


k -nächste-Nachbarn-Schätzung

Mustererkennung und Klassifikation, Vorlesung No. 7¹

M. O. Franz

29.11.2007

¹ falls nicht anders vermerkt, sind die Abbildungen entnommen aus Duda et al., 2001 

Übersicht

- 1 Nächste-Nachbarn-Dichteschätzung
- 2 Nächste-Nachbar-Regel

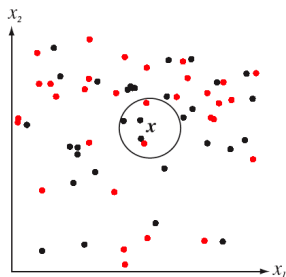
Übersicht

- 1 Nächste-Nachbarn-Dichteschätzung
- 2 Nächste-Nachbar-Regel

k -nächste Nachbarn

Problem des Kerndichteschätzers: Es ist im vornhinein unklar, welche Form des Kerns, welche Breite, welche Orientierung usw. gewählt werden sollte.

Idee: Statt einer festen Fensterfunktion lassen wir das Volumen um den Punkt x so weit wachsen, bis es k Nachbardatenpunkte enthält (d.h. die **k -nächsten Nachbarn**).



- Wenn die Dichte nahen an x groß ist, dann ergibt sich eine kleine Zelle.
- Ist die Dichte niedrig, dann wächst die Zelle solange, bis sie eine Region genügend hoher Dichte trifft.

⇒ Die Fensterfunktion paßt sich automatisch an die Dichte der Datenpunkte an.

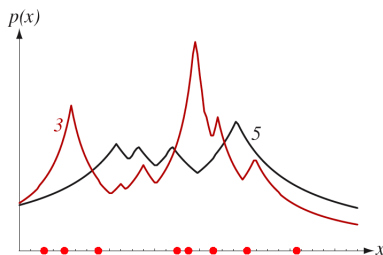
k -Nächste-Nachbarn-Dichteschätzung (1)

Vorgehensweise (gegeben seien n Samples oder **Prototypen**):

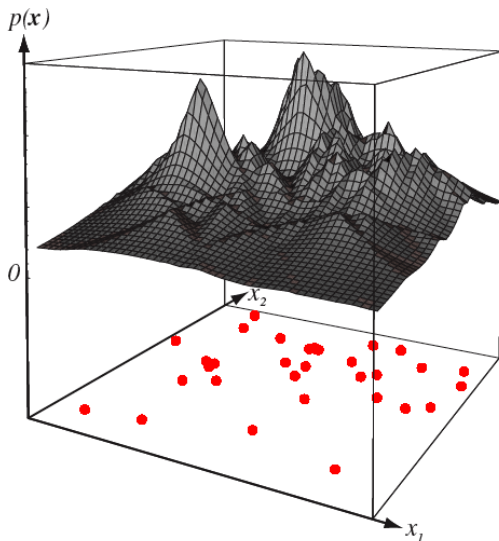
Für jeden Punkt x im Raum

- 1 Bestimme das kleinste Volumen V , das die k -nächsten Nachbarn enthält.
- 2 Dichteschätzung (wie beim Kerndichteschätzer):

$$p(\mathbf{x}) = \frac{k/n}{V}$$



k -Nächste-Nachbarn-Dichteschätzung (2)



Eigenschaften der geschätzten Dichten:

- $p(x)$ ist stetig, aber die Ableitungen nicht ("zerklüftet")
- Die Unstetigkeiten kommen selten an Datenpunkten vor, sondern meistens dazwischen.

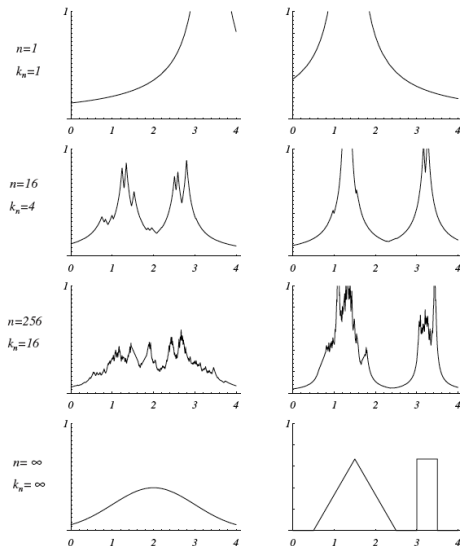
Vergleich mit Kerndichteschätzer

Für den Grenzfall
 $k = n = 1$ ergibt sich die
 eindimensionale Schätzung

$$p(x) = \frac{1}{2|x - x_1|}$$

(divergiert an x_1 , nicht
 integrierbar).

Vorteil (v.a. in
 hochdimensionalen
 Problemen): Dichte sinkt
 nirgends auf 0.



Klassifikation mit k -Nächste-Nachbarn-Dichteschätzung

Wie beim Kerndichteschätzer kann man beweisen, daß die Schätzung für $n \rightarrow \infty$ und $k = \sqrt{n}$ gegen die wahre Dichte konvergiert (falls $p(x)$ an x stetig ist).

Vorgehensweise: (ähnlich wie beim Kerndichteschätzer)

- 1 An jedem Punkt x des Inputraums lassen wir eine Zelle wachsen, bis sie k Datenpunkte enthält. k_i davon gehören zu Klasse ω_i .
- 2 Schätzung der gemeinsamen Verteilung von x und ω_i aus dem Zellvolumen V :

$$p(x, \omega_i) = \frac{k_i/n}{V}$$

- 3 A-posteriori-Wahrscheinlichkeit für Klasse ω_i :

$$p(\omega_i|x) = \frac{p(x, \omega_i)}{\sum_{j=1}^c p(x, \omega_j)} = \frac{k_i}{k},$$

d.h. **der Anteil der Datenpunkte in der Zelle aus Klasse ω_i .**

Übersicht

- 1 Nächste-Nachbarn-Dichteschätzung
- 2 Nächste-Nachbar-Regel**

Nächste-Nachbar-Regel

Eine ähnliche Leistung wie der volle Bayes-Klassifikator läßt sich erzielen durch die

Nächste-Nachbar-Regel

Entscheide immer für die Klasse ω_i des nächstliegenden Trainingsdatenpunkts (Prototypen).

Warum? Wenn die Anzahl der Prototypen groß genug ist, ist es wahrscheinlich, daß der Prototyp x' so nahe an x liegt, das die lokale Schätzung der A-posteriori- Wahrscheinlichkeit aus den Trainingsdaten auch für x gilt, d.h. die Klassifikationsentscheidung nach der Bayes-Regel setzt sich mit hoher Wahrscheinlichkeit auch in der direkten Nachbarschaft fort.

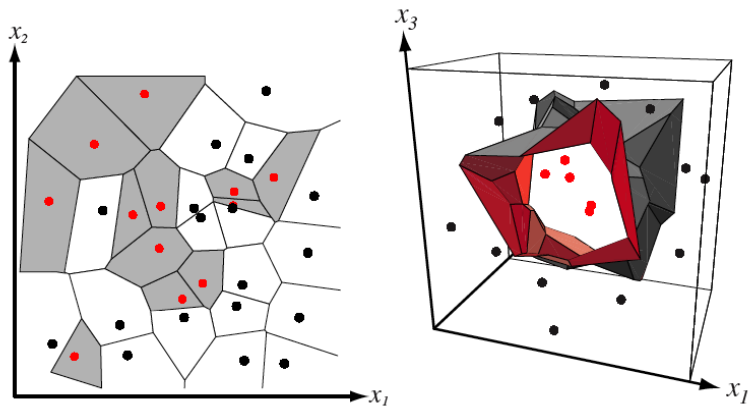
Nächste-Nachbar-Regel und Bayes-Klassifikator

Man kann zeigen (Duda et al., 2001):

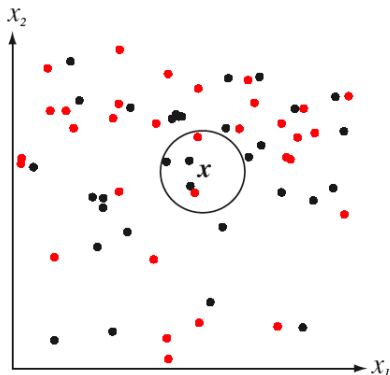
- Die Nächste-Nachbarn-Regel ist **suboptimal**, d.h. i.A. liegt der Klassifikationsfehler über dem des Bayes-Klassifikators, auch bei unendlich vielen Trainingsdaten.
- Aber: der Fehlerrate ist höchstens doppelt so hoch wie beim Bayes-Klassifikator.
- Wenn $p(\omega_i|x)$ hoch ist, dann stimmen Bayes- und Nächste-Nachbar-Regel fast immer überein. Wenn $p(\omega_i|x)$ klein ist, unterscheiden sich beide Entscheidungen häufiger, aber die Fehlerrate ist bei beiden ähnlich.

Die Nächste-Nachbarregel führt zu einer **Partitionierung** des Inputraums in Zellen, die jeweils dem nächsten Prototypen zugewiesen werden. Allen Punkten innerhalb der Zelle wird die Klassenzugehörigkeit des Prototypen zugewiesen (**Voronoi-Mosaik**).

Durch Nächste-Nachbar-Regel erzeugtes Voronoi-Mosaik



k -Nächste-Nachbarn-Regel



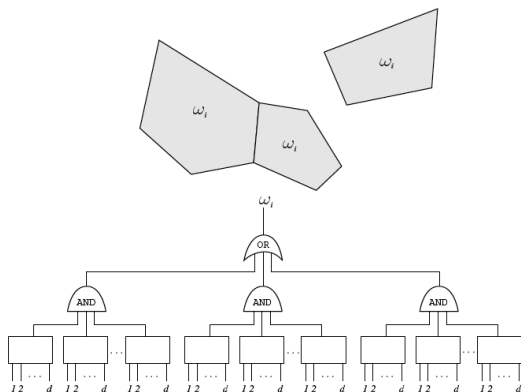
Offensichtliche Erweiterung:

k -Nächste-Nachbarn-Regel

Entscheide immer für die Klasse ω_i der Mehrheit aller Prototypen innerhalb der k -Nächste-Nachbarn-Zelle.

Im Limit unendlich vieler Daten führt die k -Nächste-Nachbarn-Regel auf kleinere Fehlerraten als die Nächste-Nachbar-Regel. Die Schätzungen sind robuster (wegen Mehrheitswahl), aber glätten über eine größere Umgebung.

Komplexität der k -Nächste-Nachbarn-Regel



Naiver Ansatz: Gehe durch alle n Prototypen, berechne Abstand zu Testdatenpunkt x , und wähle Klasse des Prototypen mit minimaler Distanz: $O(n)$ bei serieller Suche, $O(1)$ bei paralleler Suche.

Beschleunigung des NN-Klassifikators

- **Partielle Distanzen.** Für hochdimensionale Probleme: Berechne die Distanz sequentiell, d.h. erst entlang von Dimension x_1 , dann x_2 usw. Falls die Gesamtdistanz über dem momentanen Minimum liegt, brich Berechnung ab und gehe zum nächsten Prototypen.
- **Suchbäume.** Fasse Prototypen in Regionen zusammen und such einen repräsentativen Prototyp, der die dadurch gebildete Metazelle möglichst gut repräsentiert. Achtung: Methode garantiert nicht, daß tatsächlich der nächste Prototyp gefunden wird.
- **Editieren, Kondensieren, Pruning:** Eliminiere alle Prototypen, die komplett von Prototypen der gleichen Klasse umgeben sind (ändert Entscheidungsgrenzen nicht).

Erweiterungen

- Kernmethoden nutzen Fensterfunktionen, die an den Rändern glatt gegen 0 gehen, statt der harten 0-1-Entscheidung durch die k -nächsten Nachbarn.
- Modifikation des Distanzmaßes: Statt euklidischer Distanz werden andere Metriken eingesetzt, die z.B. bestimmte Dimensionen anders gewichten als andere.
- Einsatz von Abstandsmaßen, die unempfindlich gegenüber bestimmten Transformationen der Datenpunkte sind, z.B. Translation, Rotation etc.

Aufgabe 7: NN-Klassifikator

- 1 Konstruieren Sie für die Datenwolke auf dem ausgeteilten Blatt das Voronoi-Diagramm und markieren Sie die Entscheidungsgrenzen des zugehörigen Nächste-Nachbarn-Klassifikators.
- 2 Konstruieren Sie den zugehörigen 2-Nächste-Nachbarn-Klassifikator. Bestimmen Sie dazu für jeden Punkt seine beide nächsten Nachbarn und entscheiden Sie nach der Mehrheit.
- 3 Nehmen wir an, Sie wollen Klasse 1 (Kreuze) detektieren, Klasse 2 (Kreise) beschreibt den Störhintergrund. Was ist die Detektionswahrscheinlichkeit und die Falschalarmrate auf den Trainingsdaten für beide Klassifikatoren? Was erwarten Sie für bisher noch nicht gesehene Testdaten?