

Chapter 1 - Einführung in die Informationsvisualisierung

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Was sind Visualisierungen?

⇒ das Nutzen von Computer-unterstützten, interaktiven, visuellen Repräsentationen von abstrakten Daten um Kognition zu verstärken. (Card, Machinklay, Shneiderman, 1999)

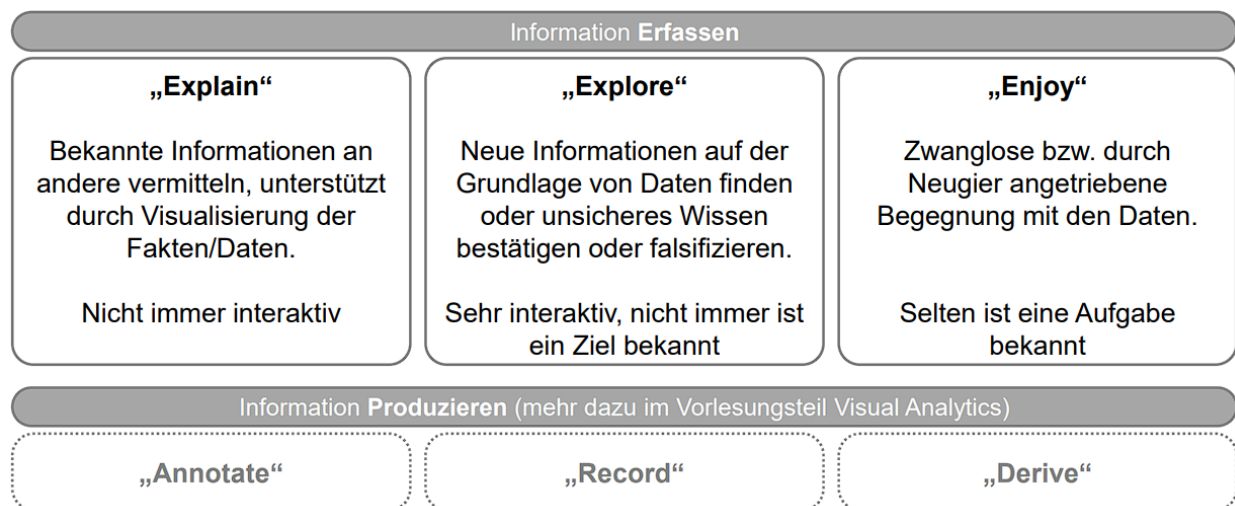
Warum überhaupt Visualisierungen? Visualisierungen sollen dem Menschen helfen, Entscheidungen zu treffen. Entscheidungen beruhen (im besten Fall) auf Daten und Informationen, welche so gut und objektiv wie möglich visualisiert werden sollen.

- Prozess: Daten → Visualisierung → Wissen → Entscheidung

1.1 Ascombe's Quartet

⇒ eine Datenmenge, die aus vier Mengen von Datenpunkten besteht (jeweils mit x und y). Diese Datenpunkte haben nahezu identische Eigenschaften, die jedoch alle andere Verhältnisse darstellen. Erst durch eine Visualisierung wird die Erkennung der Unterschiede sichtbar.

1.2 Amplify Cognition



2 Regel Nr. 1 - Kenne die Aufgabe, nimm ein gutes Werkzeug

⇒ es gibt viele Designs, nach denen eine Visualisierung entworfen werden kann. Man muss ein gutes Design wählen, sodass das Ziel der Visualisierung erfüllt werden kann.

Die Aufgaben unterscheiden sich in wesentlichen Details:

- Welche Information ist als *bekannt* vorausgesetzt?
- Welche Informationen werden gesucht?
- Was soll mit der neuen Information gemacht werden? (z.B. Vergleich, Bezug, etc.)

→ Design der Visualisierung beeinflusst, wie gut bekannte Informationen zu finden sind und wie gut neue Information zu lesen ist. (Kernaufgabe Designer: beide Aufgaben so einfach wie möglich zu erfüllen)

Werkzeugwahl hängt von den Daten, der Aufgabe und den Nutzern ab

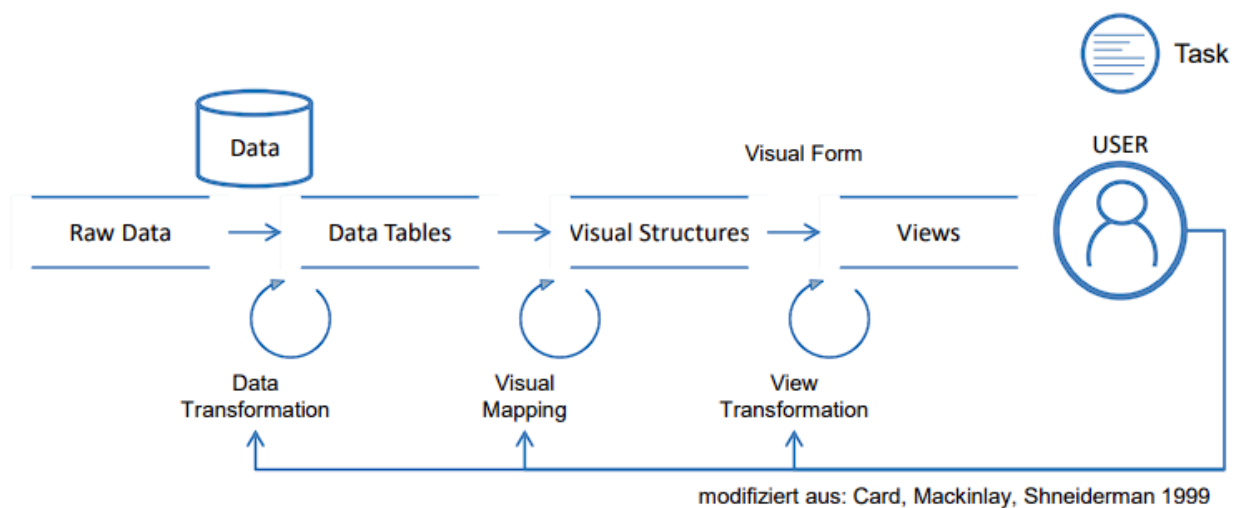
- manche Visualisierungstechniken sind nur für bestimmte *Datentypen* und *Datenstrukturen* geeignet. Die Techniken erleichtern nur bestimmte Aufgaben.
- Jedes Design basiert auf Positionierung:
 - Welche Daten sind dem Nutzer am wichtigsten?
 - Welche Aufgabe ist dem Nutzer am wichtigsten?
 - Kann/Will der Nutzer ein neues Design erlernen?

3 Fehler in Visualisierungsnutzung und -design

- Designs beeinträchtigen die Möglichkeit der Low-Level-Wahrnehmung
- Designs beeinträchtigen Mustererkennung
- Design erschwert Lesbarkeit von Charts
 - ⇒ Visual Literacy ist eine Menge an Eigenschaften, um einem Nutzer die Möglichkeit zu geben effektiv Visualisierungen zu finden, interpretieren, evaluieren und zu nutzen.
- Schlussfolgerungen basieren auf falschen Annahmen

4 Informationsvisualisierungsprozess

Das Datenflussmodell definiert Visualisierung als Transformationskette (Daten \rightarrow Bild). Gleichzeitig ist es ein Strukturmodell für den Aufbau von Visualisierungen und es definiert, wie die Reaktion auf Änderungen organisiert werden kann.



Data Transformation: Rohdaten werden zu Datentabellen umgewandelt, weil häufig das Format der Rohdaten nicht zur Visualisierung passen.

Visual Mapping: Hier wird definiert, welche Datenvariablen auf welche visuellen Strukturen abgebildet werden. Welche visuelle Strukturen hierbei zur Verfügung stehen, hängt teilweise vom Visualisierungstyp ab. Es wird zum Beispiel definiert, was die x-Achse darstellt, die y-Achse darstellt, und was die Farben repräsentieren, etc.

View Transformation: Es wird beschrieben, welche Datenwerte auf welche visuellen Variablenwerte der visuellen Struktur abgebildet werden sollen. Die View Transformation macht die Visualisierung als View sichtbar.

Achtung: Visual Mapping beschreibt z.B. ein Alter auf eine Position abgebildet wird. Die View Transformation beschreibt, welches Alter auf welche Position abgebildet wird.

Interaktion: die Möglichkeit, die Transformationen und das Visual Mapping zu steuern, sodass der Nutzer die Sicht auf Daten an neue Fragen/Aufgaben anpassen kann, ohne auf eine neue Visualisierung zu warten. Interaktion verschiebt Kosten vom Designer zum Nutzer, weil der Designer sich die genaue Anforderungsanalyse spart, und der Nutzer die Interaktion dann lernen muss.

Chapter 2 - die Visuelle Abbildung

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Visuelle Abbildungen

- einfache Regel (die nicht immer gilt): eine Datenvariable \rightarrow eine visuelle Struktur
- Position, auf die abgebildet wird, ist manchmal nur von einem einzelnen Wert abhängig, oder von [allen] anderen Datenwerten abhängig.

2 Datentypen und -strukturen

2.1 Datentypen

\Rightarrow Unterscheiden sich vor allem durch Operatoren, die auf Ihnen angewendet werden können.

- **Nominal/Kategorisch:** Menge von Werten, auf denen Gleichheit ($==$) und Ungleichheit ($!=$) definiert sind. Beispiele: Personen, Länder, Sportarten, ...
- **Ordinal:** Menge von Werten, auf denen Gleichheit, Ungleichheit und Ordnung ($<$, $>$) definiert sind. Beispiele: Schulnoten, Seriennummern, Likert-Skala, ...
- **Quantitativ/Numerisch:** Menge von Werten, auf denen Gleichheit, Ungleichheit, Ordnung, Differenzen ($-$) und (manchmal) deren Verhältnisse ($-/-$) definiert sind. Beispiele: Geoposition, Timestamp, Temperaturen, ...

Varianten quantitativer Datentypen

- **Diskrete vs. Kontinuierliche:** endliche Werte oder unendliche Werte
- **Intervallskala vs. Verhältnisskala:** Verhältnisse sind nicht sinnvoll zu berechnen und Skala hat keinen Nullpunkt (Intervall); Gegenteil bei Verhältnisskala
- **Lineare Skala vs. zyklische Skala:** Ordnungsrelation ist Totalordnung oder Ordnung muss künstlich definiert werden

\rightarrow einfache Datentypen beziehen sich immer auf eine Datenvariable

Datenvariablen: Eigenschaften von Objekten (Items)

Datenstrukturen: Beschreiben Beziehungen zwischen Datenvariablen und Objekten

\Rightarrow Visualisierung kann Kombi von Datenvariablen, Objekten und Beziehungen abbilden

2.1.1 Datenstrukturen

- **Datentabellen:** allg. Repräsentierung für sehr unterschiedliche Datenstrukturen. Spaltenattribute der Tabelle können Keys oder Values sein, wobei die Datentypen der Keys die

Datenstruktur definieren. Keys sind unabhängige Attribute, und die Values sind von den Keys abhängig. Spalten können entweder Keys oder Values sein.

→ Nur eine Zeile darf die gleiche Kombination von Keys enthalten, wodurch ein Item identifiziert wird. Aber Value Kombinationen können mehrmals vorkommen.

- **Tabelle (ohne Keys):** (uni-, bi-, multi-)variate Daten, wobei Objekte nur durch Zeilennummer unterschieden werden. Typisch wenn Objekteigenschaften wichtiger sind, als Identifikation der Items, aber notwendig bei Anonymisierung. [Univariat = 1 Variable, Bivariat = 2 Variablen, ...]
- **Tabelle (mit Keys):** enthält mind. ein Attribute als Key, welcher Nominal ist. Bei Zeitbezogenen Daten, wird mind. ein Key ein Zeitstempel sein, der absolute oder relative Zeit abbilden kann.
- **Ortsbezogene Daten:** ein Key definiert einen Ort. Hierbei kann es quantitativ sein, und mindestens zwei Keys definieren geographische Länge und Breite, oder es ist nominal, und wird mit dem Ortsnamen definiert.
- **Bewegungsdaten:** Kombination aus Zeitbezogenen und Ortsbezogenen Daten, wobei entweder an festen Orten mehrere Messungen getätigt werden oder der Ort der Messung sich mit der Zeit bewegt.
- **Graphen/Netzwerke:** mind. zwei Keys sind vom gleichen Kategorischen Datentyp, und die Keys bezeichnen eine Kante. Die Tabelle ist hierbei die Kantenliste eines Graphen. → Values sind dann Eigenschaften der Kante, nicht der Knoten
- **Hierarchien/Bäume:** ein Valueattribut enthält einen anderen Key der gleichen Tabelle. Value ist definiert als „ist Kind von“-Relation (bei Knotenliste), oder „ist Parent von“-Relation (bei Kantenliste).

⇒ Tabellendarstellung ist sinnvoll, weil die visuelle Abbildung nicht von Grund auf für jede Struktur definiert werden muss. Die technische Repräsentierung ist hiervon unabhängig.

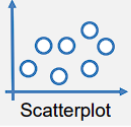

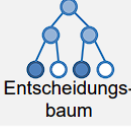
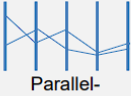
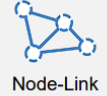
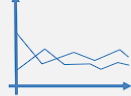


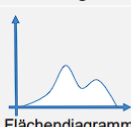
3 Visuelle Strukturen: „Marks“ und „Channels“

⇒ Jede Visualisierung beginnt mit als leeres Blatt. Raum ist die wichtigste und vielseitigste vis. Struktur. Position bezieht ohne weitere Angaben auf zwei unabhängige Variablen.

3.1 Markierungen

geometrische Elemente, aus denen die Visualisierung zusammengesetzt wird, und meistens ein Item oder eine Relation repräsentieren.

→ Markierungen können Punkte, Linien, Flächen oder Text sein. (in der Klausur keinen Text, außer bei Legende, Achse, Caption oder Titel)

Beispiel	Abbildung	Beispiel	Abbildung	Beispiel	Abbildung
 Scatterplot	1 Item → 1 Punkt Metapher: Ort im Raum, Distanz	 Matrix	2 Items → 1 Punkt Metapher: Abstraktion?	 Entscheidungsbaum	Attribut → Punkt Metapher: Abstraktion?
 Parallel-Koordinaten	1 Item → 1 Linie Metapher: Verbindung	 Node-Link Diagramm	2 Items → 1 Linie Metapher: Verbindung	 Liniendiagramm	Attribut → Linie Metapher: Kontinuierliche Änderung, Bewegung
 Balkendiagramm	1 Item → 1 Fläche Metapher: Kategorien, Anhäufung	 Venn Diagramm	N Items → 1 Fläche Metapher: Zusammen gehören, enthalten sein, Kategorien, Menge	 Flächendiagramm	Attribut → Fläche Metapher: Kontinuierliche Änderung, Anhäufung

die Auswahl der passenden Marks hängt ab von ...

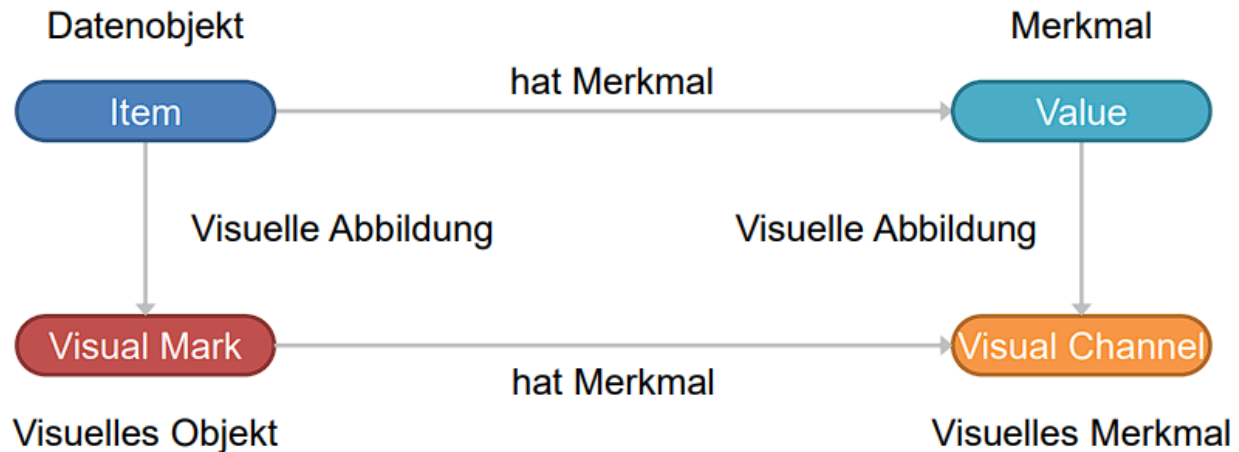
- dem was visuell repräsentiert werden soll (Items, Itempaare, Teilmengen der Items, ein Attribut)
- welche Channels gebraucht werden (Punkte haben zB. keine Länge)

3.2 Channels

⇒ visuelle Eigenschaften von Marks. Welche Channels eingesetzt werden können, hängen von den Marks ab.

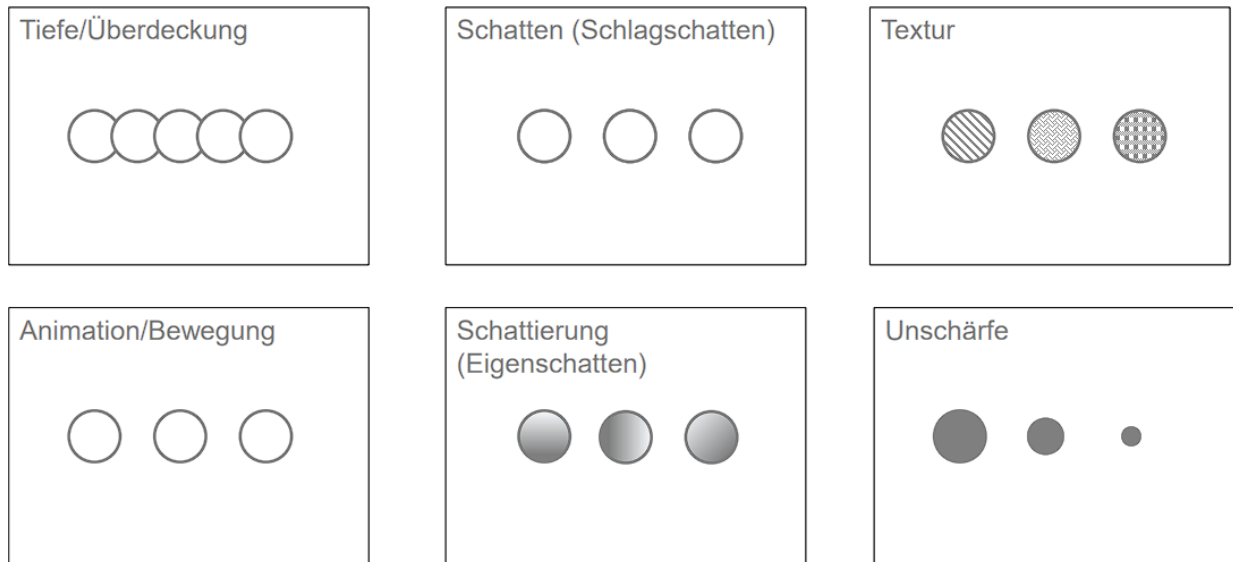
<div> <div>Position</div> <div>Color</div> <div>Brightness</div> <div>Shape</div> <div>Orientation</div> <div>Size</div> <div>Texture</div> </div> <div>Point Channels</div>	<div> <div>Position</div> <div>Color</div> <div>Size/Length</div> <div>Brightness</div> <div>Thickness</div> <div>Dash</div> <div><Dynamics></div> </div> <div>Line Channels</div>	<div> <div>Position</div> <div>Color</div> <div>Brightness</div> <div>Size/Area</div> <div>Boundary</div> <div>Texture</div> <div>Depth</div> </div> <div>Area Channels</div>
--	--	---

⇒ visuelle Abbildung erhält (normalerweise) die Beziehung zwischen Items und Merkmalen



wichtigste Channels und Exoten

- **Position:** einzige notwendige Struktur und einzige die alle visuellen Aufgaben (Suchen, Vergleichen, Ordnen) potentiell gut unterstützt; kann für alle Datentypen genutzt werden; höchste Anzahl unterscheidbarer Werte
→ Wahl der Datenvariable(n) ist die erste und wichtigste Entscheidung
- **Farbkanäle - Farbton, Helligkeit, Sättigung:** vielseitig einsetzbar, aber Helligkeit und Sättigung niemals gleichzeitig für zwei verschiedene Datenvariablen genutzt. Nutzbarkeit ist abhängig von der Farbskala, und neben Position einzige Struktur die mit Pixel funktioniert.
- **Länge:** (mit Position) einziges Attribut, das numerische Größenverhältnisse gut darstellen kann; Falls Visualisierung numerischen Vergleich unterstützen soll → nur Position und Länge!
- **Größe/Flächeninhalt:** naheliegend für qualitativen Vergleich von Größenänderungen; Vergleichbarkeit hängt von Verhältnissen ab; Differenz nicht gut vergleichbar; Nachteil: kostet Platz
- **Form:** Kann gut gelernt werden und potentiell können Visualisierungen viele Formen nutzen; Nachteil: gelerntes sind Konventionen die nicht einfach umdefiniert werden können; Normalfall: wenige neutrale Formen für nominale Datentypen
- **Orientierung:** nur mit bestimmten Markierungen und Formen verwendbar, und repräsentiert ordinale oder numerische Daten.
- **Exoten:**



keine „Regenbogen“ Farbskala für ordinale oder numerische Vergleiche nutzen

3.3 Überblick

Was sollte in einer Visualisierung enthalten sein?

- Titel
- Skala
- Achsenbeschriftung (ggf. mit Einheit)
- Grafikkörper (ohne Text)
- Quelle
- Legende
- Caption (optional; wenn eine gemacht werden muss, wird es in der Aufgabenstellung stehen)

Wie viele Channels sollte eine Visualisierung haben?

Kommt auf das Ziel der Visualisierung an.

- Explain: generell wenige Channels, sodass die Visualisierung verständlich bleibt.
- Explore: eher wenige Channels; es sollte vermieden werden dominante Channels zu nutzen, die die Wahrnehmung von Mustern verhindern.
- Enjoy: Uns überlassen.

Welche Visuelle Channels sind für welche Daten gut geeignet?

	Nominal	Ordinal	Quantitativ	Spatial	Temporal
Position	+	+	+	+	+
Länge	-	+	+	?	?
Größe	-	+	o	-	-
Farbsättigung	-	+	o	-	-
Textur	+	+	-	-	-
Farbton	+	(-)	-	-	-
Orientierung	+	+	-	-	-
Form	+	-	-	-	-

Welche Visuelle Channels sind für welche Aufgaben gut geeignet?

	Gruppierung	Selektion / Hervorheben	Vergleich-Anordnung	Vergleich-Quantitäten	#unterscheidbare Werte (ca.)
Position	+	+	+	+	display size
Länge	-	(+)	+	+	5-15
Größe	-	+	+	-	5-15
Farbsättigung	-	+	+	-	5-7
Textur	+	+	+/-	-	5-7
Farbton	+	+	-	-	7-8
Orientierung	+	+	o	-	4-6
Form	+	o	-	-	5-7 „neutrale“

Mehr dazu in der nächsten Vorlesung

Chapter 3 - Wahrnehmung, Position und Layout

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Grundlagen der Wahrnehmung & Wahrnehmungsmodell von Ware

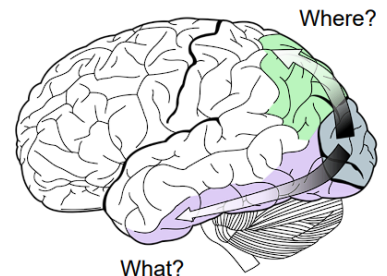
Die visuelle Wahrnehmung beginnt im Auge und endet in den Hirnarealen für räumliche Orientierung, Handlungen und Objekterkennung. Die Wahrnehmung ist hierbei ein mehrstufiger Prozess bei den elementaren Informationen in komplexere transformiert werden. Dieser Prozess wird durch höhere kognitive Prozesse beeinflusst und ist teilweise bewusst steuerbar.

Die Magie passiert in der Netzhaut, wo Lichter durch Lichtrezeptoren aufgefangen werden, und daraufhin durch Kontrasterkennung und Farbverarbeitung durchlaufen. Die Lichtrezeptoren bestehen aus ungefähr 6 Millionen Zapfen und 120 Millionen Stäbchen.

1.1 Sehzentrum

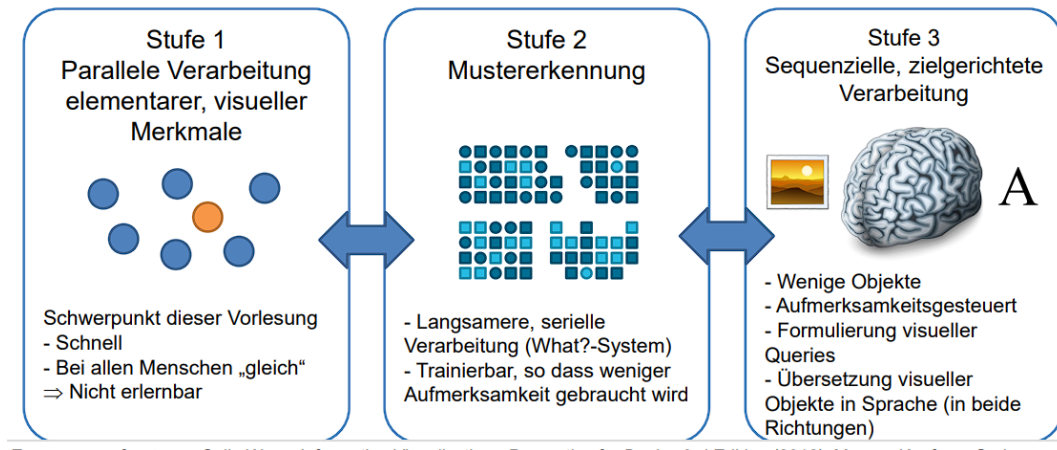
⇒ in 5 Schichten unterteilt, wobei die verarbeiteten Informationen pro Schicht stetig komplexer werden. Schichten sind in beide Richtungen verbunden.

	„What“-System	„Where“-System
Funktion	Erkennung	Lokalisierung
Erregung durch	Details	Bewegung
Speicherung	Längerfristig	Kurzfristig
Geschwindigkeit	Langsam	Schnell
Aufmerksamkeit	Bewußt	Vorbewußt



1.2 Wahrnehmungsmodell von Ware

Wahrnehmung liefert nicht zu jeder Zeit das gleiche Ergebnis, weil es durch Aufmerksamkeit, Erinnerung und andere höhere kognitive Prozesse beeinflusst wird. ⇒ absolut richtiges oder falsches Design, gibt es nur bzgl. gut verstandener, und bei jedem Menschen weitgehend gleich ablaufender Wahrnehmungsprozesse und kognitiver Prozesse.



1.3 Farbwahrnehmung und -modelle

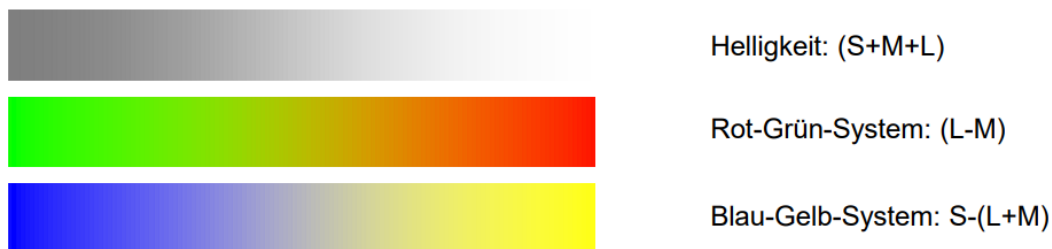
Farbe kann durch drei (fast) unabhängigen Größen definiert werden, wobei die Größe je nach Modell unterschiedlich ist. RGB-Modell hat engen Bezug zu Farbrezeptoren der Netzhaut.

Farbrezeptoren:

- Short-Zapfen: max. Empfindlichkeit bei 419nm (blau) [ca. 10% aller Rezeptoren]
- Medium-Zapfen: max. Empfindlichkeit bei 531nm (grün) [ca. 60% aller Rezeptoren]
- Long-Zapfen: max. Empfindlichkeit bei 556nm (grün-gelb) [ca. 30% aller Rezeptoren]

⇒ aufgrund der Rezeptormengen, können blaue Objekte nicht so scharf gesehen werden, und wir können dadurch mehr Grüntöne unterscheiden.

In der Netzhaut werden die Rezeptorinformationen in Helligkeit, Rot-Grün-System und Blau-Gelb-System umgewandelt.



- andere Farben entstehen aus Kombi RG/BG-System

- Erklärt, dass Helligkeit wird als unabhängig Farbqualität wahrgenommen aber nicht, warum die beiden bunten Kanäle nicht unabhängig wahrgenommen werden.
- "Unbunte" Farben → neutraler Bereich der BG-Skala

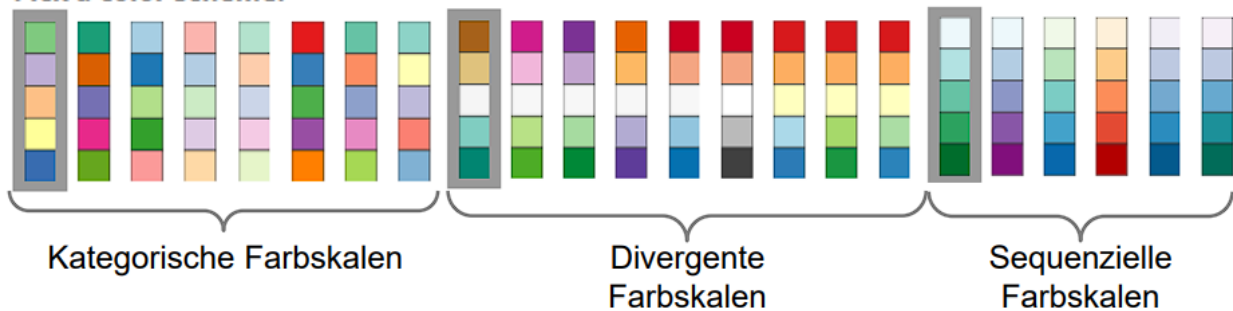
HSV-Farbmodell In der Wahrnehmung und Visualisierung besonders relevant.

- Hue = Farbton
- Value = Helligkeit
- Saturation = Sättigung

Color-Mapping ⇒ Zuweisung einer Skala von Datenwerten auf eine Skala von Farben. Problematisch ist jedoch, dass es viel mehr Möglichkeiten gibt, als bei anderen Channels.

geeignete Farbskalen beschränken Freiheiten auf folgende Weise:

Pick a color scheme:



- **Kategorische Farbskalen (wenige Werte)**: möglichst unterschiedliche Farbtöne und konstante Helligkeit
- **Divergente Farbskalen (auch kontinuierlich)**: zwei unterschiedliche Farbtöne für zwei Skalähälften mit stetiger Variation von Helligkeit und Sättigung. Mitte markiert Nullpunkt.
- **Sequenzielle Farbskalen (auch kontinuierlich)**: Konstanter Farbton (kein Gelb), mit monoton steigender/fallender Helligkeit über die Skala. Richtung der Skala ist konsistent mit der Hintergrundfarbe.

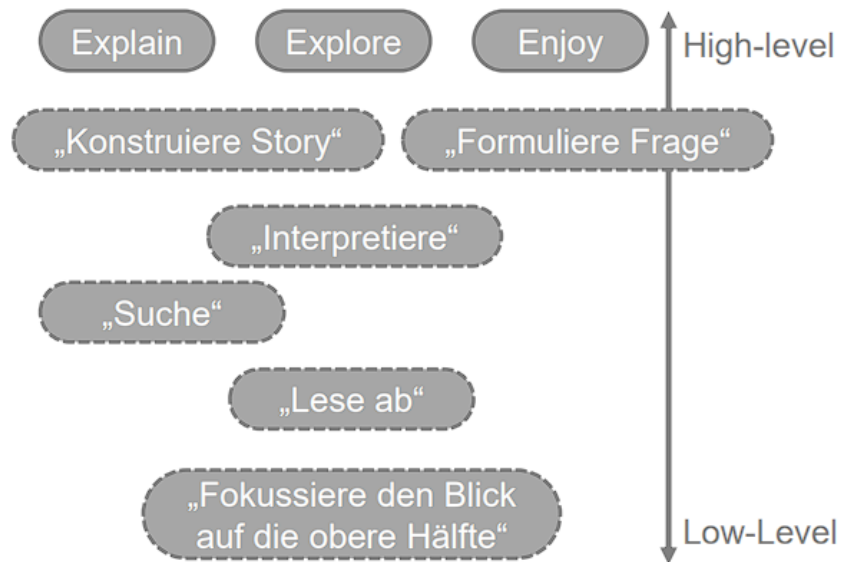
Farbfehlsichtigkeit

Nicht alle Menschen nehmen Farbunterschiede gleich wahr. Rot-Grün-Schwäche (Deuteroanomaly) ist die häufigste Fehlsichtigkeit.

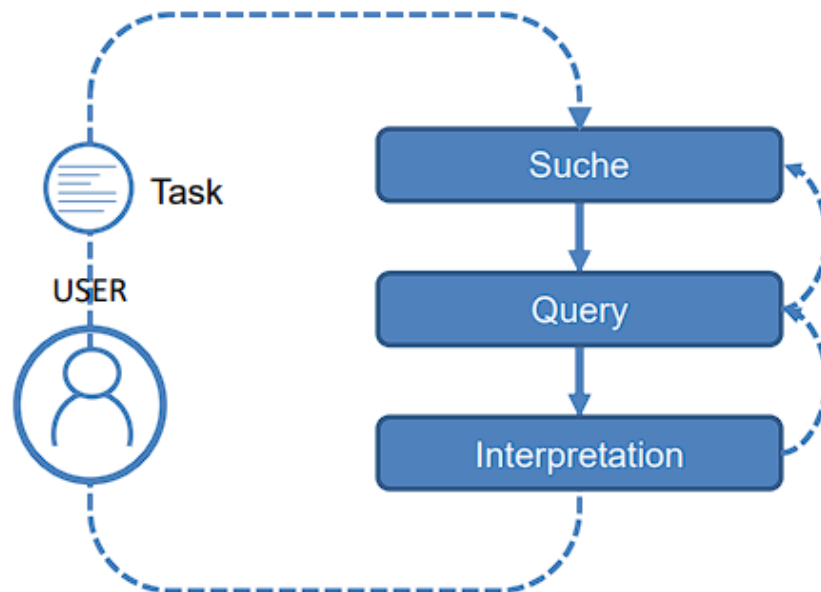
⇒ Vermeidung der Unterscheidung allein durch rot und grün (Ampel). Visualisierung sollte schon als Graubild funktionieren.

2 Elementare visuelle Aufgaben

High-level Aufgaben repräsentieren Ziele und Low-Level Aufgaben beziehen sich auf elementare Handlungen



Was muss ein Mensch mit einer gegebenen Visualisierung immer machen?



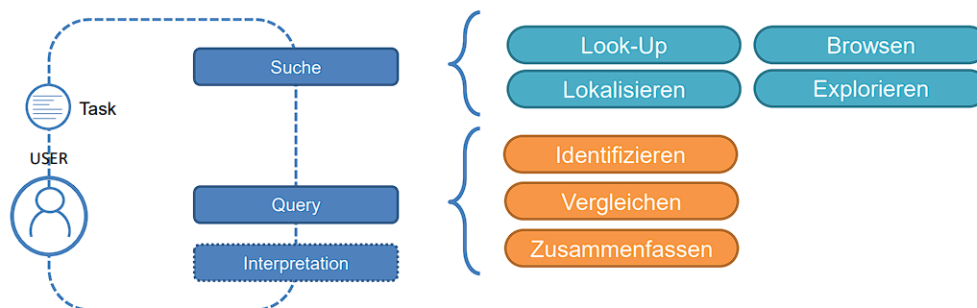
- Suchen: relevante Information finden. Im Idealfall ist klar welche Infos was sind und wo sie sind.
- Query: relevante Information muss gelesen werden. Lesen ist hierbei, die Codierung als visuelle Struktur zurück zu übersetzen.
- Interpretation: gelesene Information im Zusammenhang mit Aufgabe und Nutzerwissen interpretieren. Kann auch unabhängig von Visualisierung geschehen.
⇒ Die drei Schritte werden mehrfach durchlaufen, und im Idealfall unterstützt eine Visualisierung alle Schritte.

2.1 Suche

Es ist hilfreich zu wissen, was man beim Anwender voraussetzen kann. Es gibt vier verschiedene Suchszenarien die jeweils andere Anforderungen haben.

	ich weiß, wonach ich suche		
		Ja	Nein
ich weiß, wo ich das gesucht finden kann	Ja	Look-Up	Browsen
	Nein	Lokalisieren	Explorieren

2.2 Von der Suche zu den Queries



2.2.1 Verschiedene Query Arten

- Identifizieren: Ein Mark gesucht (ID, Name oder Eigenschaften)

- Vergleichen: Unterschiede zwischen mehreren Marks gesucht
- Zusammenfassen: Gemeinsamkeiten vieler Marks gesucht

⇒ visuelle Aufgaben lassen sich als Kombination von Suchen und Queries beschreiben. Diese Bildern die ersten Schritte beim Lösen einer komplexeren Aufgabe.

3 Wahrnehmung und Eigenschaften der visuellen Channels

- Eigenschaften visueller Channels können getestet werden.

3.1 Auswahl/Hervorhebung

Selektive visuelle Channels helfen bei Lokalisierung.

- Wahrnehmung der Hervorhebung abhängig vom Kontrast.
- Channel für hervorhebung nicht für Eigenschaftsdarstellung nutzbar
- Nutzung mehrerer visueller Channels zur Hervorhebung verschiedener Markierungen nicht sinnvoll

3.2 Ordnung

Ordinale visuelle Channels helfen beim ordinalen Vergleich (größer, kleiner, ...). Die Ausprägung ist hierbei natürlich geordnet, sodass die Channels nicht von einer Skala gelesen werden müssen. Sonderfall: Position erleichtert jede Suche als Sortierung. Die Orientierung kann für zyklische Ordnungen oder lineare genutzt werden. (Uhr \Leftrightarrow Tacho)

3.3 Differenzen

Quantitative visuelle Channels helfen beim quantitativen Vergleich. Hierbei können Unterschiede von Channelausprägungen verglichen werden. Länge und Position sind die einzigen Channels bei denen das sicher so ist.

Voraussetzungen

- Skalen sind linear (logarithmische Skalen nur für qualitative Vergleiche)
- Skalen haben Nullpunkt

3.4 Zusammenfassen

Assoziative visuelle Channels helfen dabei, ähnliche Items als Gruppen wahrzunehmen. Ähnlichkeit kann durch Abbildung von kategorischen Datenvariablen vorgegeben werden oder eine kombinierte Wahrnehmung mehrerer Channels sein (Explain or Explore).

4 Einflussfaktoren

Unterschiedswahrnehmung ist in fast allen Channels von dem Verhältnis zwischen Kontrast und Entfernung abhängig. (Nicht nur Farbkontrast)

Man sollte deswegen priorisieren welche Unterschiede wichtig sind und unbedingt wahrgenommen werden sollen.

perzeptuelle Länge: Anzahl unterscheidbarer Werte auf einem Channel.

Wenn Channels unabhängig voneinander unterschieden werden können, sind sie **separabel**.

Wenn mehr unterscheidbare Werte für eine Datenvariable erlaubt wird, sind sie **integrierende** Paare.

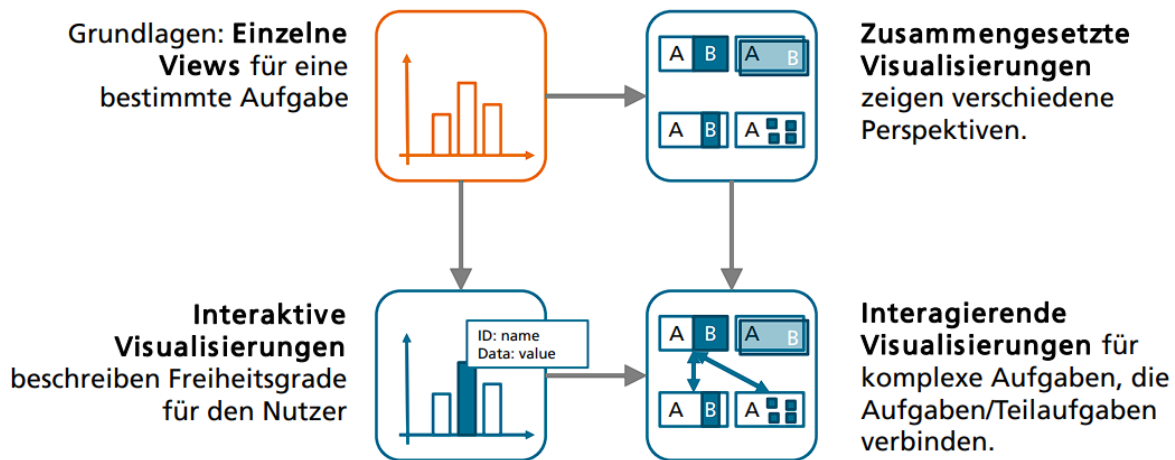
5 Position, Layout und Komposition

Jede Visualisierung gibt dem Raum eine Struktur. Sobald diese vertraut ist, wird die Suche erleichtert. Eine einmal gelesene Struktur, kann auf benachbarte Strukturen angewendet werden.

Beispiel:

- Gemeinsame Achsen (Position) \Leftrightarrow Sie wissen, wonach Sie suchen müssen
- Gemeinsame Legenden (andere Channels) \Leftrightarrow Sie wissen, wo Sie suchen müssen

5.1 Komplexe Visualisierungen



5.2 View Design Muster

Juxtaposition (Gegenüberstellung)

- Häufigstes Designmuster
- Views stehen nebeneinander
- Verbindung zwischen Visualisierung implizit (z.B. über eine gemeinsame Achse) oder explizit (über Verbindungslinien)

Superimposition (Überlagerung)

- Zwei Views nutzen gleichen Raum
- Stellt einen räumlichen Bezug zwischen Views her
- Eine oder zwei gemeinsame Achsen oder gemeinsame Markierungen

Overloading (Überladung)

- Eine ergänzende Visualisierung wird in einer Hauptvisualisierung dargestellt
- Visual Mapping auf Position neu oder von Hauptvisualisierung modifiziert
- Nutzung anderer Marks, Channels

Nesting (Einbettung)

- Visual Marks einer Visualisierung sind selbst (kleine) Visualisierungen
- Stellt einen Bezug zwischen „Überblick“ und „Details“ über die Daten her.

Quelle: Javed & Elmqvist; Exploring the Design Space of Composite Visualization; IEEE Pacific Visualization Symposium; 2012

Chapter 5 - Datenvorverarbeitung

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Datenvorverarbeitung

Das Überbringen von Rohdate zu Datentabellen mithilfe der Data Transformation.

Warum Datenvorverarbeitung?

- unsaubere rohdaten (unvollständig, vertauscht, inkonsistenz, ...)
- Garbage in, Garbage out (GIGO): akkurate Datenbasis ist sehr wichtig, weil sonst nur Müll rauskommt

Wo beginnt Datenvorverarbeitung?: Umfasst die Datentransformationen, die durch den Entwickler im Vorfeld der Nutzung erfolgen (Anpassungen, Bereinigungen, Umstrukturierung, ...)

Unterschied Datenvorverarbeitung und Datentransformation: Ob etwas Vorverarbeitung oder Transformation ist hängt von der Domäne, dem Nutzer oder der Aufgabe ab. Bsp: fehlende Werte werden durch den Entwickler geregelt; Werte die unrealistisch wirken aber der Entwickler nicht validieren kann, müssen Nutzerbedingt geregelt werde.

Wie viel Zeit verbraucht ein Designer für die Datenvorverarbeitung oder der Visualisierung selbst?: 80% der Zeit beschäftigen sich Data Scientists mit Datenvorverarbeitung. Der Rest ist Visual Mapping, etc.

1.1 Methoden der Datenvorverarbeitung

1.1.1 Metadaten und Statistik

Metadaten: hilfreich wenn man sich mit unbekannten Datenquellen beschäftigt, und es sind Daten über Daten (zB Attributnamen, Referenzpunkte, Einheiten, wichtige Symbole/Schlüsselwörter für fehlende Werte)

Statische Analyse: Erkennung von Ausreißern, Clusteranalyse, Korrelationsanalyse, Statische Plots, Histogramme, ...

1.1.2 Fehlende Werte und Datenbereinigung

- Ignorieren: aber manche Visualisierungstechniken funktionieren nicht mit fehlenden Werten
- Manuell Einfügen: Expertenwissen, präzise aber oft Teuer und Aufwendig
- Eliminieren des gesamten Datensatzes: Häufigste Methode - bei meisten Quellen fehlt bei Mehrzahl der Datensätze mind. ein Wert
- Globale Konstante: verändert die Datenverteilung, Vorsicht bei statistischer Analyse
- Mittelwert benutzen: gut für globale Statistik, wobei die individuelle Abweichung groß sein kann
- Wert basierend auf Ähnlichkeit bei anderen Werten: Annahmen benötigt, auf welchen Attributen die Ähnlichkeit beruht

Welche Methode wird gewählt?: Abhängig nach Typ, Semantik, Menge der fehlenden Werte und Expertise des Anwenders. Wichtig: bei Ersatzwerten speichern, dass es Ersatzwerte sind! Fehlerhafte Werte sind schwer zu beheben. Diese sind oft durch Menschen verursacht und sind schwer zu erkennen.

Ausreißerdetektion: Ein Ausreißer ist ein Datenpunkt außerhalb des normalen Datenspektrums. Starke Ausreißer sind Anomalien, die für Experten jedoch interessant sind. Das Erkennen von Ausreißern kann durch Visualisierung und Interaktion erfolgen oder durch Statistische Datenauswertung. Dann können statistisch unwahrscheinliche Werte ausgegrenzt werden. Man nimmt hierbei an, dass eine Normale Datenverteilung vorhanden ist. Mehrere Datenteile werden durch Clustering gefunden.

⇒ Nicht jeder Ausreißer ist ein Fehler!

1.1.3 Normalisierung und Skalierung (wichtig für Klausur)

Overplotting: Zu viele Plots in einer Stelle, die es erschwert die Datensätze zu erkennen. Dieses Problem kann durch Skalierung gelöst werden.

Skalierungs-/Normalisierungsmethoden

- min-max-Normalisierung: Man betrachtet Minimal und Maximal Wert.

- Linear: $f_{\text{lin}}(v) = \frac{v - \min}{\max - \min}$
- Logarithmisch: $f_{\text{log}}(v) = \frac{\ln(v) - \ln(\min)}{\ln(\max) - \ln(\min)}$
- Wurzel: $f_{\text{sq}}(v) = \left(\frac{v - \min}{\max - \min}\right)^2$
- Quadratisch: $f_{\text{sqrt}}(v) = \sqrt{\frac{v - \min}{\max - \min}}$

– Exponentiell:

- weitere Datenspezifische Normalisierungen:
- z-Score:
- Quantil-Normalisierung:

Wie wählt man die geeignete Normalisierung aus?: Auswahl erfolgt basierend darauf, welche Bereiche besonders starkes Overplotting haben.

Ein Problem bei Normalisierung ist die Verschiebung des Neutralwertes: Wird verursacht wenn man sehr unterschiedliche Min-Max-Werte hat. Um das Problem zu lösen kann man den min umwandeln sodass dieser entweder der ursprüngliche Min-Wert ist oder der negative Max-Wert (je nachdem welcher kleiner ist) und analog zum max.

Lokale Skalierung: Min-Max bei nur einem Datensatz; (gut für Vergleich von Wertverläufen aber nicht für direkten Wertevergleich)

Globale Skalierung: Min-Max bei allen Datensätzen; (Gut um Werte zu vergleichen, aber wenig geeignet bei großen Unterschieden)

1.1.4 Diskretisierung

⇒ Vorverarbeitung kontinuierlicher Datenmengen/Dimensionen. Hierbei wird eine diskrete Teilmenge als Approximation extrahiert (mit Genauigkeitsverlust). Häufige kontinuierliche Dimension ist die Zeit.

1.1.5 Sampling, Segmentierung und Untermengen

- Sampling: Methoden zur Reduktion von Datenmengen; häufig zufällige Auswahlverfahren für repräsentative Stichproben
- Segmentierung: Einteilung der Daten in zusammenhängende Abschnitte, die jeweils zu einer Kategorie gehören. Segmentierung ist nicht immer (eindeutig) möglich
- Untermengen: Reduktion der Datenmenge auf Untermenge. Erstellt durch Filter und andere Einschränkungen

1.1.6 Datenintegration

⇒ Fusion verschiedener Datenquellen. Hierbei wird das Schema, die Qualität und weiteres angeglichen

1.1.7 Dimensionsreduktion

siehe Vorherige Vorlesung

1.1.8 Abbildung von nominalen Werten auf Zahlen

1.1.9 Aggregation und Summenbildung

1.1.10 Glätten und Filtern

Chapter 4 - Interaktion

Informationsvisualisierung und Visual Analytics
WiSe 2024/25

1 Interaktion

1.1 Wozu Interaktion?

- Interaktion ermöglichen um mehrere Fragestellungen abzudecken
- Mehr Daten können durch Interaktion gezeigt werden
- Einblicke schaffen, die der Mensch / das System nicht alleine schaffen

TOWATCH = Hans Rosling TH ebest stats you ve ever seen TEd Talk Design of Everyady THings
- Don Norman

1.2 Bedienung und Interaktion nach D.Norman

- Entscheiden: Was ist zu tun?
- Formulieren: Was möchte ich (Nutzer) tun?
- Spezifizieren: Welche Schritte muss ich durchführen um mein Ziel voraussichtlich zu erreichen?
- Ausführen: Spezifizierte Handlung durchführen
- Wahrnehmen: Nutzer nimmt nun Umgebung NACH seiner Aktion wahr.
- Interpretieren: Interpretieren der Wahrnehmung. Was ist passiert, und warum? Was haben Folgeaktionen zu bedeuten.
- Vergleichen: Habe ich (Nutzer) mein Ziel erreicht?

Interaktion ist technisch die Möglichkeit, die Transformationen und das Visual Mapping zu steuern (siehe Card-Model)

Benutzungsschnittstelle: eine Handlung oder eine Stelle, mit der ein Mensch mit einer Maschine in Kontakt tritt.

1.3 Interaktionsmodi nach Spence

- Kontinuierliche Interaktion: kontinuierliche Veränderungen in der Visualisierung und performante Reaktion auf Nutzeraktionen

- Schrittweise Interaktion: entlang verschiedener Schritte im Rahmen der Aufgabenerfüllung und Entscheidungsfindung. (Sensitivity = Signalwahrnehmung in Umgebung/Darstellung)
- Passive Interaktion: Statische Darstellungen, Browsing, Bewegung die nicht vom Nutzer direkt beeinflusst wird
- Gemischte Interaktion: Kombination von sensorischen und motorischen Interaktionen in einem System beschreibt, um ein möglichst immersives und natürliches Erlebnis zu schaffen.
- Interaktionsdynamik (spätere Vorlesung)

1.4 Gewünschte Antwortzeit des Systems

⇒ Je nach Interaktionstypen, sollten verschiedene Antwortzeiten gewährleistet sein.

- Animation, fließende Bewegung: 0,1s (10 FPS)
- Reaktion des Systems auf Benutzeraktionen (1s)
- akzeptable Responsetime auf komplizierte Anfragen (10s[5s-30s])
- Visuelle Ladeanzeigen sollten idealerweise weniger als 1s angezeigt werden
- Progressive Visualisierung / Analytics (VA-Teil)

1.5 Interaktionstechnik

⇒ Interaktion ist die Kommunikation zwischen dem Benutzer und dem System. Eine Interaktionstechnik bezeichnet die Nutzung eines physischen I/O Geräts um eine generelle Aufgabe durchzuführen. In der Visualisierung ist es jedoch die Möglichkeiten des Nutzers, direkt/indirekt Datenrepräsentationen zu manipulieren.

1.5.1 Systemnahe Interaktionstechniken

- Selektion: Identifikation eines bestimmten Objekts durch Definieren einer Teilmenge.
⇒ Fitt's Law: Selektionszeit = $a + b * \log_2(\frac{D}{W} + 1.0)$
 - D: Distanz zum Zentrum des Ziels
 - W: Größe/Ausdehnung des Ziels
 - a, b: Empirisch determinierte Konstanten in ms
- Navigation: Tastatur, Maus, Space Maus, Trackingsysteme, ...
- Zooms (Geometrisch/Semantisch): Geometrisch = Seite wird näher ins Auge bewegt; Semantisch = zusätzliche Information bei näherer Ansicht
⇒ Shneidermans Mantra

- Überblick über alle Daten
- Zoom und Filter
- Details auf Anfrage

→ Bei zu vielen Datenpunkten nicht vorteilhaft

- Fokus + Kontext: Variation perspektivischer Transformation. Fish Eye oder Magic Lens.
- Überblick + Detail: Hierarchische Kopplung mehrerer Visualisierung (Bsp.: Street View mit Mini Map). Potenzielle Nachteile sind hierbei Verdeckung und Korrespondenz.
- Brushing + Linking: Auswahl von Elementen (Brushing) in einer View, um korrespondierende Daten (Linking) in ANDEREN Views hervorzuheben.

1.6 Kategorien der Interaktion nach Yi et al.

- Selektion: Interessantes markieren
- Exploration: Zeig was anderes
- Rekonfiguration: Zeige andere Zusammenstellung
- Encodierung: Zeige andere Repräsentierung
- Abstrahieren/Spezialisieren: Zeig mir mehr oder weniger Details
- Filtern: Zeig mir was unter bestimmten Bedingungen
- Bezug: Zeige Beziehung zwischen Elementen

2 Design der Interaktion

Good read: Designing the UI Ben Shneiderman & Catherine Plaisant

2.1 Leitsätze des Interaktionsdesigns

- Navigation
 - Standardisierung von Arbeitsabläufen
 - klare Zielbeschreibung bei Links
 - eindeutige Überschriften
 - Radiobuttons für ausschließliche Auswahl
 - Seitenentwurf, die sich drucken lassen
 - Thumbnailnutzung als Vorschau
- Organisation der Anzeige

- Konsistenz
- effiziente Informationsaufnahme durch Nutzer (Erwartungskonformität)
- Flexibel und Individualisierbar
- Anzeige nur von hilfreichen Informationen
- Grafische Darstellung anstelle von Text/Zahlen falls möglich
- Nutzer bei Design involvieren (Ustests)
- Erzeugung von Aufmerksamkeit
 - intensive Farben für wichtige Aspekte
 - Markierung und Größe
 - Max. 3 Fonts
 - Max. 4 Standardfarben
 - Vorsicht mit blinkenden Elementen (2-4 Hz)
 - Weniger ist mehr
 - Audio: weiche Töne (+), hasche Töne (-)
- Unterstützung der Dateneingabe
 - Dateneingabe-Transaktionskonsistenz
 - Inputaktionsminimierung
 - keine Codeeingaben
 - Dateneingabeflexibilität

2.2 Prinzipien des Interaktionsdesigns

- fachliches Nutzerniveau ermitteln
 - Anfänger und Erstnutzer benötigen Hilfedialoge
 - Sachkundige, gelegentliche Nutzer benötigen konsistente Abläufe, wiederkehrende Lösungsmuster, sinnvolle Meldungen
 - Experten verlangen schnelle Antwortzeiten, Feedback im Hintergrund, Shortcuts, Abkürzungen oder andere Beschleunigte Dialoge
- Arbeitsaufgaben ermitteln
 - Taskhäufigkeit wichtiger Maßstab. So bekommen häufige Aufgaben Tastenkombis, weniger häufige bekommen Auswahl in der Menüleiste und seltene Aufgaben mehrere Menüselektionen oder Formularausfüllung.
- Interaktionsstil wählen
 - ⇒ Direkte Manipulation
 - Kommandozeile
 - Eingabeformular

- Menüauswahl
- Direkte Manipulation
- Spracherkennung
- 8 goldene Regeln der Gestaltung
 - Konsistenz, wo immer möglich
 - Möglichst universelle Benutzbarkeit (Accessibility)
 - Informatives Feedback für jede Benutzeraktion
 - Design von Dialogen, die eine Gruppe von Aktion zum Abschluss führen
 - Fehlerverhinderung
 - Einfaches reversen von Aktionen
 - Unterstützung des Kontrollgefühls
 - Reduzierung der Gedächtnisbelastung

2.2.1 Menschliche Reaktionszeit

- Hick-Hyman-Gesetz: $\text{Reaktionszeit} = a + b * \log_2(C)$
- C: Anzahl der Auswahlmöglichkeiten (Choices)
- a, b: Empirische Konstanten
- Fehlertoleranz: schnellere Antwortzeit, wenn Menschen gelegentlich Fehler machen dürfen

Chapter 6 - Techniken I

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Visualisierungstechniken für mehrdimensionale quantitative Daten

⇒ fast alle Techniken sind ungeeignet um einzelne Datenwerte abzulesen. Ziel ist es Werteverteilungen sichtbar zu machen, um Muster oder Ausreißer zu finden.

1.1 Scatterplot Matrix

Bei einer Scatterplot Matrix handelt es sich um eine genestete Ansicht von mehreren Visualisierungen. Die Attributnamen werden auf Spalten und Zeilen abgebildet, während die Werte auf die Koordinaten innerhalb der Spalten und Zeilen abbildet. Alle Dimensionen sind hierbei gleichbereitet.

1.1.1 Stärken

- geeignet für 4-8 Dimensionen
- Bietet eine Übersicht über 2D-Korrelationen
- komplexere Abhängigkeiten können durch Brushing und Linking realisiert werden
- Spalten- und Zeilenanordnung sind weitgehend unabhängig
- Wahrnehmung kaum verzerrt

1.1.2 Schwächen

- skaliert schlecht bei vielen Datenpunkte
- nur paarweise Relationen werden gezeigt

1.2 Starplot

Auch Netzdiagramm, Radar-Charts oder Spider-Chart genannt. Jede Richtung definiert hierbei eine Skala, und die 360° werden gleichmäßig verteilt. Auf jeder Skala wird ein Datenwert abgetragen. Und diese Datenwerte werden als Punkte dargestellt und verbunden. Korrelationen benachbarter Achsen sind dabei gut sichtbar. Es kann sinnvoll sein Flächen statt Linien zu verwenden, wenn der Fokus auf der Beurteilung der Ähnlichkeit liegen soll.

Wenn die Achsenbeschriftungen reduziert werden, ermöglicht es den Fokus auf die Form zu legen, statt der einzelnen Werte.

Small Multiples: das Anzeigen mehrerer Starplots, die nebeneinander platziert werden. (weniger Overplotting, aber schwerer Vergleich)

1.2.1 Stärken

- geeignet für 4-15 Dimensionen
- Ähnlichkeit über die Formwahrnehmung

1.2.2 Schwächen

- nur für wenige Daten geeignet
- abhängig von der Achsenanordnung

1.3 Parallele Koordinaten

Hier werden die Daten ähnlich wie bei den Starplots dargestellt, jedoch sind die Achsen Parallel. Der Output ist jedoch KEIN Liniendiagramm. Parallele Koordinaten stellen eine populäre Technik dar.

1.3.1 Stärken

- Layout unterstützt beliebig viele Achsen
- einfache Korrelationen gut erkennbar

1.3.2 Schwächen

- Overlotting, da Linienzüge nicht verfolgt werden können
- Achsenordnung beeinflusst die Wahrnehmung
- Achsenrichtung beeinflusst die Wahrnehmung
- Korrelationen benachbarter Werte sind direkt sichtbar, aber keine anderen Korrelationen

1.3.3 Best Practice

Die Schwächen der Parallelen Koordinaten können mithilfe der Interaktion verringert werden.

- Datenkategorienfiltern

- Rangefilter pro Achse
- Achsen umsortierbar/invertierbar

1.4 RadViz

Bei RadViz sit ein radiales Layout wie bei Starplot vorhanden, jedoch werden hohe Werte entlang der Achse bis zum Achsenende "gezogen". RadViz ist hierbei eine Projektionstechnik. Um die Darstellung zu erstellen, werden die gewichtete Summe der Achsenvektoren der Datenwerte berechnet, und vorher min-max normiert.

1.4.1 Stärken

- Es spart Platz
- die relativen Datenverhältnisse sind gut sichtbar

1.4.2 Schwächen

- Achsenanordnung bestimmt die Position, und redundante Achsen verzerren das Ergebnis.
- Die Normierung der Datenwerte sorgt dafür, dass die Punktpositionen nicht eindeutig sind. Heißt $(0, 0, 0)$ hat die selbe Position wie $(1, 1, 1)$, ...

2 Visualisierungstechniken für mehrdimensionale kategorische Daten

2.1 Parallel Sets

Eine Ansicht für kategorische Variablen, wobei eine wählbare Variable zusätzlich die Farbe bestimmt. Die Achsen werden parallel angeordnet, und Kategorien entlang einer Achse platziert. Die Anzahl der Items einer Kategorie bestimmt die Breite und die Breite sowie Reihenfolge bestimmt die Position. Wenn Items in zwei gemeinsamen Kategorien benachbarter Achsen sind, werden diese als Trapez dargestellt. Ein Attribut bestimmt dabei die Farbe der Plots.

2.1.1 Stärken

- skaliert für beliebig viele Daten
- geeignet für 20+ Dimensionen

2.1.2 Schwächen

- Abhängig von der Achsenanordnung und Kategoriereihenfolge.
- Häufige Kategorien dominieren seltene und es ist abhängig davon, in welcher Reihenfolge die Kategorien gezeichnet werden (Overplotting)

2.1.3 Best Practice mit Interaktion

Folgende Dinge können durch deren Interaktion die Schwächen aufheben.

- Achsenreihenfolge
- Kategorienreihenfolge
- Highlighting interessanter Teilmengen
- Auswahl des Attributes, dass die Farbe bestimmt

2.2 Mosaic-Plot

Ein Mosaic-Plot basiert auf rekursiven Teilung von Flächen. Bei jeder Teilung wird ein anderes Attribut gewählt und die Richtung gewechselt. Die Teilung entspricht dabei der Häufigkeit der Items. Die Teilung kann theoretisch bis zur Bildschirmauflösung getrieben werden.

2.2.1 Stärken

- effiziente Nutzung des Raums (space-filling)
- skaliert für beliebig viele Daten
- Abhängigkeit des 2. vom 1. Attributs ist gut sichtbar
- Abhängigkeiten zwischen Variablen und deren Elternvariablen sind einigermaßen gut darstellbar

2.2.2 Schwächen

- Bei mehrere Variablen schwer lesbar
- schlecht vergleichbar
- Nullwerte müssen gesondert behandelt werden

2.3 Karnaugh-Veitch Map

Eine Map die bei mehr als 20 nominalen Attributen verwendet werden kann. Hierbei wird ebenfalls eine rekursive Unterteilung erstellt, wobei die Unterteilung immer gleichmäßig ist und von der Kategorienhäufigkeit nicht abhängt. Die Häufigkeit wird auf die Farbe abgebildet.

Die Rekursive Unterteilung beginnt mit einem Attribut und wird äquidistant unterteilt. Eine KVMMap mit zwei Attributen ist eine Matrix, in den Zellen der Matrix werden die Anzahl der Item gezählt. Die Farben zeigen dann ob die beiden Attribute statistisch unabhängig sind.

2.3.1 Stärken

- Geeignet für bis zu ca. 10 Attribute
- zeigt Abhängigkeiten zwischen allen Attributen an
- Frequenzen und Wiederholungen können wahrgenommen werden

2.3.2 Schwächen

- Wahrnehmung der Muster muss gelernt werden
- Muster sind nicht interpretierbar. Hierbei werden automatische Verfahren verwendet.

2.4 Tabellarische Visualisierung

Werte werden in einer Tabellarischen Ansicht graphisch dargestellt. Hierbei können die Werte einer Zelle durch die Farbe oder einer Balkenlänge dargestellt werden.

2.4.1 Stärken

- Tabellen sind vertraut und können für 20+ Attribute genutzt werden (Heatmap)
- Im Extremfall kann ein Datensatz pro Pixelzeile verwendet werden
- Attribute können Nominal oder Quantitativ sein
- Die Sortierbarkeit zeigt die Korrelation über viele Attribute

2.4.2 Schwächen

- limitiert auf mehrere Hundert Datensätze

Name	Datentypen der Dimensionen	#Dimensionen	#Datensätze	Anordnung ohne Einfluss?	Lesbarkeit [mit Interaktion]	Aufgabe
Scatterplot-matrix	Quantitativ	4-8...	100-10000	++	Gut [Sehr Gut]	Paarweise Korrelationen finden und klassifizieren
Starplot	Quantitativ [Nominal*]	4-15...	10-100	---	Mittel [Gut]	Vergleich in vielen Dimensionen
Parallele Koordinaten	Quantitativ [Nominal*]	Ca. 30 (Bildbreite)	100-1000	--	Schlecht [Gut bis Sehr Gut]	Paarweise Korrelationen finden, Suche in nD
RadViz	Quantitativ	4-15	10-1000	---	Schlecht [Eher Schlecht]	Lineare Abhängigkeiten finden
Parallel Sets	Nominal [Quantitativ*]	Ca. 30 (Bildbreite)	„unendlich“	--	Mittel [Gut bis Sehr gut]	Paarweise Korrelationen , finden, Suche in nD
Mosaic Plot	Nominal [Quantitativ*]	4-6	„unendlich“	--	Schlecht [Mittel]	Häufigkeiten vergleichen, Paarweise Abhängigkeiten
KVMap	Nominal [Quantitativ*]	4-10	„unendlich“	-	Extrem Schlecht [Mittel → VA]	Korrelationen in nD finden, Muster in nD finden
Tabellen	alle	30+	100-1000	+	Gut [Sehr Gut]	Vergleichen, Korrelationen finden

3 Strategien für Darstellung vieler Items und Attribute

3.1 Ordnen-Methoden

Ordnung ist von Relevanz, weil wichtige Daten stehen dort wo man sie sucht, und ähnliche Daten nahe beieinander. Die Frage dabei ist jedoch, wie man Ordnung, Wichtig und Ähnlich bei 1000 Attributen definiert.

- Dimensionsreduktion: Wie fasse ich Attribute zusammen?
- Feature-Selektion: Welche Attribute sind wichtig?
- Sortieren: Welche Items sind wichtig?

3.2 Grundproblem bei Dimensionsreduktion und Feature Selektion

Der Bildschirm hat nur zwei Dimensionen und mehrere Dimensionen sind nur bedingt wahrnehmbar.

Je mehr Eigenschaften für zwei Items bekannt sind, desto wahrscheinlicher unterscheiden sie sich auch in irgendwas \Rightarrow Curse of Dimensionality. Mehrere Dimensionen sorgen dabei auch für langsamere Berechnungen und redundante Dimensionen verzerren die Ergebnisse.

3.3 Definition

Gegeben seien Datensätze X mit $n > 2$ Attributen: $x = (x_1, x_2, x_3, \dots, x_n) \in X$ und eine Funktion $dist_n$ die einen Abstand zwischen Datensätzen messen kann.

Eine Dimensionsreduktion sucht allgemein eine Abbildung $red : X \rightarrow \mathbb{R}^2$, so dass gilt für alle Paare von Datensätzen: $dist_n(x, y) \equiv dist_2(red(x), red(y))$. $dist_2$ ist der euklidische Abstand, aber $dist_n$ nicht unbedingt.

Die Abbildung *red* erhält Unterschiede und Gemeinsamkeiten in den Daten. Ein großer Abstand vor der Reduktion soll einem großem Abstand nach der Reduktion entsprechen.

Eine Feature Selektion wählt aus diesen n Attributen m relevante Attribute aus mit $m < n$. Allgemein kann Relevanz so definiert werden: Wenn man vor der Selektion mit den Vektoren x eine Aussage über eine weitere Itemeigenschaft machen kann, dann soll die gleiche Aussage auch mit s möglich sein. Relevanz ist also abhängig von der Frage.


3.4 Schwierigkeiten von Dimensionsreduktion

Bei weniger Dimensionen hat man weniger Optionen um Unterschiede zu beschreiben.

3.5 Verfahren für Dimensionsreduktion

- PCA - Principal Component Analysis:
 - Gut für numerische Daten mit linearer Abhängigkeit, andere Datensätze benötigen andere Methoden. Die PCA basiert auf einer Varianzmaximierung und als Standardmethode in vielen Libraries enthalten.
 - Um PCA durchzuführen müssen Achsen im Ursprungsraum orthogonal zueinander stehen. Neue Dimensionen sind dann Linearkombinationen aller anderen Achsen. Eine Achse geht entlang der größten Varianz, die zweite entlang der zweitgrößten, ...
 - Hauptkomponenten sind dann orthogonal und bei der Visualisierung werden idR die ersten Hauptkomponenten genutzt. Der Informationsverlust, der pro fehlender Hauptkomponente auftritt, kann berechnet werden.
 - LDA - Linear Discriminant Analysis (Skipped)
 - MDS - Multidimensional Scaling and Sammons Mapping
 - Bei MDS werden Daten im niedrigdimensionalen Raum so geordnet, sodass ihre Distanzen aus dem Originalraum im Zielraum gut abgebildet sind. Man versucht also die Distanzverzerrung zu minimieren. Dies funktioniert auch für Daten, von denen nur die Entfernung bekannt ist. Eine globale nicht-lineare Transformation zu finden ist ein Optimierungsproblem.
 - t-SNE - t-distributed stochastic neighbor embedding
 - SOM - Self-organizing Maps
 - eine SOM ist ein lernendes Neuronales Netz, das sich an Datentopologie anpasst. Benachbarte Knoten sind ähnlich und es findet eine nicht-lineare Transformation statt.
1. Jede Zelle wird mit zufälliger Zeitserie initialisiert
 2. Jede Zeitserie, wird in jene Zelle mit jeweils ähnlichsten Prototyp zugeordnet

3. Für jede Zelle wird Mittelwert bestimmt. Die Prototypen in einer Zelle und ihre direkten Nachbarn werden zum Mittelwert hingezogen. Die fernen Nachbarn werden verändert von diesem Mittelwert weggezogen.
4. Schritt 2-3 wiederholen, bis sich nichts mehr ändert.

Verfahren	Besonderheiten/Stärken	Schwächen
PCA	Lineares Verfahren. Erhält die Varianz der Datenverteilung im Originalraum. Einfach, geschlossene Lösung.	Kann nicht-lineare Zusammenhänge nicht abbilden. 
LDA	Lineares Verfahren, benötigt gelabelte Daten, Sucht Achsen, die Labels möglichst gut trennen	Kann nicht-lineare Zusammenhänge nicht abbilden.
MDS	Nicht-Lineares Verfahren, Benötigt nur Distanzen, keine Punkte.	Lösung wird nur approximiert.
SOM	Nicht-Lineares Verfahren. Auch für die Gruppierung der Daten geeignet.	Lösung wird nur approximiert. Relativ viele Parameter

In keinem der Verfahren haben die neu erzeugten Achsen eine Bedeutung, außer dass sie die originalen Achsen kombinieren.

Chapter 7 - Techniken II

Informationsvisualisierung und Visual Analytics
WiSe 2024/25

1 Visualisierung Zeitbasierter Daten

- Zeit ist keine reine numerische Variable und wird meist auf Position abgebildet. Durch die Wahl der Markierung, kann die Interpretation beeinflusst werden. (Liniendiagramm -> kontinuierlich; Scatterplot -> diskrete Messung; ...)

1.1 Vermeidung Overplotting

- Filter: durch Brushes, die durch Boxen dargestellt werden, und eine Zeitreihe muss durch die Brush verlaufen.
- Heatmap: Höhe wird durch Farbe ausgetauscht. Entlang der Zeitachse kann man gut vergleichen, entlang der Werteachse eher nicht.
- Horizontplot: Kompromiss zwischen Liniendiagramm und Heatmap. Werte werden auf einen Farbverlauf abgebildet.
 - hierbei wird jedem Wertebereich ein Farbbahn zugeordnet (divergierende Farbskala)
 - die Bänder werden an der Kurve des Linienverlaufs abgeschnitten
 - negativer Bereich wird auf positive Seite gespiegelt
 - alle Farbbänder werden übereinander gelegt.

⇒ Stärken und Schwächen des Horizonplots: es ist platzsparend und auf der Wertachse besser Vergleichbar als zB Heatmaps. Es kann auf etwa 50% bis 100% parallele Zeitreihen skalieren. Aber der genaue Wert an einer Stelle abzulesen ist so gut wie unmöglich.

2 Visualisierung vieler Zeitreihen

- Heatmaps mit Aggregation: Pro Pixel wird normalisierte Liniendichte bestimmt (Density Heatmap).
- Small Multiples: viele gleiche Visualisierung mit jeweils verschiedenen Datensätzen, wobei nur nah benachbarte Zeitreihen gut vergleichbar sind.
- Small Multiples mit Aggregation: Gruppen ähnlicher Zeitreihen werden als Small-Multiples geplottet. Es erfordert eine Gruppierung/Clustering der Datensätze, welches Zeitaufwändig sein kann.

3 Periodische Zeitreihen

Viele Abläufe wiederholen sich, wie zum Beispiel Schall, Puls oder Jahre, Monate, ...

Das Ziel ist nun eine Vergleichbarkeit über mehrere Zyklen hinweg zu schaffen. Man muss jedoch Periodenlängen bestimmen (Falls nicht gegeben) und sich ändernde Frequenzen handhaben.

3.1 Spiral Layouts

Zeitreihe wird aufgerollt und Polarkoordinaten erzeugen eine Spirale. Die Zyklenlänge ist potenziell variabel und der Zyklus ist als Muster erkennbar.

- Stärken: Kompakt
- Schwächen: Verzerrung, weil Zeitachse nach außen breiter wird. Es funktioniert nur bedingt als Liniengraphik

3.2 Periodische Zeitreihen im Matrixlayout

Zeitachse ist gestapelt. Die View Transformation basiert auf Fivision mit Rest (Mod und Div). Es funktioniert im Prinzip auch beim Liniendiagramm. Die Zyklenlänge ist potenziell variabel und Zyklus ist als Muster erkennbar.

4 Diskrete Werteachse - Events und Dauer

Events haben einen Zeitpunkt, aber nicht unbedingt ein quantitatives Attribut, also ist die y-Achse nicht als kontinuierliche Skala verwendbar. Wenn die Reihenfolge wichtiger ist als die Zeitpunkte, dann können die Events als Sequenz dargestellt werden. Wenn die Dauert zwischen Events wichtiger sind, dann kann ein Gantt-Chart verwendet werden.

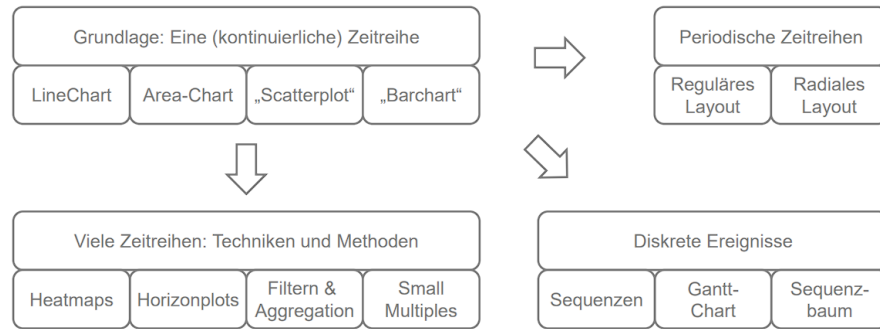
4.1 Mustererkennung in Zeitsreien: Sequenzbaum

Zeitserie wird in wiederkehrende Muster zerlegt. Dies basiert auf Eventfolgen oder diskretisierten Werten. Event-Art ist colorcoded.

Im Sequenzbaum zählt man die Länge der Sequenzen einer bestimmten Länge.

5 Transparency Vis

⇒ Viewer für Datenexports von Google, Facebook, Ein Event ist alles was diese über Sie wissen.



6 Graphen und Bäume

Ein Graph ist mindestens durch eine Kantenliste definiert. Die Kantenliste kann auch Attribute der Kanten enthalten.

6.1 Baum

Folge von Knoten ist ein Pfad, wenn aufeinanderfolgende Knoten durch Kanten verbunden sind. Ein Graph ist zusammenhängend, wenn zwischen je zwei Knoten ein Pfad existiert. Ein Graph ist gerichtet, wenn die Kanten nicht-symmetrisch sind. Ein Graph ist zyklensfrei, wenn es keinen Weg gibt, der einen Knoten mehrfach erreicht. \Rightarrow Ein Baum ist ein zyklensfreier, zusammenhängender Graph. Diese Unterscheidung ist wichtig, da mit einer Ordnung sich Graphen deutlich einfacher darstellen lassen.

6.2 Node-Link Diagramme von Bäumen

Normale Baumansicht, wobei Wurzel oben ist und die Kinder je nach Level weiter unten angeordnet sind.

Kriterien um ein Node-Link Diagramm zu erstellen (nicht alle immer erfüllbar):

- keine Kreuzungen
- alle Knoten einer Hierarchiestufe auf gleicher Höhe
- möglichst schmal
- Elternknoten zentriert über Kindknoten
- Symmetrien sollte man erkennen

- Ordnungserhaltend
- Linearzeit

Ein Problem bei der Baumvisualisierung ist in der Regel die Breite des Baumes.

6.3 Radiales Layout

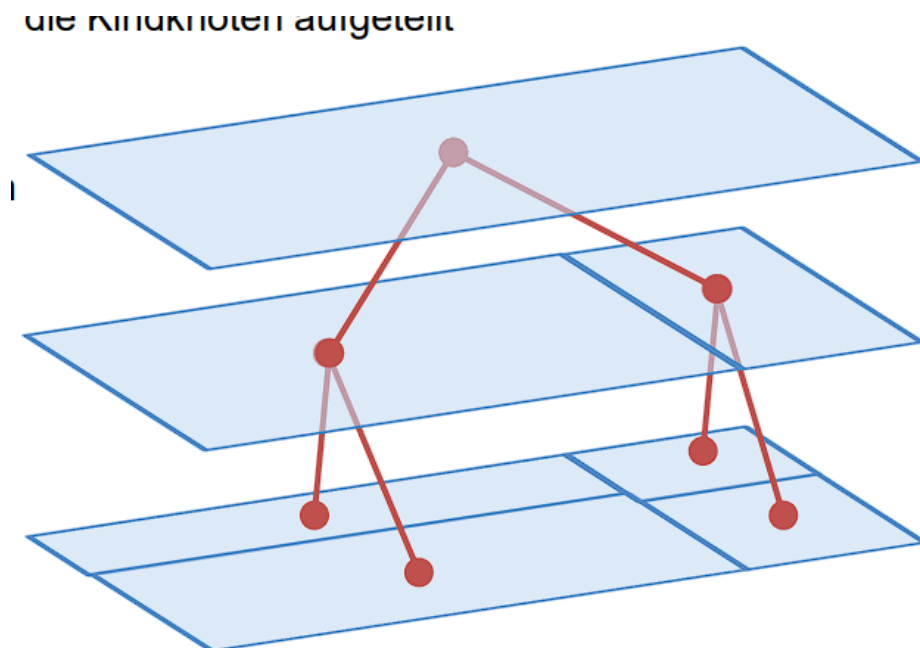
Man kann Bäume Zyklisch darstellen. Hierarchieebenen bilden Kreisringe um Wurzelknoten. Hierbei wird der Platz auf Kreisringen möglichst ausgenutzt. Es limitiert die Breite des Baumes, verdoppelt aber die Höhe.

6.4 Tree-Map

Beziehung zwischen Knoten wird als "enthalten in" dargestellt. Gesamte Visualisierungsfläche ist der Wurzelknoten, Kindknoten sind Unterteilungen (siehe Mosaicplot)

6.4.1 Wie werden Sie erstellt?

Fläche wird rekursiv unterteilt. Fläche eines Elternknotens wird auf die Kinderknoten aufgeteilt, Kanten werden aber NICHT dargestellt. Flächengröße kann aus Knoteneigenschaft abgeleitet werden.



Cushions sind Farbverläufe, mit denen man sich outline zur Begrenzung ersparen kann.

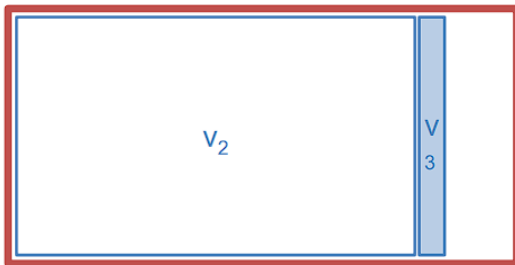
6.5 Squarified Treemaps

pro Hierarchieebene wird sowohl horizontal und vertikal geteilt (je nachdem, wie Platz ist).

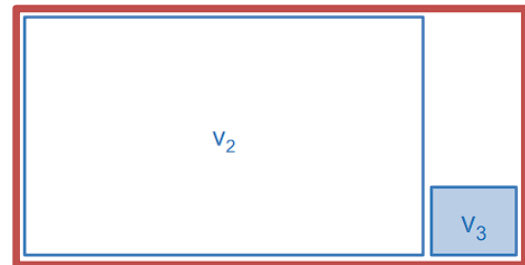
- Vorteil: besser lesbare Treemaps, vermeidet entartete Rechtecke
- Nachteil: Direkte Eltern-Kind-Beziehungen sind nicht mehr eindeutig

6.6 Normale vs. Squarified Treemaps

Unterschied wird am deutlichsten, wenn ein Kindknoten eines KNotens deutlich größer ist als die anderen.



Normale Treemap: Unterteilung immer gleich



Squarified Treemap: Unterteilung wechselt

6.7 Icicle Plots und Sunburst

- Größen über Flächen darstellen
- Sunburst: icicle Plot mit radialem Layout
- Kanten werden nicht explizit dargestellt: Kindknoten stehen unter ihrem Elternknoten (Icicle-Plot), an der Außenseite ihres Elternknotens (beim Sunburst).

7 Visualisierung allgemeiner Graphen

7.1 Node-Link Diagramm

Stärken

- Pfade nachverfolger
- gerichtete Kanten explizit

- theoretisch 10k Knoten

Schwächen

- zwei Layout-Dimensionen
- Dichtbesetzte Graphen kaum lesbar

7.2 Adjazenzmatrix

Stärken

- Keine Überlappung
- Dichte Graphen gut darstellbar
- Gerichtete Kanten \Rightarrow Asymmetrie
- Nur eine Layout-Dimension (Sortierung)

Schwächen

- Verfolgen von Pfaden schwieriger
- Dünnbesetzte Graphen nutzen Platz nicht
- Sortierung beeinflusst die Struktur
- für hunderte Knoten

7.2.1 Kriterien für Graphlayout für Node-Link Diagramme

- unnötige Kantenüberschneidungen vermeiden
- Overplotting vermeiden: Knoten nicht übereinander
- verbindende Knoten sollten nahe beieinander sein \Rightarrow Kanten also kurz halten
- stark verbundene Teilgraphen erkennbar machen
- gegebener Platz soll genutzt werden
- Benachbarte Knoten im Durchschnitt gleich weit voneinander

\Rightarrow allg. Graphlayout ist vielleicht das schwierigste algorithmische Visualisierungsproblem. Verwandte Problemstellungen wie Clustering oder Dimensionsreduktion sind ähnlich schwierig, aber erscheinen nicht nur bei der Erstellung einer Visualisierung. Es handelt sich um ein allgemein ungelöstes Problem.

7.3 Force-Directed Layout: Fruchterman-Reingold

eines der besten Verfahren für diese Klasse. Modelliert die Knoten und Kanten als Masse-Feder-System. Anziehende Kräfte nehmen linear mit dem Abstand zu. Abstoßende Kräfte nehmen quadratisch mit dem Abstand ab.

Stärken

- Einfachheit: kein Wissen über Graph notwendig
- Erlaubt Interaktion
- Kanten und Knoten können Gewichte erhalten

Schwächen

- lange Laufzeit
- keine optimale Lösung

7.4 Layer-Based Layout: Sugiyama

- für allgemeine Graphen anwendbar, wenn man Schleifen entfernt, den Graphen künstlich richtet und Schleifen danach gesondert darstellt
- Kantenrichtung gibt eine Hauptrichtung vor
- Knoten werden in Spalten einsortiert
- Höhe des Knotens in der Spalte zunächst unbestimmt.

7.4.1 einfachste Heuristik

Algorithmus beginnt am vorderen Ende. Eine Iteration testet einfache Modifikation am Graphen. Knoten innerhalb einer Spalte werden getauscht und verschoben. Jede Modifikation wird bewertet, Verbesserungen werden angenommen. Wenn keine Verbesserung mehr möglich ist, geht man in die nächste Spalte über.

Stärken

- einfaches Verfahren
- strukturiert azyklische Graphen

- definiert Hauptrichtung des Graphen

Schwächen

- Graphen mit wenigen Zyklen müssen gesondert behandelt werden
- für Graphen mit vielen Zyklen nicht brauchbar

7.5 Constraint-Based-Layout: Metro-Maps

- Knoten auf einem Gitter
- Kanten verlaufen nur in bestimmten Winkeln
- Kantenkreuzungen häufig in Knoten
- Knotenform an Kreuzungsart angepasst
- Verfahren übernimmt Ideen des Sugiyama

Stärken

- Anwendbar auf allgemeine Graphen, aber Knoten und Kantenverhältnis im Graph sollte ausgewogen sein
- visuell stark strukturiert, vergleichsweise einfach zu lesen

Schwächen

- Teilweise komplexe Heuristiken
- lokal optimale Lösungen
- Pfade müssen ggf. definiert werden

7.6 Edge-Bundling

Reduktion der Komplexität durch Bündeln der Kanten. Kanten werden nicht als kürzeste Wege gezeichnet. Kantengruppen werden identifiziert.

7.6.1 Hierarchical Edge-Bundling

Knoten liegen innerhalb einer Hierarchie

- Statt des direkten Wegs durchläuft eine Kante die Hierarchie bis zum gemeinsamen Vorfahren
- Kantenzug wird über Splinekurven approximiert
- Folge: Mehrere Kanten durchlaufen den gleich Pfad

7.7 Search, Show Context, Expand on Demand

Komplette Graphen sind für viele Aufgaben nicht notwendig

Fokussierte Darstellung eines größeren Graphen. Jeder Knoten wird nach Wichtigkeit bewertet. Zum Beispiel, zuletzt geklickte Knoten, Nachbarschaft, ...

Vor dem Layout werden die Knoten des Graphen sinnvoll gefiltert. Vereinfachung des Layouts auf die Aufgabe. Fokus ist hierbei auf eine im Detail erfassbare Menge. Die Verbindung zu weiteren, "unsichtbaren" Bereichen sichtbar. Navigation im Graph verschiebt dieses Fenster.

Chapter 8 - Visualisierung Geobasierter Daten und Karten

Informationsvisualisierung und Visual Analytics
WiSe 2024/25

1 Karten

Karten funktionieren auch mit unbekannten Daten weil die Assoziationen eine Hilfe bei der Interpretation liefern und man meistens nicht immer auf die eine Assoziation angewiesen ist.

Die Semantik der Karte beeinflusst Distanzen und Unterscheidung von Land und Wasser verzerrt die Distanzen.

2 Kartenmetapher

2.1 Datenstrukturen - Ortsbezogene Daten

Ein Ort wird durch mind. einen key definiert. Ein Ort kann durch Ortsnamen definiert sein oder Koordinaten.

2.2 Datenstrukturen - Bewegungsdaten

Sind eine Kombination aus zeitbezogenen Daten und Ortsbezogenen Daten. Man misst zu verschiedenen Zeitpunkten den Ort.

2.3 Minard Map

2.4 Achtung

Man kann geographische Daten auf Karten darstellen, oder gleiche Graphen auf nicht Karten darstellen. Oder Nicht-geographische Daten, auf Karten dargestellt, oder nicht-geographische Daten, die nicht auf Karten dargestellt.

⇒ Geographische Daten können unterschiedlich abstrakt visualisiert werden. Wenn reale Raumbezüge wiedergegeben werden sind es geographische Karten. Wenn etwas abstrahiert wird mit anderem Fokus sind es schematische Karten (auch Schematisierung genannt).

Wenn abstrakte Daten auf Karten dargestellt werden sind es Imitationen.

2.5 Teil-Fazit

- Aufgrund Nutzung von Karten, kann man mehr Vorkenntnisse vorraussetzen.
- Verschiedene Metaphern können vorausgesetzt werden
- Vorkenntnisse erlauben reichhaltige Assoziationen zum eigenen Weltwissen
- Karten zeigen NICHT immer Geographie

3 Visualisierung geobezogener Daten

3.1 Mapped

Wenn geographische Koordinaten auf 2D oder 3D Koordinaten abgebildet werden, wird eine View Transformation von Geoposition auf Länge und Breite durchgeführt. Das ist die bekannteste Abbildung.

Probleme

- Entfernungen auf der Erdoberfläche proportional zu Entfernungen auf der Karte
- Flächen auf der Erdoberfläche proportional zu Entfernungen auf der Karte
- Winkel auf der Erdoberfläche proportional zu Entfernungen auf der Karte
- Grund: Kugeloberfläche nicht verzerrungsfrei auf eine Ebene abwickelbar

3.2 Kartenprojektionen - Plattkarte

⇒ Koordinaten werden direkt auf (x, y) abgebildet. Je näher man aber an die Pole kommt, desto weiter verstaucht sind Flächen.

3.3 Kartenprojektionen - Mercator

Erdkugel wird auf einen Zylinder projiziert.

$$\begin{aligned}x &= \text{Länge}(L) \\ y &= \arctan(\sin(B))\end{aligned}$$

3.4 Kartenprojektionen - Winke-Tripel

Hierbei sind alle Ellipsen die auf der Karte wären ungefähr gleich groß. Es reduziert somit Flächenverzerrung.

3.5 Projektionsproblem umgehen

Erdoberfläche ist auf kleinen Gebieten in der ersten Näherung flach. Fehler bei der Projektion auf Ebene sind niemals Null aber für viele Zwecke vernachlässigbar.

3.6 Kartenprojektionen - Grundlagen

Es ist möglich getreue Abbildungen der ganzen Erdoberfläche auf einer Kugel zu erstellen. Das Problem wird aber nur verlagert. Eine Projektion von 3D auf 2D ist immernoch schwer. Also Verdeckung und Verzerrung bleiben. Eine getreue Abbildung der Erdoberfläche auf 2D nur näherungsweise möglich.

3.7 Eigenschaft

3.7.1 Erhaltung der Nachbarschaft

Wenn Nachbarschaften erhalten sind, wird die Navigation erleichtert.

3.7.2 Wiedererkennungswert der Orte

Eine Karte die gelernten Konventionen entspricht erleichtert Identifikation und Suche von Orten.

3.7.3 Verzerrung

Bei Verzerrungen wird getreue Darstellung aufgegeben. Verzerrung ändert oft Winkel, Flächen, ...

3.8 Kartogramm

Klasse von sehr unterschiedlichen Techniken um Karten darzustellen.

3.8.1 Stetige Kartogramme

Hier werden Gitter mit originalen Raumkoordinaten und Knotengewichten verwendet. Und Gewichtung drückt benachbarte Gitterknoten in die leeren Bereiche.

3.8.2 Kartogramme basierend auf Distanzdaten

bei solchen Kartogrammen wird keine Eigenschaft eines geographischen Ortes verwendet. Es werden die Distanzen zwischen Paaren von Orten genutzt. Die Reiselänge ist hierbei nicht die Luftliniendistanz.

3.8.3 Dorling Cartogram

Eine diskreter Kartogramme der keine Formserhaltung hat und mäßige Nachbarschaftserhaltung bietet.

3.8.4 Diskrete Kartogramme als Small-Multiples

Bietet einen guten Kompromiss zwischen einfacher Grundform für den Chart und eine grobe Erhaltung der Nähe.

3.9 Abstrakte Geovisualisierung

Im Extremfall sind geographische Positionen nicht relevant für das Layout. Kommt jedoch auf Visualisierung an.

3.10 Zwischenfazit

Karten liegen im Kontinuum zwischen geographisch getreuen und abstrakten Darstellungen.

4 Schematisierung

4.1 Allgemeine Herausforderung: Spatialization

Problem 1: Abbildung von Eigenschaften der Daten auf zweidimensionale Koordinaten

Problem 2: Auswahl relevanter Landmarks für die Orientierung

4.2 Themescapes

Eine Visualisierung der Themen eines Textkorpus. Themen sind definiert über charakteristische Wörter. Die Worthäufigkeit zwischen Themen definiert Ähnlichkeit (über Position definiert).

4.3 Themengebiete

Themen und Themengrenzen werden durch verschiedene Detailstufen dargestellt. Hierbei werden eine hierarchische Metapher von Ländern, Provinzen und Regionen genutzt.

4.4 Imitation Metromap

⇒ Haltestellen sind Roadmap Punkte und es wird eine Geschichte erzählt

4.5 Zwischenfazit

Imitation von Karten umfasst Imitation geographische und schematische Abbildungen. Allgemeine Herausforderung: Spatialization. Weitere Herausforderungen sind die Konstruktion einer Topographie, die im Hintergrund dargestellt werden kann oder die Wahl geeigneter Metaphern die Informationen und Beziehungen effektiv und konsistent übersetzen.

5 Kartographische Abbildung auf Marks und Channels

5.1 Glyphen

Ein kleines unabhängiges visuelle Objekt, dass mehrere Merkmale eines Items zeigt oder eine Menge von Items.

- Wie viele Glyphen möchte ich darstellen? Kommt drauf an wie viele Items man darstellt.

Glyphen sollten visuell einfacher sein wenn Muster statt Werte relevant sind oder die Werteverteilung chaotisch sein kann.

- Sollen viele Werte einzeln lesbar sein? Kommt auf die Wahl der separierenden Channels an
- Sollen viele Werte als Ganzes wahrgenommen werden? Kommt auf Wahl integrierender Channels an

Chapter 9 - Einführung in Visual Analytics

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Visual Analytics

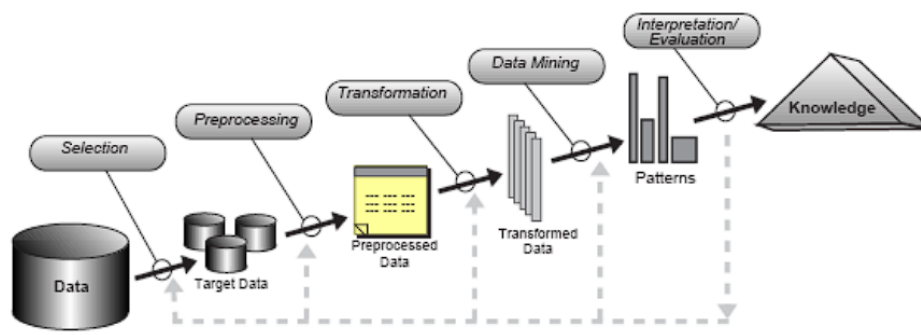
Zentrale Idee: Visual Analytics ist mehr als die Darstellung von Analyseergebnissen. Es geht darum Visualisierung und Data-Mining-Ansätze so zu verbinden und parallel zu nutzen, um so gut wie möglich von Daten zu Wissen zu kommen.

Definition: VA ist die Kombination automatischer Analysetechniken mit interaktiven Visualisierungen um große und komplexe Daten zu verstehen und auf deren Basis Entscheidungen treffen zu können.

Definition Informationsvisualisierung: Nutzung von computergestützten, interaktiven, visuellen Repräsentierungen abstrakter Daten mit dem Ziel, das Erkenntnisvermögen zu verbessern.

Definition Knowledge Discovery in Databases (KDD): Nicht-trivialer Prozess für die Suche nach validen, neuen, potentiell relevanten und nutzbaren Mustern in Daten.

1.1 Knowledge Discovery Process Modell



1.2 Vergleich InfoVis Prozess und KDD

Gemeinsamkeiten

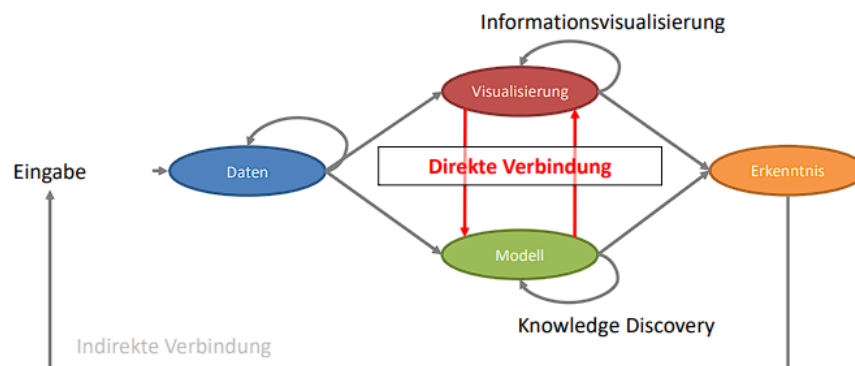
- beide Modelle sind Datenflussmodelle
- alle Prozessschritte sind interaktiv und iterativ

- Ziel ist bei beiden Erkenntnisgewinn
- Methoden und Techniken für Datapreprocessing und Datarepresentation sind im Prinzip gleich

Unterschiede

- InfoVis fokussiert auf Visual Mapping und View Transformation. Die Ergebnisse werden als Bild ausgegeben
- KDD fokussiert auf Data Mining. Darstellung der Ergebnisse ist nicht definiert
- Bei KDD werden Visualisierungstechniken und Data-Mining Techniken meist ohne Modifikationen nebeneinander eingesetzt
- KDD ist eher Automatisierungsfokussiert. InfoVis eher Interaktion fokussiert.

1.3 Visual Analytics Process Modell



1.4 Wissenschaft des analytischen Schließens? Was ist das?

Man kann allgemeine Fragen nicht auf der Grundlage einer einzelnen Beobachtung beantworten. Analytisches Schließen ist hierbei das Herleiten von Entscheidungen aus empirischen Daten und Entwicklung von Methoden für die Herleitung.

2 Stärken und Schwächen von Mensch und Maschine

2.1 Was ist ein Muster?

⇒ Ein Muster ist eine nichtzufällige Teilmenge aller Datensätze.

Was ist ein visuelles Muster? subjektiver Sinneseindruck, der Informationen über mehrere Elemente des Bildes zusammenfasst

Worin unterscheiden sich Muster mit Modell und ohne Modell?

2.2 Modellarten

- Beschreibungsmodell: Ermöglicht es Muster X von Grundmenge ohne Muster zu unterscheiden
- Prognosemodell: Ermöglicht es unbekannte Muster, die nicht Teilmenge der Grundmenge sind X oder Grundmenge X zuzuordnen
- Bei KDD werden Muster fast immer zsm mit dem Modell erzeugt

⇒ wenn eine Maschine ein Muster erkennen soll, benötigt es eine Modellbeschreibung/Algorithmus, ein Mensch braucht eine geeignete Visualisierung.

2.3 Worin unterscheiden sich Muster in InfoVis und KDD

Infovis

- Muster sind Sinneseindrücke, subjektiv, nicht formalisiert und nicht direkt kommunizierbar

KDD

- Muster sind Ausdrücke in formaler Sprache. Muster ist durch Modell definiert, objektiv, reproduzierbar und nachweisbar unzufällig.

2.4 Stärken und Schwächen zws. Mensch und Maschine in der Mustereerkennung

Maschine:

- Stärken: Ergebnis ist eine formale, nutzbare Beschreibung
- Schwächen: Suche ist schwierig. Man grenzt bei der Suche früh ein, wodurch die Suche erschwert wird.

Mensch:

- Stärken: Muster können ohne Beschreibung gefunden werden. Menschen können Erkennung komplexester Muster lernen
- Schwächen: Ergebnisse sind Anwenderabhängig und visuelle Muster sind nicht direkt nutzbar

Stärken des Menschen

- Flexible, robuste Wahrnehmung
- Mustererkennung unter schwierigsten Bedingungen
- Wahrnehmung ungewöhnlicher und unerwarteter Ereignisse
- Umgang mit Unsicherheit
- Kreativität
- Langzeitgedächtnis, Weltwissen
- Urteilskompetenz
- [Induktives Schließen]
- Abduktives Schließen (vom Ergebnis auf eine Prämisse, Hypothesenbildung)

Stärken der Maschine

- Kontrollierbarkeit
- Wiederholbarkeit
- Großer Arbeitsspeicher
- Deduktives Schließen (von Prämisse und Regel auf das Ergebnis)
- Schnelligkeit
- Präzision
- Zuverlässigkeit
- Multitasking
- Ausdauer

Schwächen des Menschen

- Kleines Arbeitsgedächtnis
- Kognitiver „Flaschenhals“ - sequentielle, sprachliche Verarbeitung
- Umgang mit hochdimensionalen Daten
- Geringe Ausdauer
- Abhängigkeit von äußeren Einflüssen
- Nichtreproduzierbarkeit von Ergebnissen
- Abduktives Schließen (Falsche Prämissen)

Schwächen der Maschine

- Umgang mit Rauschen in den Daten
- Umgang mit Mehrdeutigkeiten
- Umgang mit Datenunsicherheit
- Formale Beschreibung notwendig
- Sehr begrenzte Anpassung an neue Reize
- Umgang mit *sehr* hochdimensionalen Daten

Chapter 10 - Analyse für die Visualisierung

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

1 Visualisierungsanalyse

- Nenne technische Möglichkeiten für die Kopplung zwischen interaktiven und automatischen Verfahren
- Wie kann eine Visualisierung durch automatische Verfahren verbessert werden
- Zu Schwächen existierender Verfahren geeignete Ansatzpunkte für die Kopplung identifizieren

Wo haben Visualisierungen ihre Schwächen?

- Bildschirmplatz ist begrenzt
- menschliches Arbeitsgedächtnis begrenzt
- Daten können nicht einzeln zu Information verarbeitet werden
- Drei- und mehrdimensionale Beziehungen nicht direkt darstellbar
- vier- und höherdimensionale Beziehungen nicht vorstellbar
- beliebig hochdimensionale Beziehungen können nicht dargestellt werden
- Ergebnisse gehen über die Zeit vergessen
- Visualisierungsinterpretation ist kein deterministischer Prozess (Verschiedene Nutzer erzeugen verschiedene Ergebnisse)
- Visualisierungen machen Muster wahrnehmbar, die keine sind
- visuelle Auffälligkeit \neq statistische Auffälligkeit
- Interaktion verführt, das erwartete Resultat hervorzuheben
- durch Interaktion werden verschiedene Visualisierungen schlechter vergleichbar
- Interaktion bedeutet Lernaufwand

Welche Ansatzpunkte gibt es um Visualisierungen zu Verbessern?

- Reduktion der Datenmenge durch Sampling, Aggregation/Clustering oder durch Filtering
- Bereinigung der Daten durch visuelle Aggregation, Hervorheben der relevanten Daten oder Layout

Was ist Sampling?

eine Reduktion der Anzahl der Datenelemente, die für eine Visualisierung noch sinnvoll ist. In der Regel eine zufällige Auswahl, aber nicht so einfach wie es klingt.

Wie viele Samples sind mind. notwendig, um eine zentrale Aussage nicht zu verfälschen?

Was ist inkrementelles Sampling?

Anwendet entscheidet selbst, wann genug Daten gezeigt werden

Was ist Aggregation/Clustering?

Aggregation: Zusammenfassen von Datenelementen zu Mengen und Visualisierung dieser Mengen

Was ist Clustering?

Es ist ein Vorverarbeitungsschritt zur Aggregation. Hierbei wird eine ClusterID eines Datenattributes erzeugt und diese werden wie normale Daten behandelt.

Was ist Filtering?

eine Berechnung von einem Degree of Interest. Für jeden Datenpunkt wird berechnet, ob diese für aktuelle Visualisierung relevant ist. Wenn ja wird sie angezeigt.

1.1 automatische Verfahren

Wie kann man eine Visualisierung durch automatische Verfahren verbessern?

- Dimensionsreduktion
- Feature Selection
- Features Sortieren

1.1.1 Dimensionsreduktion

Projektion von x -Dimensionalen Punkten zu y -Dimensionalen Punkten, wobei gilt: $x > y$. Dies kann durch folgende Projektionen durchgeführt werden:

- Lineare Projektion
- lokal-lineare Projektion
- nicht-lineare Reduktion

1.1.2 Feature Selection

Visualisierung kann nur n -Features darstellen. Man sucht nach Kombinationen von n existierenden Features, mit maximal vielen Information. Gesuchte Features sind statistisch unabhängig, und Duplikate entfernen.

1.1.3 Feature Sorting

Visualisierung oft abhängig von der Featureanordnung (siehe: Parallelkoordinaten, Star-Coordinates, ...). Ob Muster erkannt werden ist dann Glückssache. Hierbei sucht man guten Anordnungen. Bsp.: Matrixsortierung

1.1.4 Was sind Historymechanismen?

Managementmechanismen, um Ergebnisse beizubehalten.

- Muster, Visualisierungseinstellungen, Notizen, ...

1.1.5 Welche Organisationswerkzeuge gibt es?

Werkzeuge um Ergebnisse zu abstrahieren. Der Vergleich, Bewertung, Austausch der Ergebnisse findet hier statt

1.1.6 Wie werden Entdeckungen gemanaged?

- aus Interaktion werden relevante Muster identifiziert
- Muster werden als Mengen abgespeichert
- Mengen werden als neuer Datentyp interpretiert

1.1.7 Wieso erkennen Menschen Muster die keine Muster sind?

- Menschen konstruieren Zusammenhänge oftmals nur aus wenigen Samples.
- Mustererkennung geht häufig eine nicht gute Allianz mit der Fähigkeit Hypothesen zu bilden ein

1.1.8 Was ist das Police-line-Up Szenario?

Fünf von Sechs Plots werden künstlich erzeugt und nur ein Plot hat die echten Daten. Wird dieser eine Plot von Nutzern gefunden?

1.1.9 Was ist automatische Steuerung in der Interaktion?

1.1.10 Wie kann man Nutzer durch eine Interaktion führen?

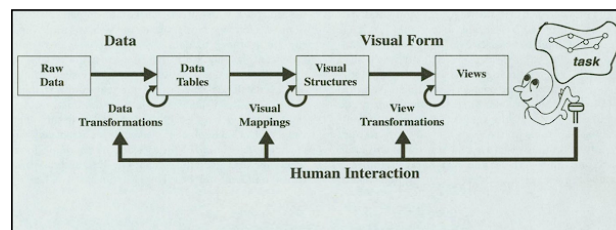
1.1.11 Wie kann man die Anzahl der Parameter verringern?

Man sucht nach den optimalen Parametern. Hierfür erzeugt man mehrere Parameterkombinationen und probiert aus.

1.1.12 Warum sollte man Parameter automatisch einstellen?

Bei zu vielen Parametern weiß man nicht welche Parameter man ändern muss. Der normale Prozess sieht so aus: Muster identifizieren und interpretieren. Und dann Suche in der UI und den Einstellungen. Bei der automatischen Parametereinstellung wird dieser Prozess verkürzt.

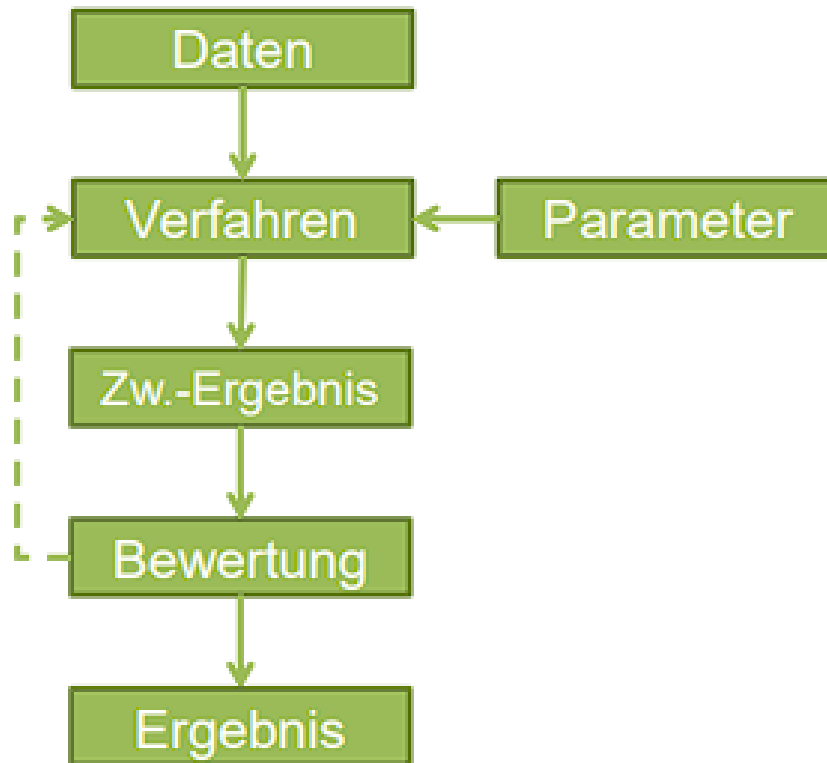
1.2 Wie kann das InfoVis Modell durch automatische Verfahren beeinflusst werden?



- Dimensionsreduktion bei Data Transformations
- Feature Selection / Sorting bei Visual Mapping
- Dimensionsreduktion (Kamera) bei View Transformation
- Resultmanagement am Ende

2 Verbesserung der Modellierung durch Visualisierung

2.1 Wie sieht das automatische Verfahren allgemein aus?



2.2 Wie funktioniert K-Means?

Man definiert n Zentroiden. Man markiert alle Datenpunkte mit der Farbe zum nächsten Zentroiden. Wenn alle Datenpunkte gefärbt sind, berechnet man den Mean von jeder Farbe um den Zentroiden zu verschieben.

2.3 Was sind Schwächen von Automatischen Verfahren?

- Häufig ist Ergebnis eines Data-Mining Verfahren ein Modell. Modell ist nicht sichtbar. Lösung davon ist Modellvisualisierung.
- Manchmal ist das Ergebnis ein Muster, welches nicht richtig oder falsch ist.
 - Warum bilden Elemente ein Muster?
 - Worin unterscheiden sich zwei Muster?
 - Lösung: Musterexploration (PatExp)

Chapter 11 - Von Daten zu Muster - Visualisierung für die Analyse I

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

Fragen für Karteien

- Checken ob In Karte: Was sind Glyphen?
-
- Wie definiert man Ähnlichkeit?
- Ist die Menschliche Mustererkennung und Ähnlichkeitsbegriff abhängig?
- Wie kann man automatische Verfahren durch Visualisierungen verbessern?
- Was ist das zentrale Problem bei der Ähnlichkeit?
- Was sind Clusteringverfahren?
- Was ist die Analytische Fragestellung beim Clustering?
- Was ist die Dimensionsreduktion?
- Was ist die analytische Fragestellung von Dimensionsreduktion?
- Unterschiede und Gemeinsamkeiten von Clusteringverfahren und Dimensionsreduktion
- Welche verschiedenen Distanzmaße sind üblich? (Manhattan, Euklidische, Maximum)
- Warum machen wir Mustererkennung?
- Wie könnte eine Faustregel für Clustering und Dimenionsreduktion aussehen?
- Wie definiert Vorverarbeitung die Ähnlichkeit mit?
- Worin unterscheiden sich informelle und formale Ähnlichkeitsdefinition?
- Was ist die Ground Truth und weshalb wird diese definiert?
- Wo haben automatische Verfahren ihre Schwächen?
- Was ist die Black-Box Integration?
- Was ist Semi-Supervised Clustering?
- Definiere Supervised, Partial Labelling, Partial Constraints, Unsupervised.
- Was ist Visual Input Editing?
- Was ist White Box Integration?

- Was ist Model-Data-Linking?
- Was ist Hierarchisches Clustering?
- Wie funktioniert DB Scan? Und was ist dessen Vorteil?
- Wieso sind komplexere Strukturen problematisch mit Clustering?
- SOM als nichtlineare Dimensionsreduktion
- Welche Dimensionsreduktionsverfahren sind linear?
- Was sind die Einschränkungen von SOM?
- Was ist der Hilfreiche Teil von Visualisierung bei Clustering?
- Unterschied Modifikation und Interaktion?

1 Unterstützung von Analyse durch Visualisierung

1.1 Grundbegriffe

1.1.1 Was sind Clusteringverfahren?

Diese Verfahren berechnen eine automatische Gruppierung von großen Datenmengen nach ähnlichen Eigenschaften. Clustering wird in der Regel ohne Vorwissen ausgeführt.

Analytische Fragestellung: Worin besteht die Vielfalt einer Datenmenge?

1.1.2 Was ist Dimensionsreduktion?

Dimensionsreduktion dient der Suche nach relevanten Eigenschaften in den Daten. Dimensionsreduktion wird in der Regel ohne Vorwissen durchgeführt.

Analytische Fragestellung: Welche numerischen Eigenschaften beschreiben die Vielfalt am besten?

1.1.3 Vergleich Clustering und Dimensionsreduktion

Sei ein Datenobjekt ein Vektor \vec{v} von Attributen.

- Clustering:
- Dimensions

Chapter 12 - Visualisierung für die Analyse II

Informationsvisualisierung und Visual Analytics

WiSe 2024/25

- Folie 9 Lernziele!
- Welche Klassifikationsverfahren sind modellbasiert? Welche sind Nicht Modellbasiert?

1 Grundbegriffe

1.1 Unterschied zwischen Clustering und Klassifikation

Clustering und Klassifikation erlernen eine Funktion.

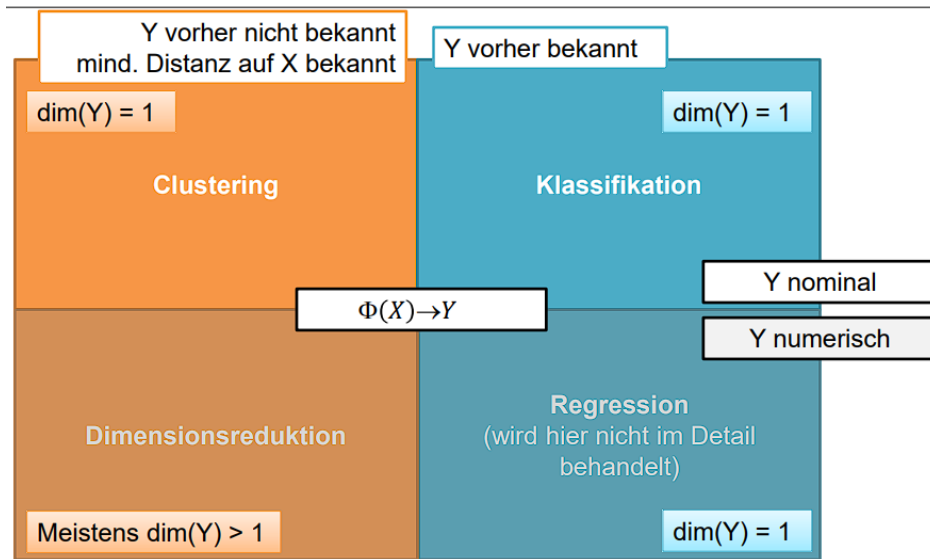
$$\phi(X) \rightarrow Y$$

Unsupervised Learning: Datenobjekte erhalten keine Information über Y

- Clustering: Datenobjekte $d \in X$ gegeben (Y ist nominal)
- Dimensionsreduktion: Datenobjekte $d \in X$ gegeben (Y ist numerisch)

Supervised Learning: Datenobjekte d enthalten Information über X und Y .

- Klassifikation: Datenobjekte $d \in X \times Y$ gegeben (Y ist nominal)
- Regression: Datenobjekte $d \in X \times Y$ gegeben (Y ist numerisch)



1.1.1 Qualitätskriterien für supervised Verfahren

- getreute Wiedergabe der gewünschten Abbildung $\phi(X) \rightarrow Y$
- Verallgemeinerbarkeit: Modell soll auch bei neuen Daten korrekte Ergebnisse liefern
- Geringe Komplexität, durch kurze Beschreibung, Modell soll durch Menschen lesbar und interpretierbar sein

⇒ meist können nicht alle Kriterien erfüllt werden, weil die Daten in der Regel nicht so sauber sind

2 Klassifikationsverfahren und Entscheidungsbäume

2.1 Entscheidungsbäume

Klassifikationsmodell mit einer Hierarchie, in der innere Knoten Entscheidungen repräsentieren und Kanten die Entscheidungsoptionen repräsentieren.

Eine Klassifikation eines Datenobjekts x durchläuft einen Pfad in einem Entscheidungsbaum. Erreichter Blattknoten definiert Ergebnis Y des Klassifikators.

2.1.1 Qualitätskriterien bei Entscheidungsbäumen

Ziel ist es die Entscheidungsbaumkomplexität gering zu halten.

- geringe Tiefe
- viele Fälle durch wenige Entscheidungen abdecken
- Entscheidungen nach Wichtig ordnen
- Jede Entscheidung sollte Klassen gut trennen

2.1.2 Entscheidungsbaumvarianten

- Zwei Teilbäume oder Variabel
- Univariate Attribute pro Knoten
- Attributpartitionierung (i, i, ...)

Der Entscheidungsbaum wird durch die Trenngenauigkeit der Klassen bewertet.

2.1.3 Visuelle Verfahren von Entscheidungsbäumen

- Automatisches Verfahren:
 - Suchen bei numerischen Attributen "Split-Points"
 - Ziel ist es Teilintervalle die homogene Mengen sind (wenn möglich)
 - eigentlich ein Clustering Problem
 - Brute-Force Möglich
- Interaktive Verfahren:
 - Visualisierung der Klassenverteilung über Wertebereich
 - Split-Points interaktiv definiert
 - Teilintervalle mit möglichst homogenen Mengen

⇒ für nominale Attribute ist Split-Point Bestimmung schwerer, weil die Sortierung/Ordnung der Werte egal ist.

2.1.4 Messung der Homogenität durch Gini-Index

$$\text{Gini}(G) = 1 - \sum_i p_i^2$$

p_i ist der Anteil von y_i an einer Gruppe

Gini-Index gibt an wie homogen die Gruppe ist. Ist der Index 0, wenn Gruppe homogen ist. ⇒ Formel beschreibt die Homogenität einer Gruppe G. Die Qualität der Gruppierung bestimmt durch Homogenität aller Gruppen

$$\text{Gini}_{\text{Ges}} = w_1 * \text{Gini}(G_1) + w_2 * \text{Gini}(G_2)$$

w_i : relative Größe von G_i

⇒ der Wert ist vor allen relevant um mehrere Split-Points zu vergleichen

2.1.5 Messung der Homogenität durch Entropie

$$\text{Entropy} = - \sum_i p_i * \log_2(p_i)$$

Index ist 0, wenn Gruppe homogen. Index ist maximal, wenn Klassen gleichverteilt sind.

Information Gain mit Entropiefunktion: bestimmt die Verbesserung der Homogenität

$$\text{IG} = -\text{Entropy}(G) + \sum_i \frac{|G_i|}{|G|} \text{Entropy}(G_i)$$

Die linke Entropie ist die Entropie vorher, rechts nacher.

3 Integrationsvarianten mit Visual Analytics

3.1 Overfitting Problem

Overfitting ist das Überanpassen eines Modells an die Trainingsdaten. Hierdurch werden die Trainingsdaten gut repräsentiert, aber andere Testdaten würden nicht akkurat genug abgebildet werden.

Trainingsdaten ist ein der Teil der Datensätze, mit denen das Modell berechnet wird. Testdaten ist der Teil der Datensätze, mit denen das Model getestet wird.

3.1.1 Strategien gegen Overfitting aus dem Data Mining

- Pruning:
- mehrere Modelle auf unterschiedlichen Trainingsdaten: Ergebnisse werden durch Voting gemittelt. Arbeitet mit Kreuzvalidation und Random Forest.

⇒ Problem: Grund für das Overfitting ist, dass Rauschen im Datensatz mitmodelliert wird. Das Verfahren kennt die Grenze zwischen Muster und Rauschen nicht. Ziel ist es Muster vom Rauschen zu trennen durch den Menschen.

3.2 Visual Input Editing

Automatische Verfahren sind schlecht in der Trennung von Muster und Rauschen. Hierbei kann Visual Input Editing helfen, in der die Arbeit zwischen Mensch und Maschine geteilt wird. Der Mensch erkennt hierbei die Muster und die Maschine modelliert das Modell.

Idee: Markierung des Musters werden durch den Menschen gesetzt. Originallabel Y werden durch getauschte Labels ersetzt.

Voraussetzung: Daten-Visualisierung zeigt die existierenden Klassen an, sodass der Mensch das Rauschen identifizieren kann.

3.2.1 Ablauf

Label Y wird ersetzt durch neues Label \bar{Y} .

- \bar{y}_1 : wurde vom Nutzer selektiert
- \bar{y}_2 : wurde gerade nicht vom Nutzer selektiert

Die Muster werden interaktiv definiert. Dieses enthält dann kein Rauschen.

Visual Input Editing funktioniert für alle Klassifikationsverfahren, solange die Visualisierung die Klassen zeigt.

3.3 Model Data Interaktion

Automatische Verfahren können nicht beliebige Muster gut modellieren. Es ist nicht sichtbar, ob das markierte Modell auch wirklich modelliert wird.

Model Data Interaktion ist eine Erweiterung von Model-Data-Linking. Modell liefert Feedback über das erzeugte Muster und das Modell wird auf die Daten X angewendet. Es bietet einen visuellen Vergleich des erzeugten Musters und des Originals.

Der Anwender selektiert das Muster, und aus den Musters wird automatisch ein Modell erstellt. Es beschreibt, welcher Regel die Selektion folgt. Im Feedback wird das Modell wieder auf die Originaldaten angewendet.

3.4 Visual Model Verification

Automatische Verfahren kann sich nicht selbst bewerten. Bewertungsschema für alle Daten und Klassen ist gleich. Dadurch werden systematische Fehler nicht erkannt. Alle Fehler werden gleich gewichtet. Fehler können nicht lokalisiert werden.

Das Ziel der Visual Model Verification ist die Klassifikation und die Klassifikationsfehler zu bewerten. Hierbei wird der Unterschied zwischen modellierten und tatsächlichen Daten als Kenngröße verwendet. Beim Data-Mining werden alle Fehler zu einem Wert zusammengefaßt. Dies gilt als allgemeines Qualitätskriterium. Aber das ist schlecht für die Suche nach Verbesserungsmöglichkeiten.

Unterschied zwischen modellierten und tatsächlichen Daten wird in einer Confusion Matrix dargestellt.

	Y=Wut	Y=Ärger	Y=Freude
Φ = Wut	110	55	5
Φ = Ärger	60	124	14
Φ = Freude	3	21	180

Es liefert Informationen darüber welche Klasse gut, welche weniger gut unterschieden werden. Außerdem ist Fehlklassifikation mit unterschiedlichen Kosten verbunden. Die Kosten können durch Gewichte in der Confusionsmatrix angegeben werden. Die Gewichte können bei der Optimierung der Klassifikation einfließen.

Differenz kann nominal, ordinal oder numerisch sein. Die Differenz kann jedem Datenobjekt zugeordnet werden. Die Differenz kann behandelt werden, als wäre es ein normales Datenattribut.

3.4.1 Abhängigkeit Fehler und Daten

Wenn die Klassifikationsfehler ein Rauschen bilden, ist es kein großes Problem. Dann braucht man mehr Daten. Wenn es ein Muster erzeugt, dann gibt es einen systematischen Fehler.

3.4.2 Visual Model Verification - Strategie

Es versucht detaillierte Visualisierung der Klassifikationsfehler für die Diagnose zu erzeugen. Die Klassifikationsfehler werden dargestellt und optional gibt es interaktive Optimierung der Fehlergewichtung.

VMV ist nützlich für Klassifikation und Clustering. Es kann verwendet werden um Muster zu definieren, oder um Modellierungsfehler zu lokalisieren.

4 Klassifikationsverfahren jenseits von Entscheidungsbäumen

4.1 Support Vector Machines

Entscheidungsbäume haben eine immer achsenparallele Grenze zwischen Klassen. Support Vector Machines suchen Ebene die zwei Klassen trennt mit möglichst großen Abstand.

$$y_i = \text{sgn}(\vec{w} * \vec{v} + \vec{b})$$

Wenn nicht alle Klassen durch eine Ebene trennbar sind, dann wird die Anzahl der Dimensionen erhöht.

$$(x_1, x_2) \rightarrow (x_1, x_2, \frac{x_1^2}{x_2})$$

Es werden in der Regel polynomische Kernel genutzt. Dies bedeutet, dass die Trennlinie nun eine Polynomfunktion sein kann. Die Punkte die in der Nähe der Grenze sind dienen als Stützpunkte.

4.2 K-Nearest Neighbor

Ein Sample-basiertes Klassifikationsverfahren, dass nicht modellierbar ist. Die Samples sind gelabelte Datenitems und der rest ungelabelt.

Man weist ungelabelte Objekte den gelabelten in der Nähe zu.

4.2.1 Unterschied zu K-Means

- K-Means: Clustering
- K-NN: Klassifikation
- K-Means: k = Clusteranzahl
- K-NN: k = Zahl der Nachbarn

4.3 Weitere Verfahren

- Bayes-Klassifikation: nutzt Bayes-Theorem für bedingte Wahrscheinlichkeit
- Markov Netze
- Deep NN