Universität
des
Saarlandes
FR 6.2 Informatik

Prof. Dr. Kurt Mehlhorn, Dr. Konstantinos Panagiotou **WiSe 2011/2012**

Übungen zu Computational Thinking

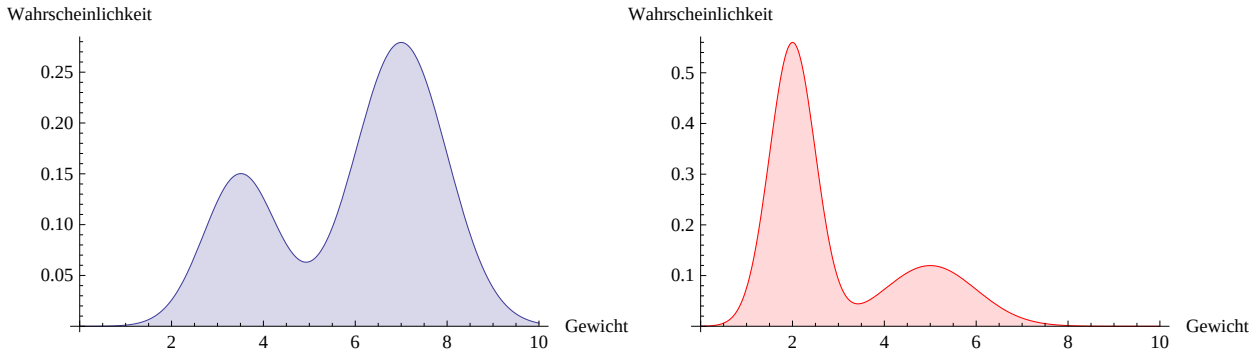
http://www.mpi-inf.mpg.de/departments/d1/teaching/ws11/ct/

Blatt 10 Abgabeschluss: 23.1.12 16:00

Regeln: Bis zum Semesterende müssen mindestens 42% der maximal erreichbaren Punkte aller Übungszettel erworben werden.
Programmcode ist elektronisch per E-Mail abzugeben. Zusätzlich müssen die Ausgaben einer exemplarischen Programmausführung mitgeliefert werden.

Bleistiftaufgaben

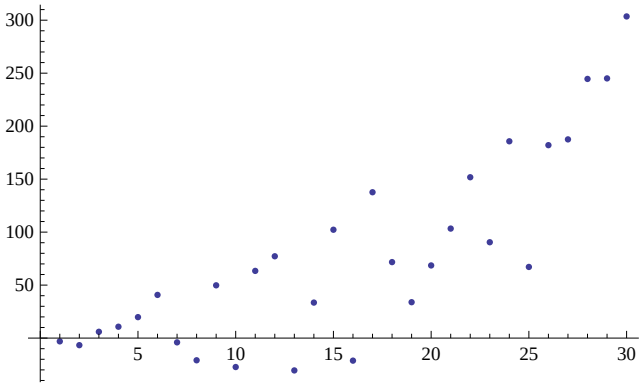
Aufgabe 1 (*15 Punkte*) Sie haben von einer Fischfabrik den Auftrag bekommen eine Maschine zum Sortieren von Fischen zu programmieren. Es werden zwei Arten von Fischen gefangen, „blaue“ Fische und „rote“ Fische. Eine Waage bestimmt für jeden Fisch das Gewicht. Aus einer großen Anzahl an Fischen haben Sie diese Gewichtsverteilungen bestimmt.



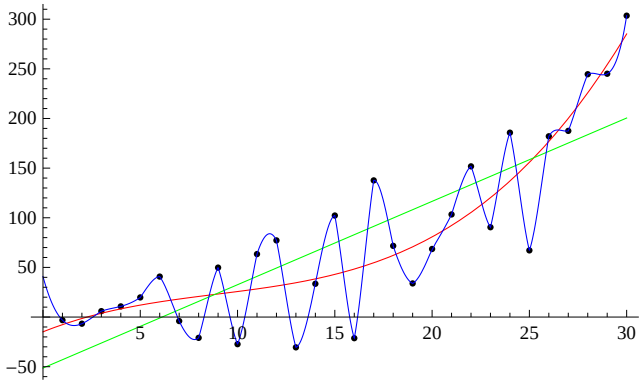
a) Wie würden Sie die ein Fisch mit dem Gewicht 2, 3.5, 5 und 8 klassifizieren? Was ist die allgemeine Regel? Wie ändern sich ihre Aussagen, wenn wir annehmen, dass 80% aller Fische „blau“ sind?

b) Was für Eigenschaften könnten sich außer dem Gewicht noch eignen um die Fische voneinander zu unterscheiden? Wie kann man Eigenschaften, die sich gut zum Klassifizieren eignen von solchen unterscheiden, die wenig nützen?

Aufgabe 2 (*15 Punkte*) Nehmen Sie an Sie wollen aus einer Reihe von (fehlerbehafteten) Messdaten den Zusammenhang zwischen zwei Größen lernen, zum Beispiel den Zusammenhang zwischen Strahlendosis und Krebsrate. Graphisch dargestellt kann das etwa so aussehen:



Da Sie den Zusammenhang noch nicht kennen, wissen Sie nicht was für eine Art von Funktion Sie annehmen sollten. Verschiedene Möglichkeiten könnten so aussehen:



a) Welche Funktion scheint Ihnen am besten geeignet, um Erkenntnisse aus Ihren Messdaten zu ziehen?

b) Wie äußert sich diese Problematik in anderen Bereichen des Lernens, zum Beispiel wenn man aus Photos von Hunden lernen will, welche Eigenschaften Hunde ausmachen? Was kann man tun?

Python

Aufgabe 3 (*30 Punkte*) Wir wollen ein Programm schreiben um handgeschriebene Ziffern zu erkennen. Dazu finden Sie auf der Veranstaltungsseite ein Archiv mit von uns bereitgestelltem Code sowie den Dateien `digits-training.txt` und `digits-testing.txt`, die die Bilddaten enthalten. In `beispiel.py` wird gezeigt, wie man damit arbeiten kann. Sie sollen Klassifikatoren schreiben, die aus `digits-training.txt` lernen und sie auf `digits-testing.txt` ausprobieren.
Erzeugen Sie aussagekräftige Statistiken für die Qualität der folgenden Verfahren und versuchen Sie Gründe für die Unterschiede zu finden.

a) Klassifikation an Hand des nächsten Nachbarn.

b) Klassifikation an Hand der nächsten k Nachbarn.

c) Klassifikation an Hand der Zentren der Ziffernklassen.

d) Klassifikation an Hand der Zentren, die der k-Means Algorithmus findet:

```
p = k zufällige Vektoren aus der Trainingsmenge
solange substantielle updates passieren:
  für alle p_i:
    l_i = alle Punkte, die näher an p_i als an allen anderen p sind
  für alle p_i:
    verschiebe p_i in das Zentrum der Punkte in l_i
```

Wie müssten die Verfahren angepasst werden, um möglichst effizient aus neuen Eingaben kontinuierlich weiterzulernen?