Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

NEAREST-NEIGHBOUR-KLASSIFIKATION

Fachseminar "Machine Learning"

Letztes Update: 3. Februar 2016

Martin Beller

Studienbereich Informatik
Hochschule RheinMain

GLIEDERUNG

1. Einleitung

2. NN-Klassifikation

3. Beschleunigung

4. Diskussion

EINLEITUNG

WORUM GEHT ES IN "MEINEM" PAPER?

Es geht um...

- Objekt- und Szenenerkennung auf Bildern
- unter Verwendung von Nearest-Neighbour-Klassifikation
- auf einem großen Datensatz aus kleinen Bildern

Daten:

- ca. 80 Millionen Bilder aus dem Web
- zu jeweils einem von ca. 75.000 taxonomisch angeordneten Nomen

WINZIGE BILDER

Was ist hier zu sehen?

WINZIGE BILDER - ETWAS GRÖSSER

Was ist hier zu sehen?

WAS WERDEN WIR UNS ANSEHEN?

Wir schauen uns an...

- Was NN-Klassifikation überhaupt ist und was dazu gehört
- Wie sich die Performance verhält und wie man sie verbessern kann
- Fazit

NN-KLASSIFIKATION

DIE GRUNDIDEE

Klassifikation durch Vergleich mit Trainingsdaten: Ein neuer Fall wird so eingeordnet wie der ähnlichste bereits bekannte Fall.

Dazu braucht man:

- Ein Maß für die Ähnlichkeit
- Trainingsdaten

Lernen durch Speichern der Daten ("lazy learning" oder "Speicher-basiertes Lernen")

DISTANZMASSE

Ähnlichkeit: Umso höher, je geringer Distanz im Merkmalsraum
Häufig verwendet: Euklidische Distanz

Euklidische Distanz mit Gewichtung der Dimensionen

$$d_w(x, y) = |x - y| = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2} \quad (1)$$

Viele andere Distanzmaße denkbar, z.B. Hamming-Distanz oder Komprimierungs-bezogene Distanz

Sinnvoll: Gewichtung nach Abstand

BEISPIEL AUS DEM PAPER

Sum of squarred differences (D-SSD): Differenzen der Pixel in den Farbwerten zweier normalisierter Bilder quadrieren und aufsummieren:

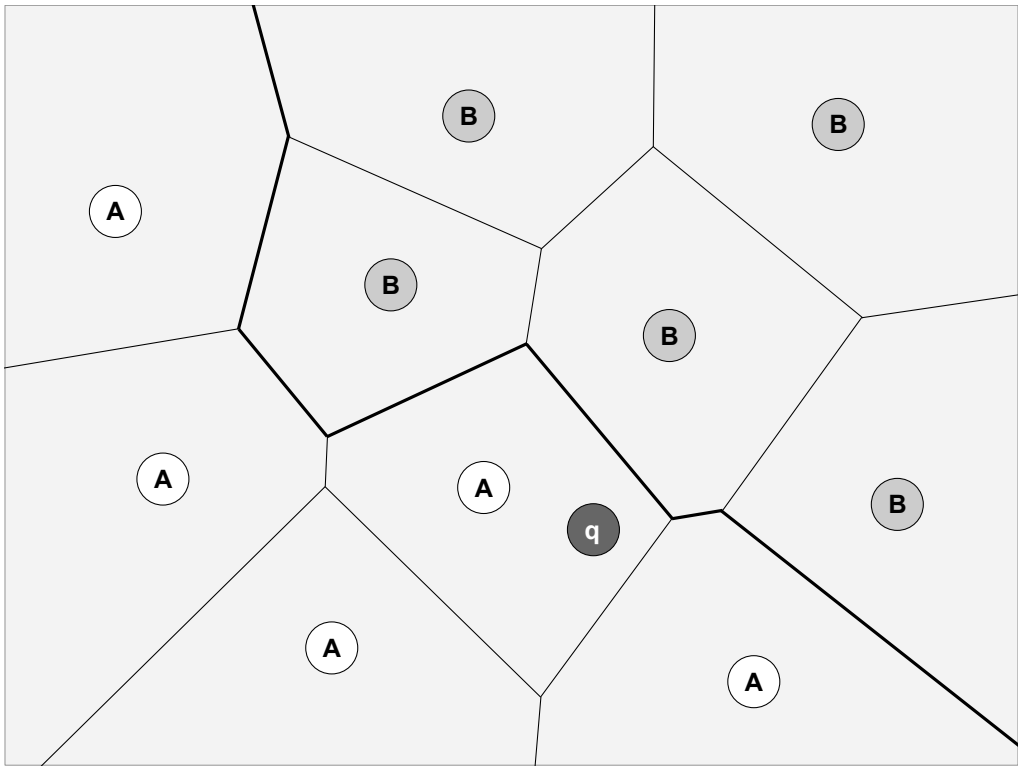
D-SSD

$$D_{ssd}^2 = \sum_{x,y,c} (I_1(x,y,c) - I_2(x,y,c))^2 \quad (2)$$

BEISPIEL AUS DEM PAPER

"Warping" und "Pixel-Shifting"

DATEN UND KLASSIFIKATION



1NN: q wird wie sein nächster Nachbar klassifiziert
Voronoi-Diagramm macht Grenze zwischen Klassen sichtbar

K-NN

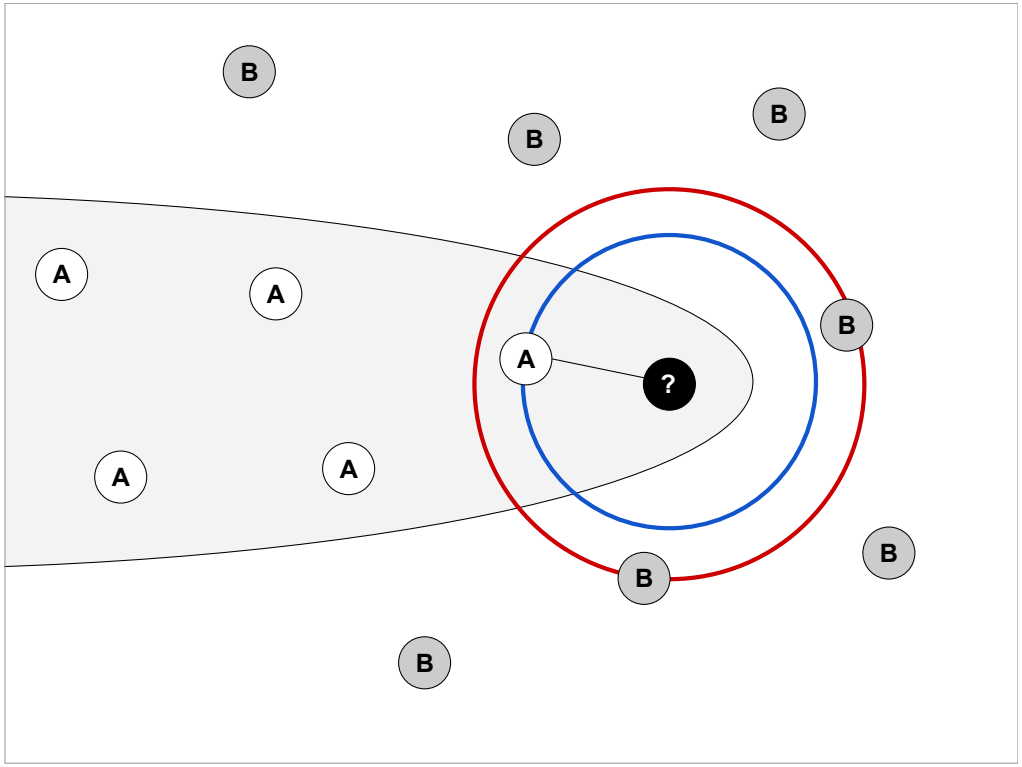
Problem Überanpassung: Ausreißer verfälschen die Klassifikation

Lösung: k-NN – Mehrheitsentscheid unter den k nächsten Nachbarn

k hat großen Einfluss – passende Auswahl ist wichtig

Klassifikation wird stabiler bei größerem k, Details verschwinden

EINFLUSS VON K



Einfluss von k auf den Rand einer Klasse: Die “Spitze” wird kleiner bei größerem k.

BESCHLEUNIGUNG

O(N)

Bei vollständiger Suche wächst Aufwand linear mit der Zahl der Samples

Problem bei hoher Anzahl der Samples

Was tun, um Suche zu beschleunigen?

- Dimensionsreduktion / PCA
- Approximation der nächsten Nachbarn
- Verwendung geeigneter Datenstrukturen und Suchverfahren

VERFAHREN ZUR SCHNELLEN NN-SUCHE

Drei Kategorien:

- Graphen-basierte Verfahren
- Hashing-Verfahren
- Binäre Raumpartitionierungsbäume (engl. “Binary Space Partitioning Trees” / “BSP-Trees”)

Wir schauen uns einen Raumpartitionierungsbaum näher an: Den k-d-Baum

DER K-D-BAUM

- Binärer Suchbaum
- Speichert Punkte aus einem Raum mit k Dimensionen
- Teilung des Raums an jedem Knoten in einer der Dimensionen ("Diskriminator")
- Teilung am Median für balancierten Baum
- Speicherung des Diskriminators mit Teilungs-Wert in Knoten
- Teilung der Teilräume in einer anderen Dimension

Achtung: anderes "k" als bei kNN-Klassifikation!

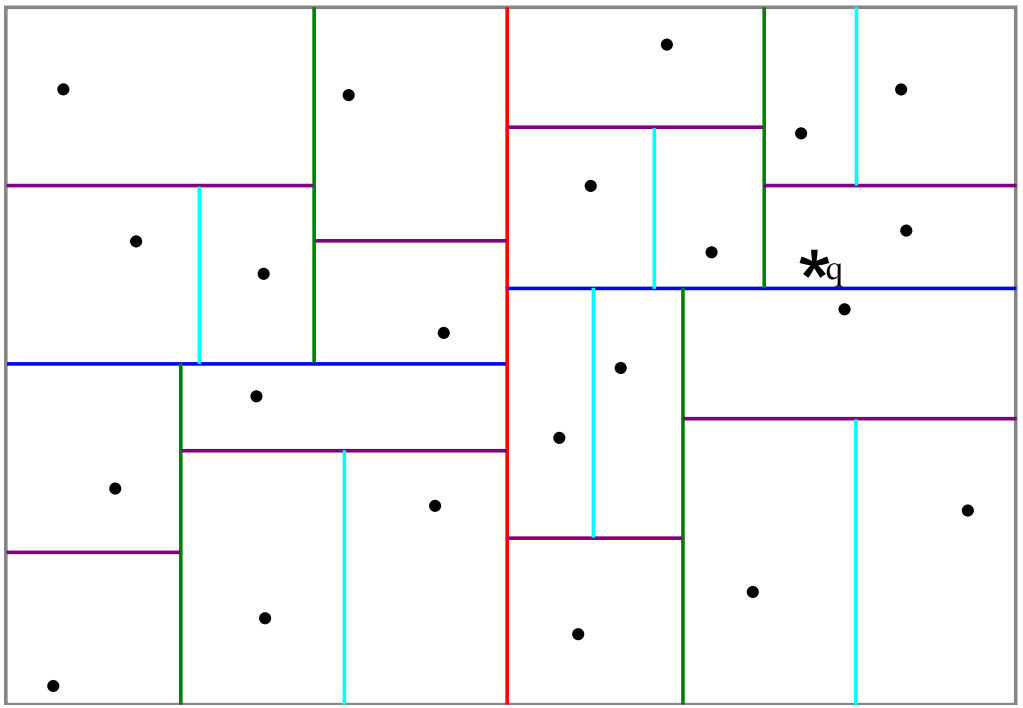
HOMOGENE / INHOMOGENE K-D-BÄUME

Zwei Versionen von k-d-Bäumen:

- Homogen: An jedem Knoten ein Datensatz
- Inhomogen: Innere Knoten enthalten lediglich Schlüssel; Teilung, bis Maximalanzahl von Datensätzen in Teilräumen verbleibt; Datensätze liegen in Blättern ("Buckets")

BEISPIEL

Beispiel für Raumunterteilung:

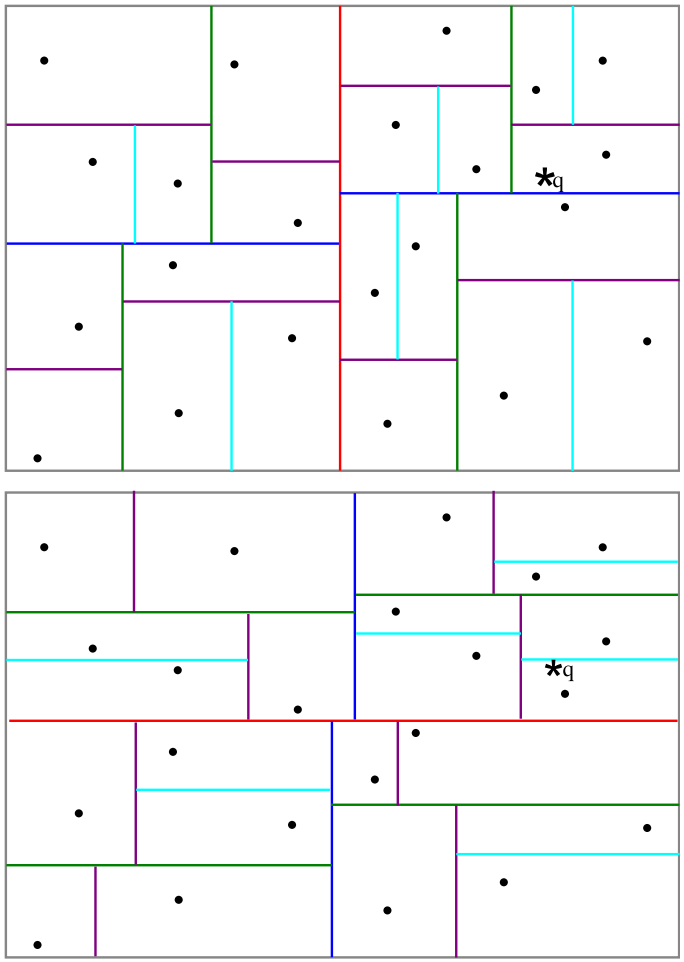


Problem hier: Anfragepunkt nahe an Entscheidungsgrenze, nächster Nachbar auf anderer Seite

RANDOMIZED K-D-TREE-FOREST

Verschiedene Unterteilungen des gleichen Raums

Unten: Anfragepunkt und nächster Nachbar im gleichen Bucket



DISKUSSION

NACHTEILE

- Viele Trainingsdaten nötig
- Speicher für Trainingsdaten dauerhaft belegt
- Ggf. hoher Aufwand für Vergleiche

VORTEILE

- Keine Fehler durch unpassende Modelle
- Breit einsetzbar
- Lernen geht schnell
- Suche gut parallelisierbar

EINSATZMÖGLICHKEITEN

- Automatisches (Bild-)Tagging
- (Vor-) Sortierung von Daten
- Auto-Vervollständigung von Suchphrasen

FAZIT

Hervorragend einsetzbar, wenn...

- Trainingsdaten in ausreichender Menge vorhanden sind
- Die Daten sich sinnvoll vergleichen lassen
- Die nötige Rechenleistung vorhanden ist

QUELLEN

- Muja / Lowe: “Scalable Nearest Neighbour Algorithms for High Dimensional Data” (FLANN-Paper) – Schnelle Suche
- Cunningham / Delany: “k-Nearest Neighbour Classifiers” – Interessante Distanzmaße