technische universität
dortmund

fakultät für
informatik

Support Vector Machine
und
Quantum Computing

Mikail Yayla
TU Dortmund

Überblick

1. Einführung in die Quantenwelt
2. Vom Bit zum Quantenregister
3. QC Algorithmen + Mathematik
4. von QC zu SVM
5. Schlüsselfeature des QC für SVM

Prinzipien der Quantenphysik

- Welle-Teilchen-Dualismus
- Superposition
- Verschränkung

Quantum Computing – Das Bit

$$1 \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv |1> \qquad 0 \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv |0>$$

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{C}^2 \qquad \boxed{\alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha\,|0> + \beta\,|1>}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Rechnen mit Quantenbits

$$\vec{A} \cdot \vec{x} = \vec{y}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H|0> = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} (|0> + |1>)$$

Ein kleiner Algorithmus

1. $|x> \leftarrow |0>$
2. $|x> \leftarrow H|x>$
3. $\mathit{Messen}|x>$

n-Bit Quantenregister

$$R = \sum_{i=0}^{2^n - 1} \alpha_i |i> \qquad \sum_{i=0}^{2^n - 1} |\alpha_i|^2 = 1$$

$$R \in \mathbb{C}^{2^n}$$

- n Bits : 2^n mögliche Zustände
- Superposition aller Zustände
- mit n Bits 2^n Informationen speichern

Tensorprodukt

Was geschiet genau, wenn ich mehr als ein Bit habe ?

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} \in \mathbb{C}^{2^n} \quad \text{oder} \quad \mathbb{C}^4$$

$$\begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{C}^2 \qquad \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{C}^2$$

von QC zu ML

„QM is about linear manipulation of vectors in large vector spaces,

ML is about linear manipulation of vectors in large vector spaces.“

- Seth Lloyd, 29. Januar 2014 Google LA TechTalk

SVM

- Ziel: (Hyper-)Ebene finden
- Ebene soll Punkte unterteilen
- Ebene so wählen, dass um sie herum der Abstand zu Punkten maximal
- zwei am nächsten liegende Punkte sind entscheidend

SVM

$$\vec{w} \cdot \vec{u} \geqslant c$$

$$\vec{w} \cdot \vec{u} + b \geqslant 0$$

- falls wahr, dann +
- finde $\vec{w}$ und b
- $\vec{w}$ Orthogonal zu Ebene
- b konstant

SVM - Probleme

- reicht noch nicht
- welches b nutzen?
- es gibt mehrere $\vec{w}$
- Gegend um Ebene wird nicht berücksichtigt

SVM - Verbesserung

$$\vec{w} \cdot \vec{x}_{+} + b \geqslant \quad 1$$
$$\vec{w} \cdot \vec{x}_{-} + b \leqslant -1$$

- genauere Seperation
- Berücksichtigen der „Straße“
- zum Vergleich

$$\vec{w} \cdot \vec{u} + b \geqslant 0$$

SVM – weitere Verbesserung

- Einführung von Variable y_i
- y_i ist +1 für $\vec{x}_+$
- y_i ist -1 für $\vec{x}_-$
- multipliziere beide Seiten der vorherigen Gleichungen mit y_i

$$y_i\left(\vec{w}\cdot\vec{x}_+ + b\right) \geqslant 1$$
$$y_i\left(\vec{w}\cdot\vec{x}_- + b\right) \geqslant 1$$

SVM – weitere Verbesserung

$$y_i\left(\vec{w}\cdot\vec{x}_i + b\right) - 1 \geqslant 0$$

Gleichung wird 0 wenn support vectors eingesetzt werden

$$y_i\left(\vec{w}\cdot\vec{x}_i + b\right) - 1 = 0$$

$\vec{w}$ bestimmen

$$\text{Breite} = (\vec{x}_+ - \vec{x}_-)\frac{\vec{w}}{|\vec{w}|}$$

mit

$$y_i(\vec{w}\cdot\vec{x}_i + b) - 1 = 0$$

wird zu

$$\text{Breite} = \frac{2}{|\vec{w}|}$$

$\vec{w}$ bestimmen

- maximiere $\frac{2}{|\vec{w}|}$

  oder
- Minimiere $\frac{1}{2}|\vec{w}|^2$

- Lagrange-Formel findet Lösungen

Lagrange Formel

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j$$

- maximiere L
- Lösung hängt nur vom Skalarprodukt von Trainingsdaten ab
- Entscheidungsregel auch

Höhere Dimensionen/Kernel Trick

- Keine Ebene gefunden?
- Versuche in höheren (sehr hohen) Dimensionen
- bei Rücktrafo wird sie nichtlinear
- „large vectors in large vector spaces“, Skalarprodukte
- Passt zu QC

Was QC bietet

- n bits, 2^n Dimension
- Arbeiten mit Vektoren im 2^n auf n Bits
- exponentieller speed-up!
- Skalarprodukt/Matrixmult. auf n Bits bei Vektorgröße 2^n

Fazit

- QC ist noch am Anfang
- mit dem speed-up könnten heutige Grenzen überwunden werden
- für ML/SVM Algorithmen:

  $O(\log NM)$ für Training+Classification

Quellen

- Adams, Allen (2013): Vorlesung Quantum Physics I, MIT
- Homeister, Matthias (2013): Quantum Computing verstehen, 3. Aufl., Wiesbaden
- Lloyd, Seth/Mohseni, Masoud/Rebentrost, Patrick (2013): Quantum algorithms for supervised and unsupervised machinle learning, MIT, Google Research
- Lloyd, Seth/Mohseni, Masoud/Rebentrost, Patrick (2014): Quantum support vector machine for big data classification, MIT, Google Research Venice
- Winston, Patrick (2010): Vorlesung (Herbst), MIT Artificial Intelligence