Intelligente Systeme

Praktikum

Aufgabe 2

Dokumentation

Julian Parr, Thomas Jürgensen

Inhaltsverzeichnis

Aufgabenstellung

Gegeben waren drei Datensätze; einer stellte eine mit Daten gefüllte Matrix dar und zwei weitere markierten Koordinaten von A und B Objekten (True- und False-positives) in diesem Datensatz.

Es sollte ein Verfahren entwickelt werden, mit dem A und B Objekte mit statistischen Verfahren automatisch voneinander unterschieden werden können. Hierfür mussten geeignete Kriterien selbst erarbeitet und gewählt werden.

Im weiteren Verlauf der Bearbeitungszeit wurden zwei weitere Datensätze zur Klassifizierung zur Verfügung gestellt, um das Programm testen zu können.

Die Idee

Zuerst ergaben sich die generellen Fragen: Was stellen die Daten genau dar? Wie können diese geeignet visualisiert werden, um für den Menschen verarbeitbar zu gestalten?

Hierfür wurden die Daten der Matrix zunächst als Höhendaten interpretiert, eine Karte daraus erstellt und die Koordinatenpunkte eingefügt.

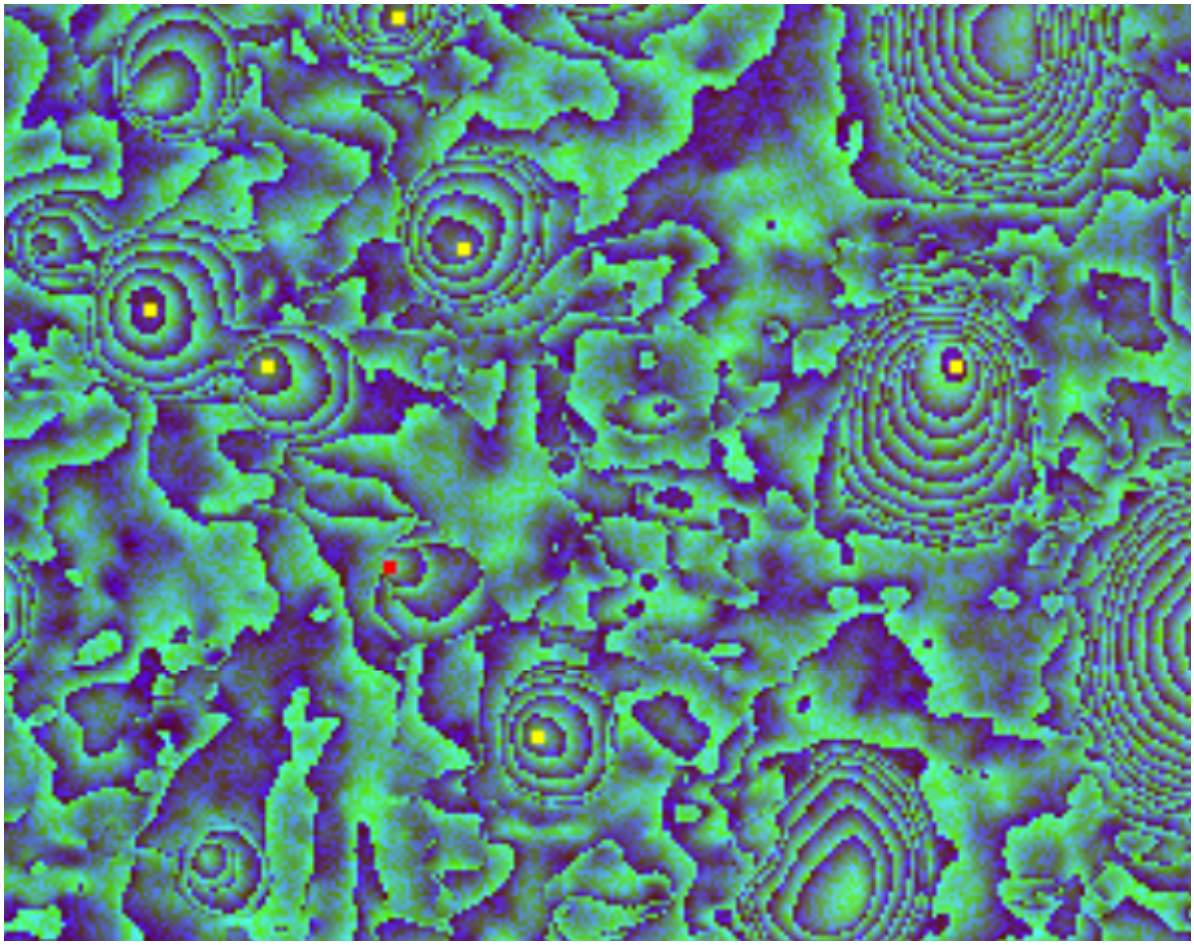

Abbildung 1: Ausschnitt aus der Höhenkarte

Abbildung 1 zeigt einen Ausschnitt aus der Höhenkarte. Zu erkennen sind Strukturen, die in die Höhe gewachsen sind, was aus den Höhenlinien abzulesen ist; die gelben Punkte markieren die gegebenen A-Objekte, die roten Punkte B-Objekte. Im Folgenden werden Objekte als Erhöhungen interpretiert und Berge genannt.

Herleitung von Merkmalen

Aus dem Schluss, dass es sich um bergartige Strukturen handelt, ergaben sich direkt Ideen für Merkmale: Es sollten Breiten, Höhen, Neigung, und Gefällte der Berge ermittelt und untersucht werden. Weiterhin ist auf der Karte abzulesen, dass einige Berge anscheinend weniger spitz als andere zulaufen und somit Plateaus besitzen. Die Plateaugrößen sollten auch als Merkmal dienen.

Die Umsetzung

Die Objektdaten wurden zunächst in in kleinere Teile zerlegt, um Trainings- und Testdatensätze zu erhalten. Als die weiteren Datensätze zur Verfügung gestellt wurden, wurden die älteren Datensätze komplett als Trainingsdatensatz verwendet.

Die Objekte einzeln

Zum einlesen der Objekte wird der Datensatz benötigt. Für jedes Objekt wird ein Hill-Objekt (im Weiteren als Berg benannt) erstellt, welches seine Koordinaten kennt und in der relativen Umgebung diverse Berechnungen durchführt. Weiterhin weiß der Berg, ob er ein A-Objekt, B-Objekt oder ein unbestimmtes Objekt ist. Durch diese Aufteilung ergibt sich die Möglichkeit der Einteilung in Trainings- und Testdaten.

Da jeder Eintrag in der Datenmatrix als Punkt mit X- und Y-Koordinate interpretiert wird, kann für umliegende Punkte ausgehend von der Markierung des Objektes die Entfernung zu ihr berechnet werden. Im Kontext kann dies als durchschnittliche Entfernung der Punkte zur Markierung und somit als Breite des Objektes bezeichnet werden. Der Eintrag selbst wird als Höhe des Punktes interpretiert, wodurch sich die Höhe des Berges und im Zusammenhang mit der Entfernung die Steigung ergibt. Das Ergebnis ist in Abbildung 2 visualisiert. Die blauen Punkte sind die Testobjekte, die roten die B-Objekte des Trainingssatzes, die gelben Punkte die A-Objekte des Trainingssates, die farblichen Unterschiede der Striche markieren die Zusammengehörigkeit zum Berg (weiß) beziehungsweise des Ausläufers (schwarz).

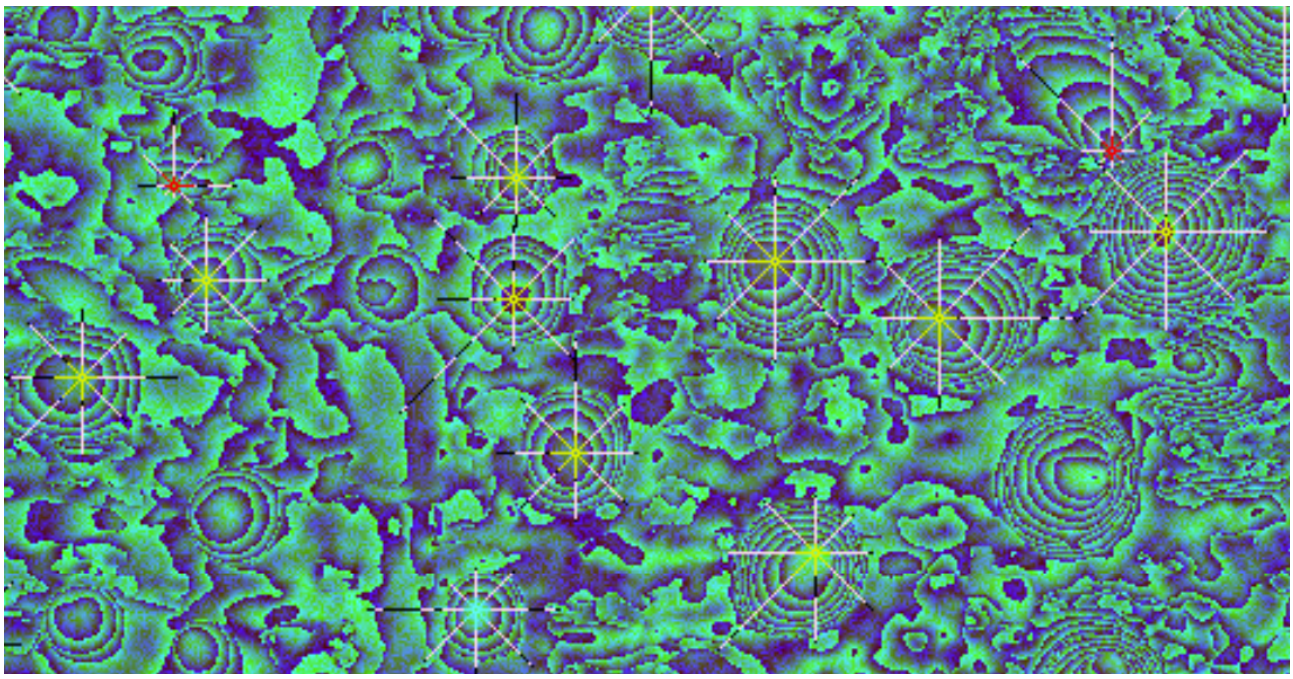

Abbildung 2: Flächenerkennung der Berge

Durch dieses Vorgehen kennt jeder Berg seine Höhe, seinen Durchmesser, die Größe seines Plateaus, sein Gefälle und dessen Größe, seine Neigung, die Größe seines Ausläufers und diverse Variationen dieser Informationen.

Wahl der Parameter

Klassifikationsparameter, welche sich als nicht nützlich erwiesen haben, wurden im Nachhinein aus dem Programmcode entfernt. Übrig geblieben sind die duchschnittliche minimale Höhendifferenz der Steigung zum Bergausläufer („avgMinHeightDifference“) und maximale Differenz zwischen einem Punkt auf dem Berg zur Objektmarkierung („maxDifferenceToHilltop“).

Der erste Parameter gibt an ab welchem Punkt der Berghang zum Ausläufer wird. Somit bestimmt dieser die Größe des Berghanges und beeinflusst somit dieses Merkmal. Bei einer zu hohen Einstellung wird der Berghang kleiner, was beispielsweise Einfluss auf das Kriterium des Ausläufers hat.

Der zweite Parameter dient zur Erkennung des Berggipfels: Dieser ist auf 0.3 Höheneinheiten eingestellt, welche die Größe des Berggipfels beeinflusst. Wenn dieser Wert zu hoch gesetzt wird, wird mehr als nur der Gipfel als solcher erkannt. Ist der Wert zu niedrig, so wird im Extremfall ein Hügel auf dem Gipfel als Ende des Berges erkannt. Dies kann Auswirkungen auf die Kriterien

haben; so kann beispielsweise das Kriterium des Berghanges negativ beeinflusst werden, da der Berghang unter Umständen nicht vollständig erkannt wird.

Klassifikation von Testobjekten

Aus den Trainingsdaten werden die für das gewählte Merkmal relevante Durchschnittswerte für die Objektklassen (also Objekt der Klasse A oder B) berechnet und in den Bergen selbst gespeichert..

Für jedes Testobjekt werden die Daten des relevanten Merkmals mit den Daten jedes Objektes aus dem Trainingsset verglichen. Wenn das Merkmal des Testobjektes näher am Durchschnittswert der Objektklasse ist als das Trainingsobjekt selbst, wird das Testobjekt dieser Objektklasse als „ähnlich“ eingestuft und die Anzahl der Ähnlichkeiten mitgezählt. Dieser Wert wird durch die Anzahl der Objekte dieser Klasse dividiert, woraus sich die Wahrscheinlichkeit der Zugehörigkeit zu dieser Klasse ergibt. Dieses Verfahren wird auf alle Trainingsdaten angewendet.

Die gewählten Klassifizierungsmerkmale sind statistisch unabhängig und können somit zur Gesamtklassifizierung multipliziert werden. Somit wird jedes Testobjekt eindeutig einer Objektklasse zugewiesen.

Güte der Merkmale

Viele Merkmale wurden aus dem Programmcode wieder entfernt, da sie sich als schlechte Kriterien erwiesen haben. So war beispielsweise die relative Distanzabweichung zum Bergmittelpunkt in X- und Y-Richtung ein Kriterium, welches sämtliche Klassifizierungsversuche im Zusammenhang mit anderen Merkmalen verschlechtert hat. Einzeln betrachtet unterscheidet dieses Kriterium offenbar kaum zwischen A- und B-Objekten und verfälscht weitere Klassifizierungsversuche.

Als sehr gute Merkmale haben sich die relative Höhe der Berge, also die Höhe ungeachtet der Bodenhöhe, und die Größe des Berghanges in X- und Y-Richtung erwiesen – offenbar unterscheiden sich die Objektklassen in diesen Merkmalen stark.

Als weiteres Hilfskriterium wurde die absolute Höhe gewählt. A-Objekte scheinen häufiger in höheren Ebenen vorzukommen als B-Objekte. Dieses Kriterium ist einzeln gesehen eher schwach, jedoch optimiert es das Zusammenspiel der starken Kriterien.

Tabelle 1 zeigt die getesteten Kriterien und deren Güte. Diese hat sich im Verlauf der Tests entwickelt; schlechte Kriterien wurden aus dem Programmcode entfernt.

Merkmal	**Güte**
Neigungswinkel	Schlecht
Relative Höhe	Gut
Allgemeine Breite	Schlecht
Größe des Plateaus	Schlecht*
Größe des Berghanges	Gut
Größe der Bergausläufer	Sehr Schlecht
Rundheit	Schlecht*
Allgemeine Höhe	Mittel
relative Distanzabweichung zum Bergmittelpunkt in X- und Y-Richtung	Sehr schlecht

Tabelle 1: Merkmale und deren Güte

** dieses Merkmal hat sich bei diesen Tests als schlecht erwiesen; Es liegt jedoch die Vermutung nahe, dass sich diese Merkmale bei einer anderen Herangehensweise als wertvoll erweisen könnten.*

Das entstandene Programm

Das Ergebnis dieser Aufgabe ist ein Java-Programm, welches mit Hilfe von Trainingsdaten ein Testset klassifizieren kann. Mit den gegebenen Trainingsdaten A0 und B0 wird für die Testdaten A1 eine Rate von 78% und für B1 von 85% korrekter Klassifikation erreicht. Es verwendet das Vorgehen, welches in dieser Ausarbeitung beschrieben wurde und nutzt die Merkmale der relativen Berghöhe, Bergbreite und absolute Berghöhe.

Die Ausgaben erfolgen auf der Konsole.

Funktionen / HowTo

Um Daten zu klassifizieren, müssen in der main-Methode Pfade zu Trainings- und Testset gesetzt werden. Es findet eine Überprüfung auf Windows und Linux-Betriebssysteme statt. Auf iOS wird nicht geprüft und wird als Windows behandelt.

Es bietet die Funktion der Erstellung der Höhenkarte mit eingezeichneten Bergen (wie in Abbildung 2); hierfür muss in der logic/Hill-Klasse der Parameter “printImage” auf “True” gesetzt werden. Der Erstellpfad muss in der main-Methode gesetzt werden.

Um Daten zu klassifizieren, müssen in der main-Methode Pfade zu Trainings- und Testset gesetzt werden.

Es bietet die Funktion der Erstellung der Höhenkarte; hierfür muss in der logic/Hill-Klasse der Parameter “printImage” auf “True” gesetzt werden. Der Erstellpfad muss in der main-Methode gesetzt werden. *Anmerkung: Wird die printImage Methode verwendet, findet keine Klassifizierung statt. Das Erstellen der Karte verändert die Höhenwerte der Objekte, da diese direkt als Farbwerte interpretiert und verändert werden.*