Prof. Dr. L. Paditz, 11.02.2019

HTW Dresden,

paditz@htw-dresden.de

Einführung in die wahlobligatorische LV:

"Verarbeitung polystrukturierter Datenmengen"

Woher kommen **große Datenmengen** und wie sind diese **strukturiert**?

Wie kann man diese **BigData verarbeiten** und welche Erkenntnisse kann man daraus gewinnen?

Welches **SoftwareProdukt** kann mit BigData gut umgehen? **TensorFlow!**

1. Selbsttest-Frage als eActivty mit ClassPad

==========

Datenexplosion

Betrachten Sie das folgende Beispiel für

Formel-1-Rennwagen: Jeder Rennwagen hat 512 Sensoren, jeder Sensor zeichnet 32 Ereignisse pro Sekunde auf, wobei jedes Ereignis 64 Byte groß ist. Wie viele Daten werden von einem F1-Team prodziert, wenn ein Team zwei Autos im Rennen hat und das Rennen 2 Sunden dauert?

Benutzen Sie für die Berechnung bitte die folgenden Einheiten:

1000Byte=1kB, 1000kB=1MB. 1000MB=1GB.

Lösung:

Gesamtzeit=2h=

2·60·60s

7200·s

Anzahl der Ereignisse pro Fahrzeug:

7200s·512Sensor·32Ereignis/s/Sensor

117964800·Ereignis

Gesamte Anzahl Ereignisse pro Team:

2·ans

235929600·Ereignis

Gesamte Datenmenge pro Team:

64Byte/Ereignis·ans

$15099494400 \cdot Byte$

$ans * 10^{-9} GB/Byte$

$$\frac{1179648 \cdot GB}{78125}$$

approx(ans)

$15.0994944 \cdot GB$

Innerhalb der zwei Stunden wurden ca. 15.1GB Daten erzeugt.

2. Selbsttest-Frage als eActivty mit ClassPad

==========

zunächst Vorbetrachtungen:

Wörterbuchkodierung (dictionary encoding)
wörterbuch-kodierte Spalten:
Vornamensspalte (z.B. **49Byte/Eintrag**)

Beispiel für Wörterbuch-Kodierung:
durch kurzen **Integer-Wert 23-Bit,**
um z.B. 5Millionen verschiedene Vornamen (weltweit)
zu kodieren:

$\log_2(5 \cdot 10^6) < 23$

$$\frac{7 \cdot \ln(5) + 6 \cdot \ln(2)}{\ln(2)} < 23$$

approx(ans)

$22.25349666 < 23$

Column "fname" **Dictionary for "fname"**

recID	fname	valueID	Value
...	...	...	...

39	John	23	John
40	Mary	24	Mary
41	Jane	25	Jane
42	John	26	Peter
43	Peter	...	...
...	...		

Attribute Vector for "fname"

position	valueID
...	...
39	23
40	24
41	25
42	23
43	26
...	...

Weltbevölkerungstabelle:

8Mrd. Zeilen und 200Byte/Zeile

Attribut	**Anzahl** (einmalige Werte)	**Größe**
Vorname	5 Millionen	49Byte
Nachname	8 Millionen	50Byte
Geschlecht	2	1Byte
Land	200	49Byte
Stadt	1 Millionen	49Byte
Geburtstag	40000	2Byte
Summe	-	**200Byte**

3·49Byte+50Byte+1Byte+2Byte

200·Byte

Die komplette Datenmenge beträgt 1.6TB:

$8 \cdot 10^9$Zeile·200Byte/Zeile

1600000000000·Byte

ans/$(10^{12}$Byte$)$·TB

$\frac{8 \cdot TB}{5}$

approx(ans)

1.6·TB

Für jede Spalte:

Wörterbuch-Vektor und **Attribut-Vektor**

Jedes **Wörterbuch** speichert alle einmaligen **Werte (Values)** mit ihrer jeweilige Position, die **WertIDs (=ValueIDs).**
Die Position braucht hierbei nicht explizit gespeichert zu werden, da die Reihefolge der Einträge im Wörterbuch diese implizit bestimmt.

In einer **wörterbuch-kodierten Spalte** speichern die **Attribut-Vektoren** jetzt lediglich die **WertIDs,** welche den **WertIDs** im **Wörterbuch** entsprechen.
Die **DatensatzID** (Zeilennummer) wird implizit über die Position eines Eintrages im **Attribut-Vektor** gespeichert.

Zusammenfassung:

Über die **Wörterbuch-Kodierung** können alle

Informationen als Integer-Werte anstelle von anderen, in der Regel größeren, Datentypen gespeichert werden.

1. **Wörterbuch-Kodierung** der **Vornamen**

Wie viele Bits sind nötig, um alle 5Millionen einmalige Werte der Vornamen-Spalte "fname" darzustellen?

$\log_2(5 \cdot 10^6)$

$$\frac{7 \cdot \ln(5) + 6 \cdot \ln(2)}{\ln(2)}$$

approx(ans)

22.25349666

Es sind **23Bits** ausreichend, um alle einmaligen Werte für die gewünschte Spalte darzustellen, **anstatt 365,1GB** für die Vornamen-Spalte zu verwenden:

$8 \cdot 10^9 \text{Zeile} \cdot 49\text{Byte/Zeile}$

$392000000000 \cdot \text{Byte}$

$\text{ans}/(1024^3 \text{Byte}) \cdot \text{GB}$

$$\frac{95703125 \cdot \text{GB}}{262144}$$

approx(ans)

$365.0784492 \cdot \text{GB}$

Der **Attribut-Vektor** kann auf die Größe von **21,4GB** reduziert werden:

$8 \cdot 10^9 \text{Zeile} \cdot 23\text{Bit/Zeile}$

$184000000000 \cdot \text{Bit}$

$\text{ans}/(8\text{Bit}) \cdot \text{Byte}$

$23000000000 \cdot \text{Byte}$

ans/(1024^{3}Byte)·GB

$$\frac{44921875 \cdot GB}{2097152}$$

approx(ans)

$$21.42041922 \cdot GB$$

Außerdem wird ein zusätzliches **Wörterbuch** eingeführt, welches **0,23GB** an Speicher benötigt:

$5 \cdot 10^6$Wert·49Byte/Wert

$$245000000 \cdot Byte$$

ans/(1024^{3}Byte)·GB

$$\frac{3828125 \cdot GB}{16777216}$$

approx(ans)

$$0.2281740308 \cdot GB$$

Erreichter **Kompessionsfaktor≈17:**

$$\frac{\text{unkomprimierte Größe}}{\text{komprimierte Größe}} =$$

$$\frac{365.0784492 \cdot GB}{21.42041922 \cdot GB + 0.2281740308 \cdot GB}$$

$$\frac{4218691149617484235950}{250161928070388539003}$$

approx(ans)

$$16.86384168$$

Somit wird die Spaltengröße um den Faktor 17 reduziert, d.h. es werden nur ca. **6%** des anfänglichen Hauptspeicherplatzes benötigt:

$$\frac{21.42041922 \cdot GB + 0.2281740308 \cdot GB}{365.0784492 \cdot GB}$$

$$\frac{250161928070388539003}{4218691149617484235950}$$

approx(ans)

0.05929846941

Umrechnung in %:

approx(ans)·100

5.929846941

2. Wörterbuch-Kodierung des Geschlechts

Hier gibt es nur zwei einmalige **Werte** ("m" und "f"). Für die Darstellung des Geschlechts ist für jeden Wert nur 1Byte (ohne Kompression) notwendig. Damit beträgt die Menge der Daten (ohne Kompression) **7,45GB:**

$8 \cdot 10^9$Zeile·1Byte/Zeile

8000000000·Byte

ans/(1024^3Byte)·GB

$$\frac{1953125 \cdot GB}{262144}$$

approx(ans)

7.450580597·GB

Wenn Kompression verwendet wird, reicht 1Bit aus, um dieselbe Information darzustellen. Der **Attribut-Vektor** benötigt **0,93GB:**

$8 \cdot 10^9$Zeile·1Bit/Zeile

8000000000·Bit

ans/(8Bit)·Byte

1000000000·Byte

ans/(1024^3Byte)·GB

$$\frac{1953125 \cdot GB}{2097152}$$

approx(ans)

0.9313225746·GB

Das **Wörterbuch** benötigt zusätzlich **2Byte:**

2·1Byte

2·Byte

ans/(1024^3Byte)·GB

$$\frac{\text{GB}}{536870912}$$

approx(ans)

1.862645149E-9·GB

Erreichter **Kompessionsfaktor≈8:**

$$\frac{\text{unkomprimierte Größe}}{\text{komprimierte Größe}} =$$

$$\frac{7.450580597\cdot\text{GB}}{0.9313225746\cdot\text{GB}+1.862645149\text{E-}9\cdot\text{GB}}$$

$$\frac{139985764952118838176875000000}{17498220653541545733331679 31}$$

approx(ans)

7.999999984

Die **Kompessionsrate** ist sowohl von der **Größe des ursprünglichen Datentyps** als auch von der **Entropie der Spalte** abhängig, die von zwei Kardinalitäten bestimmt wird:

- **Spalten-Kardinalität** (die Anzahl der einmaligen Werte in einer Spalte)
- **Tabellen-Kardinalität** (Gesamtanzahl der Zeilen in der Tabelle oder der Spalte)

Es gilt:

$$\textbf{Entropie}(\text{der Spalte}) = \frac{\text{Spalten-Kardinalität}}{\text{Tabellen-Kardinalität}}$$

Je kleiner die **Entropie der Spalte** ist, desto höher

ist die **Kompessionsrate**, die erreicht werden kann. Die **Entropie** ist ein Maß, das abbildet, wie hoch die **Informationsdichte innerhalb einer Spalte** ist.

Erster und gleichzeitig wichtigster Effekt der Wörterbuch-Kodierung: alle Operationen in einer kodierten Datentabelle werden mithilfe der **Attribut-Vektoren** ausgeführt, die ausschließlich aus ganzzahligen **Integer-Werten** bestehen. Dies führt zu einer (impliziten) **Beschleunigung** (Speedup) aller Operationen, da die CPUs dafür entwickelt wurden: **Operationen werden mit Zahlen und nicht mit Zeichen durchgeführt.**

Wir kommen jetzt zur **2. Selbsttest-Frage:**

==========

Gegeben ist eine Bevölkerungstabelle (50Millionen Zeilen) mit den folgenden Spalten:

- **Vorname** (49Bytes, 20000 einmalige Werte)
- **Nachname** (49Bytes, 100000 einmalige Werte)
- **Alter** (1Byte, 128 einmalige Werte)
- **Geschlecht** (1Byte, 2 einmalige Werte)

Wie groß ist der

Kompressionsfaktor= $\frac{\text{unkomprimierte Größe}}{\text{komprimierte Größe}}$ bei

Anwendung der **Wörterbuch-Kodierung**?

Lösung:

zuerst Berechnung **ohne** Wörterbuch-Kodierung:

Gesamt-Größe pro Zeile:

(49+49+1+1)Byte/Zeile

$$100 \cdot \frac{\text{Byte}}{\text{Zeile}}$$

Gesamt-Größe (mit allen Zeilen):

ans·50·10^6Zeile

5000000000·Byte

ans/(10^6Byte)·MB

5000·MB

nun Berechnung **mit** Wörterbuch-Kodierung:

Anzahl der **Bits,** die für die **Attribute** benötigt werden.

- **Vorname**

$\log_2(20000) < 15$

$$\frac{4 \cdot \ln(5) + 5 \cdot \ln(2)}{\ln(2)} < 15$$

approx(ans)

14.28771238<15

- **Nachname**

$\log_2(100000) < 17$

$$\frac{5 \cdot \ln(5) + 5 \cdot \ln(2)}{\ln(2)} < 17$$

approx(ans)

16.60964047<17

- **Alter**

$\log_2(128) = 7$

7=7

- **Geschlecht**

$\log_2(2) = 1$

$$1=1$$

Gesamt-Größe der **Attribut-Vektoren:**

$$(15+17+7+1)\text{Bit/Zeile}\cdot 50\cdot 10^{6}\text{Zeile}$$

$$2000000000\cdot\text{Bit}$$

$$\text{ans}/8\cdot\frac{\text{Byte}}{\text{Bit}}$$

$$250000000\cdot\text{Byte}$$

$$\text{ans}/(10^{6})\cdot\frac{\text{MB}}{\text{Byte}}$$

$$250\cdot\text{MB}$$

Größe der **Wörterbücher:**

- **Vorname**

$$20000\cdot 49\text{Byte}$$

$$980000\cdot\text{Byte}$$

$$\text{approx}(\text{ans}/1000)\frac{\text{KB}}{\text{Byte}}$$

$$980\cdot\text{KB}$$

- **Nachname**

$$100000\cdot 49\text{Byte}$$

$$4900000\cdot\text{Byte}$$

$$\text{approx}(\text{ans}/10^{6})\frac{\text{MB}}{\text{Byte}}$$

$$4.9\cdot\text{MB}$$

- **Alter**

$$128\cdot 7\text{Byte}$$

$$896\cdot\text{Byte}$$

- **Geschlecht**

$$2\cdot 1\text{Byte}$$

$$2\cdot\text{Byte}$$

Größe des gesamten Wörterbuches **6MB:**

$980\mathrm{KB}/10^3\cdot\frac{\mathrm{MB}}{\mathrm{KB}}+4.9\mathrm{MB}+(896\mathrm{Byte}+2\mathrm{Byte})/10^6\cdot\frac{\mathrm{MB}}{\mathrm{Byte}}$

$$\frac{2940449\cdot\mathrm{MB}}{500000}$$

approx(ans)

$$5.880898\cdot\mathrm{MB}$$

Gesamtgröße:

Größe der Attribut-Vektoren+Größe der Wörterbücher=

250MB+6MB

$$256\cdot\mathrm{MB}$$

Kompressionsrate≈20:

$$\frac{5000\mathrm{MB}}{256\mathrm{MB}}$$

$$\frac{625}{32}$$

approx(ans)

$$19.53125$$

Quelle:

H. Plattner: **Lehrbuch In-Memory Data Management**

(2013)

ISBN 978-3-658-03212-8

(in Bibl. der BA Dresden vorhanden)

Download für dieses Dokument:

http://www.informatik.htw-dresden.de/

~paditz/Tensorflow-Ue01.pdf