau einer Sammlung von Meßergebnisdatenbank
- Beurteilung des Performanceverhaltens

Die einzelnen Teilaufgaben werden nachstehend weiter ausgeführt.

3.1 Frameworkentwicklung

Die Entwicklung des Frameworks stellt den die erste Teilaufgabe dar und bildet die Grundlage für alle weiteren Aufgaben. Hierfür muß es über die bereits eingangs erwähnte Plug-In-Architektur verfügen, um die geforderte Erweiter- und Anpaßbarkeit sicherzustellen. Darüber hinaus muß es sowohl die lokale als auch die verteilte Ausführung von Tests erlauben, da Mahout auch Algorithmen enthält, die auf Apache Hadoop aufbauend in Clustern verteilt ausgeführt werden können. Da nicht vorausgesetzt werden kann, daß jeder Mahout-Entwickler über einen eigenen Rechner-Cluster verfügt, mit dem eine hinreichende Anzahl von Konfigurationen getestet werden kann, wird das Framework über eine Unterstützung der Amazon Elastic Compute Cloud (EC2) verfügen[1].

Eine mögliche Erweiterung des Testframeworks besteht in der Integration eines Mini-Benchmarks, mit dem die wesentlichen Performance-Werte, der für die Ausführung von Mahout verwendeten JVM, bestimmt werden. Die Aufnahme dieser Werte in die Performancemessung würde eine bessere Beurteilung der Leistungsfähigkeit von Mahout erlauben. Für die Entscheidung, ob eine solche Erweiterung möglich ist, müßten eine

[1] Für akademische Zwecke und die Entwicklung von Apache Mahout werden durch Amazon Ressourcen kostenlos bereitgestellt.

Übersicht über existierende Java-Benchmarks und deren lizenztechnische Verfügbarkeit erstellt werden.

Für einen möglichst hohe Akzeptanz, weiter Verbreitung und häufige Nutzung des Testsframeworks ist ein hoher Grad an Automatisierung bei der Testausführung notwendig.

3.2 Identifizierung von Testdatenbeständen

Eine Voraussetzung für eine gute Vergleichbarkeit der durch die Referenztestsuite zu gewinnenden Testergebnisse stellt ein stabiler Bestand an Testdaten dar. Nur, wenn die Testergebnisse über einen längeren Zeitraum auf dem gleichen Datenbestand aufbauen, ist eine gute Vergleichbarkeit der Ergebnisse zu gewährleisten. Daher muß für die Referenztestsuite ein mit dem Framework gebündelter Bestand an Testdaten bereitgestellt werden. Hierfür ist eine Übersicht über existierende und verfügbare Datenbestände zu erstellen, die für möglichst viele Algorithmen verwendet werden können. Können nicht ausreichend viele freie Datenbestände gefunden werden, muß mit Hilfe der Mahout-Entwickler- und -anwendergemeinschaft versucht werden, fehlende Datenbestände zu erstellen oder bereitzustellen.

3.3 Implementierung einer Referenztestsuite

Aufbauend auf den verfügbaren Datenbeständen soll für eine möglichst große Anzahl von durch Mahout bereitgestellte Algorithmen ein Referenztest erstellt werden und die Referenztestsuite für Apache Mahout bilden.

3.4 Aufbau einer Meßergebnisdatenbank

Im Abschluß der Studienarbeit wird eine Datenbank mit den Meßergebnissen der Referenztestsuite auf einer möglichst großen Anzahl von unterschiedlichen Rechnerkonfigurationen aufgebaut. Diese so gewonnen Meßergebnisse sollen jedoch nur die Grundlage der Datenbank bilden und durch die Meßergebnisse anderere Mahout-Entwickler und -Anwender ergänzt werden. Die initialen Messungen werden wahrscheinlich mit Hilfe der Amazon Elastic Compute Cloud gewonnen werden.

3.5 Beurteilung des Performanceverhaltens

Mit dem Aufbau der Meßergebnisdatenbank bietet sich auch eine Beurteilung des Performanceverhaltens von Apache Mahout an. In die Beurteilung werden drei Faktoren einfließen: Die horizontale Skalierbarkeit, das Verhalten bei unterschiedlichen Eingabedatenmengen und die Güte der Ergebnisse.

4 Grundlagenliteratur

TBW