Setup

1. Stellen Sie sicher, dass die Datei mit den zu analysierenden Daten im .csv Format vorhanden ist.
2. Platzieren Sie die Daten-Datei in dem Dataset-Ordner (in dem Projektordner).
3. Installieren Sie Python 3.7.
4. Öffnen Sie die Command Line und navigieren Sie zu dem Projekt-Ordner.
5. $ python \_\_main.py\_\_
6. Folgen Sie den Anweisungen des Programs.
7. Sie finden den Output in der Datei output.txt.

Aufgabenstellung

Es wird eine CSV-Datei bereitgestellt, welche aus einem Video Gesichts-Charakteristika (bzw. Features) extrahiert. In den ersten vier Feature-Spalten wird die Position des Frames sowie die Größe des Frames angegeben. In den darauffolgenden Spalten werden in folgender Reihenfolge tatsächliche Features im Gesicht des Beobachteten dokumentiert:

* Stirnfalte
* Augenöffnung und Brauenabstand auf jeweils beiden Seiten
* horizontale und vertikale Falten auf der Nasenwurzel
* Faltenbildung auf beiden Wangenseiten
* Mundöffnung

Jede Zeile in der CSV-Datei stellt einen Frame des Videos dar. In der Aufgabenstellung ist eine bestimmte Reihenfolge bzw. Abfolge von Emotionen festgelegt.

Die Aufgabe ist es nun, für jeden Frame der Videosequenz (dem Programm im Format einer CSV-Datei bereitgestellt) mithilfe der ‚Dempster-Shafer-Regel‘ eine Emotion auszugeben, basierend auf den eingelesenen Features. Zwar sollen erst die Emotionen der letzten 25 Frames ausgegeben werden, da der Gruppe aber unbekannt ist wie die Testdaten bei der Bewertung aussehen, wird für jeden Frame die erkannte Emotion ausgegeben. Zusätzlich ist zu beachten, dass bei den Testdaten der Bewertung unvollständige Featurewerte auftreten können.

Theorie

# Mapping: Aufgabenstellung auf Dempster-Shafer-Regel

In dieser Bearbeitung existiert eine Menge von Alternativen Ω , die Menge aller möglichen Emotionen. Die Alternativenmenge ist durch die Aufgabenstellung vollständig gegeben und es ist anzunehmen, dass diese sich gegenseitig ausschließen. Somit sind alle Voraussetzungen für die Evidenztheorie erfüllt.

Gehen wir nun für die Emotionsbestimmung von einem Frame aus. Jedes gemessene Feature (ab Spalte 5) wird zu einem Basismaß. Dieses Basismaß besteht immer aus mindestens 2 Einträgen. Ein Eintrag setzt sich zusammen aus einer Menge an Emotionen, auch genannt ‚fokale Menge‘ und deren Konfidenz. Jede Evidenz hat immer ein Ω - Element, also einen Eintrag, in der die fokale Menge die gesamte Menge von Alternativen ist.

Schlussendlich werden alle Basismaße, also alle gemessenen Features, akkumuliert. Daraus entsteht das Basismaß mgesamt.

Aus den Basismaßen in mgesamt kann dann für jede Emotion die Plausibilität errechnet werden. Die Emotion mit der höchsten Plausibilität wird schlussendlich als erkannte Emotion ausgegeben.

# Mapping: Datenaufbereitung

Die Daten, die in der CSV-Datei gegeben werden, können nicht genauso für den Algorithmus übernommen werden, da sie allein nicht für Aussagen über mögliche Emotionen ausreichen. In der Aufgabenstellung ist wörtlich beschrieben, welche Merkmale mit welcher Ausprägung für welche Emotion sprechen. Beispielsweise sind bei Traurigkeit eine hohe Zahl an Stirnfalten und eine große Augenöffnung festzustellen. Um diese Zuordnung vornehmen zu können, müssen die Zahlen also in Kategorien groß, mittel und klein (engl. large, medium, small) eingeordnet werden. Ein erster Ansatz wäre, manuell Grenzen für die Kategorien festzulegen. Dies funktioniert jedoch nur bei einer Datei, da sich die Zahlenbereiche in den eingegebenen Dateien gravierend unterscheiden können. Die Zahlen müssen also normalisiert werden.

Bei genauerer Untersuchung der Zahlen konnte festgestellt werden, dass sich nicht nur die Messwerte, sondern auch die Position und Größe der aufgenommenen Frames unterschieden. Bei der Umsetzung werden die Messwerte nun durch die Höhe des jeweiligen Frames (Spalte „ylow“) geteilt. Auf diese Weise erhält man über sämtliche Dateien hinweg Zahlen in den gleichen Zahlenbereichen. Da es sich nach der Normalisierung zum Teil um sehr kleine Werte handelt, werden einige Spalten mit Faktoren multipliziert, um die Auswertung zu vereinfachen. Anhand dieser neuen Werte können nun Grenzwerte für die einzelnen Gesichtsmerkmale und ihre Ausprägungen festgelegt werden. In allen Spalten konnte jeweils drei Wertebereiche gefunden werden, die genügend großen Abstand zueinander haben, um sie als Kategorien zu erkennen.

Im nächsten Schritt werden mögliche Emotionen für einen Frame anhand der Größen der Gesichtsmerkmale in diesem ausgewählt. Dazu wurde der Text aus der Aufgabenstellung auf eine Datenstruktur übertragen und alle Emotionen, auf die die Merkmale passen aufgenommen.

Der letzte Schritt beinhaltet die Ermittlung der Konfidenz *K* zu diesen fokalen Mengen *M*. Die Konfidenz ist ein Maß für die Zuverlässigkeit der Messung. Aus der Aufgabenstellung geht nicht hervor, unter welchen Umständen die Messung vorgenommen wurde und auch nicht, wie die Zuverlässigkeit bewertet wird.

Aus diesem Grund nehmen wir an, dass die Wahrscheinlichkeit dafür, dass eine Menge an Emotionen durch dieses Merkmal ausgewählt wird, die Konfidenz in angemessener Weise beschreiben kann. Sie wird daher in der Lösung berechnet mit: . Ein weiterer Ansatz war, die Messgenauigkeit mit einzubeziehen, indem die Wahrscheinlichkeit, dass eine Emotion auch mit dem Merkmal in einer anderen Größe auftreten kann, einberechnet wurde.

Formalisiert:

Nach Umsetzung dieser Kalkulation verschlechterten sich jedoch die Ergebnisse. Besonders auffällig waren die ersten 5 Frames jeder Datei. Diese sollten laut Aufgabenstellung stets den Gesichtsausdruck „neutral“ darstellen. Bei der ersten Berechnung für die Konfidenz konnte dieses Ergebnis bestätigt werden, während mit dem zweiten Ansatz mit sehr hoher Plausibilität eine andere Emotion erkannt wurde. Daher ist in der finalen Umsetzung der simple Ansatz zu finden.

Methode

# Umsetzung Algorithmus

Gesteuert wird der gesamte Ablauf des Programms von der Main-Klasse aus.

Input CSV

Output Datei

Main

Dempster-S-Rule

Basismaß

Normalisierung

Entry

Der Ablauf ist in der Grafik dargestellt. Es wird in der Main der Input in Form einer CSV Datei eingelesen. Der Inhalt dieser wird dann in einer Schleife pro Zeile/Frame und zur Normalisierung weitergegeben. Die Normalisierung ist ein langer Prozess und deshalb im nächsten Subkapitel beschrieben.

Zurück kommt für den aktuellen Frame eine Liste; für jedes Feature im Input wird die Fokalmenge sowie die Konfidenz für diese Fokalmenge zurückgegeben. Im nächsten Schritt wird eine Liste von ‚Basis-Basismaßen‘ angelegt, es wird also für jedes Feature ein Basismaß angelegt. Diese Basismaßklasse greift im Konstructor auf die Klasse Entry zu. Es wird zuerst ein Eintrag für die gegebene Fokalmenge und Konfizenz angelegt und zu der Entry-List dieser Evidenz hinzugefügt. Folgend daraus wird ebenfalls der Omega-Eintrag angelegt mit der Gegenwahrscheinlichkeit bzw. -Konfidenz.

Im nächsten Schritt wird durch diese ‚Basis-Basismaße‘-Liste iteriert und sie werden Schritt für Schritt akkumuliert. Dazu werden jeweils immer zwei Basismaße an die Dempster-Rule Klasse übergeben. Diese greift wiederum nocheinmal auf die Basismaßklasse zu, um als Output auch ein Basismaß zurückgeben zu können. Dort wird dann tabellenartig jeder Eintrag miteinander verrechnet. Gegeben seien dazu die Basismaße mk und mj, die akkumuliert werden sollen zum Output-Basismaß mo. Die DSR-Klasse unterscheidet dazu folgende Fälle:

1. Eines der Einträge ist Omega:   
   Es wird in mo ein Eintrag angelegt mit der Konfidenz bestimmt durch das Produkt der Konfidenzen von mk und mj und der fokalen Menge gleich der fokalen Menge des Elements das nicht Omega ist.
2. Beide Einträge sind Omega:  
   Es wird in dem Output-Basismaß ein Eintrag angelegt, mit der Konfidenz bestimmt durch das Produkt der beiden Input-Einträge und der fokalen Menge Omega
3. Keines der Einträge ist Omega:  
   Hier werden zuerst die beiden fokalen Mengen miteinander verglichen und die Schnittmenge betrachtet
   1. Die Schnittmenge ist leer  
      Es wird trotzdem ein Eintrag in dem Output-Basismaß mit entsprechender Wahrscheinlichkeit angelegt, dies wird später separat bearbeitet
   2. Die Schnittmenge ist nicht leer

Es wird in dem Output-Basismaß ein Eintrag angelegt mit der Konfidenz bestimmt durch das Produkt der beiden Input-Einträge und der fokalen Menge gleich der bestimmten Schnittmenge

In jedem dieser Schritte wird vor dem Erstellen des Eintrags sicher gestellt, dass ein Eintrag mit derselben fokalen Menge nicht schon in dem Output-Basismaß existiert. Ist dies der Fall, wird die ausgerechnete Konfidenz des aktuellen Eintrags der Konfidenz des bereits existierenden Eintrags hinzuaddiert. Die einzige Ausnahme, in der dieser Vergleich nicht durchgeführt wird ist der Fall, in dem beide Input-Einträge Omega sind, da dieser Fall per Definition nur einmal pro Akkumulation vorkommt.

Als letzter Schritt bevor das Output-Basismaß zurückgegeben werden kann, werden die Einträge mit leeren fokalen Mengen bereinigt. Dazu wird die Konfidenz dieses Eintrags bestimmt und dann anhand des Faktors Fkorrektur = die Konfidenzen der restlichen Einträge neu gesetzt.

Es ist zu beachten, dass nach der Neuberechnung nochmal ein Kontrollcheck implementiert wurde, der überprüft, ob die Konfidenzen addiert noch immer 1 ergeben.

Bei dem Kontrollcheck wie auch bei den Tests, die zur Kontrolle der Funktionen implementiert wurden, mussten wir leider auf Runden zurückgreifen. Da die Konfidenzen als float implementiert wurden, kam es zu Rundungsfehlern bei dem Ergebnis. Wir haben uns deshalb für den Check und die Tests dafür entschieden, die zu vergleichenden Ergebnisse auf 5 Nachkommastellen zu runden.

Wurde das Basismaß mgesamt aus allen vorhergehenden Basismaßen berechnet, wird auf dieses Basismaß die Funktion calculate\_plausibility() aufgerufen. Diese gibt eine Liste aller vorhandenen Emotionen in mgesamt  aus, sowie ihre Plausibilität. In der Main Klasse wird dann die Emotion mit der höchsten Plausibilität als Output gewählt und zusammen mit ihrer Plausibilität in die Output-Datei geschrieben.

Ergebnisse

Sie finden die outputdatein hierfür im Ordner blibla

Datei 1:

Ersten 5 sind neutral, das ist richtig, mit decent plausibilität (immer 0.68), nochmal auf reihenfolge verweisen und alle 4 pärchen neutral wurden erkannt, ohne outlier oder fehler.

Nächsten 5 sind unbekannt, bei uns aber uniform erkannt, mit hoher plausi, disgust. Komplett erkannt. (0.98)

Zweite unbekannte scheint joy zu sein mit einem outlier fear, joy plausibilität ist quite hoch (schwankt zwischen 0.62 bis 0.92), fear plausi ist nicht so hoch (0.57), deshalb wahrscheinlich outlier.

Letzes unbekanntes neutral, joy, joy, joy, neutral. Das kann sein weil neutral ein übergangsausdruck ist, da die plausibilität von diesem frame sich von den anderen unterscheidet. Leider ist die plausibilität mit 56 bei joy nicht so hoch.

Ab dann beginnt durcheinander:

Da die meisten Emotionen mit zwei oder mehr frames (in pärchen) erkannt wurden, könnte man annehemen das das richitg ist. Besonders sicher sind wir uns hier bei digust (0.98) und sorrow (0.95)

Datei 2:

Für die vier pärchen neutral gleich wie oben.

Disgust wieder, gleich hoch wie vorhin.

Hier wieder die zweite unbekannte emotion joy, hier jedoch komplett 5x mit höherer plausi (ab 0.79)

Dritte unbekannte scheint hier neutral zu sein. Die plausi ist nicht hoch mit 0.41 bzw 0.42. Wir verweisen hier auf die vorherige datei, in der joy schon nicht klar von neutral unterschieden werden konnte. Es kann sein, dass sich diese beiden emotionen ählnich sind und nicht immer genau unterschieden werdne können. Unsere Vermutung ist folgende: in dem zweiten Dataset ist die bounding box kleiner als in der ersten, also komprimiert, weshalb es wahrscheinlich zu ungenauigkeiten in den angegebenen Zahlen kommen kann.

Hier tritt jede Emotion in paaren von mindestens zwei auf, besonders sicher sind wir uns wieder bei disgust und sorrow, und konnten auch joy mit einer hohen plausibilität erkennen.

Datei 3:

Datei ist sehr ähnlich zu date 2.

Generell erkennen wir disgust und sorrow immer sehr hoch. Joy und neutral unterscheiden wir nur schwer. Das kann sein weil das ‚muster‘ gleich ist, die ausprägung ist größer. Explain here. Dann nochmal bounding box weil messfehler/abweichung, boom, erklärung.

Wir vermuten schlussendlich, dass alle drei datasets die gleiche Reihenfolge von Emotionen sind, der unterschied sind die bounding box größen.

Future Work

Bei der Bearbeitung der Aufgabenstellung gab es in der Gruppe zwei Vorschläge für mögliche Erweiterungen, die aufgrund der Zeit leider nicht mehr eingebunden werden konnten. Wir möchten dies aber trotzdem nennen, da wir denken, dass diese Erweiterungen die Analyse nochmals verbessern könnten.

**Kontext bei real-world Anwendung**

Für diese Bearbeitung haben wir uns dazu entschieden, für jeden Frame die erkannte Emotion auszugeben. Laut der Aufgabenstellung ändert sich (zumindest am Anfang) in bestimmten Abständen die Emotion. Nun ist es nicht unwahrscheinlich, dass die Emotion nicht durchgehend klar zu erkennen ist oder die Testperson beim Wechseln des Gesichtsausdrucks keine ‚klare‘ Emotion zeigen kann. In einem realen Anwendungsszenario wäre es also sinnvoll – wenn man die Abfolge und Länge der Emotionen kennt – die Analyse eines Frames mit Kontext zu betrachten. Folgendes Outputbeispiel verdeutlicht dies (es ist bekannt, dass eine Emotion für 5 Sekunden gehalten wird):

|  |  |
| --- | --- |
| Fear – 0.97  Fear – 0.97  Fear – 0.97  Fear – 0.97  Neutral – 0.56 | Joy – 0.82  Joy – 0.82  Joy – 0.82  Sadness – 0.60  Joy – 0.82 |

Hier ist deutlich zu sehen, dass man leicht Fehlinterpretationen eliminieren kann, in dem man bei einem oder zwei Outliern diese ‚ignoriert‘ bzw. nur eine Emotion pro 5 Sekunden ausgibt.

**K-Means Clustering**

Die aus der Datenverarbeitung gewonnenen Werte können in der Anwendung so zur Berechnung genutzt werden. Dennoch wurden die Intervalle für die Kategorisierung nach Größe/ Ausprägung des Merkmals per Hand festgelegt. Bei der Nutzung einer solchen Anwendung in der Welt sind diese Werte womöglich je nach Person unterschiedlich und die Intervalle unterschiedlich groß. Aus diesem Grund wäre ein Ansatz, die Wertebereiche für die Größen festzulegen, auf alle Zahlen einer Spalte in der CSV-Datei einen K-Means-Clustering-Algorithmus laufen zu lassen. Dieser soll die Zahlen in 3 Cluster unterteilen. Dasjenige Cluster, das die größten Zahlen enthält, entspricht den Zahlen für Kategorie „l“, das mit den kleinsten entspricht „s“ und das mittlere der Kategorie „m“. Auf diese Weise können die Intervalle für jeden Datensatz und jede Person individuell erstellt werden.