

# Mensch-Maschine-Interaktion

M. Eng. Elektrotechnik

Klausur vom 2. Februar 2022: Beispiellösungen

Jörn Loviscach

Versionsstand: 11. Februar 2022, 20:20



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

1. Wenn die S-Zellen fehlen, fällt die Unterscheidung schwer, ob ein (additiver) Blauanteil enthalten ist oder aber nicht. Also ist Rot (R) schwer von Magenta (R+B) und Grün (G) schwer von Cyan (G+B) zu unterscheiden. Blau (B) und Gelb (R+G) sind dagegen von den anderen angegebenen Farben unterscheidbar.
2. Gemäß Fitts wächst die Zeit zum Bedienen mit dem Verhältnis von Abstand zu Größe. Wenn man zum Beispiel die Bewegung von der Mitte der Taste „1“ zu irgendeiner Stelle der Taste „2“ betrachtet, ist dieses Verhältnis bei der rechten Tastatur etwas größer (Skalierung der Abmessungen um den Faktor 2, aber mit Zwischenraum, also relativ verringerter Größe). Also wäre die linke Tastatur schneller zu bedienen als die rechte. Das ist unplausibel, weil man sich zeitraubend bemühen müsste, die winzigen Tasten der linken Tastatur korrekt zu treffen.
3.
  - a) Checkliste zum Abhaken: Lapse (es fällt schwerer, Schritte auszulassen, hinzuzufügen oder zu vertauschen).
  - b) Abschließbare Kappe über Schalter: Regelverstoß (keine Bedienung ohne Schlüssel oder Hammer), Slip (gewisser Zwang, zum Aufschließen innezuhalten)
  - c) Undo-Funktion: Slip, Lapse, Mistake. Man kann viele Arten von Fehlern ungeschehen machen, nachdem man deren Resultat gesehen hat.
  - d) Ausführliche Beschriftung von Knöpfen: Lapse (man kann nicht so leicht vergessen, ob der Knopf das tut, was man will), Mistake (es wird einem klar, dass dieser Knopf nicht zum Ziel führen kann).
  - e) Klare Strukturierung von Abläufen: Lapse (man vergisst nicht so leicht die Elemente und deren Reihenfolge), Mistake (es ist klar, wozu wann was getan werden muss).
4. Das Programm beauftragt mit einem oft `update` oder `invalidate` genannten Kommando das darunterliegende System, bald ein Paint-Event zu

---

feuern. In der Ereignisbehandlungsroutine für dieses Paint-Event führt das Programm dann die nötigen Zeichenoperationen aus. Auf diese Art kann das System Zeichenoperationen zusammenfassen und bei hoher Last verzögern; außerdem gibt es im Code des Anwendungsprogramms eine zentrale Stelle zum Zeichnen.

5. Ein Webserver (ein Programm wie Apache, aber ebenso der Rechner, auf dem dieses Programm läuft) nimmt Anfragen von Webbrowsern entgegen und liefert denen Daten in Formaten wie .html, .js, .css, .jpg zurück. Diese Daten können aus real existierenden Dateien stammen, können aber auch – erst auf die Anfrage des Webbrowsers hin – auf dem Server per Programm oder Skript erzeugt werden. Insbesondere lassen sich HTML-Dateien um Skripte in der Sprache PHP erweitern, so dass der Server das HTML mit dem darin eingebauten Ergebnis der Skripte ausliefert.
6.
  - a) Geheimtür: eine Affordanz, die man – zumindest ohne Tipp – nicht erkennt, dann also keine *Perceived* Affordance.
  - b) Ausgegrauter Menüeintrag: sozusagen eine negative Affordanz, eine *Unmöglichkeit* der Bedienung; allerdings steckt darin eine Information – nämlich darüber, was aktuell *nicht* geht.
  - c) Unterstreichung zur Hervorhebung von Text: eine nur scheinbare Affordanz, weil man den Text nicht – was man glauben könnte – als Hyperlink anklicken kann.
  - d) Alt-Taste zur Tastaturbedienung von Menüs: mangels erkennbarem Hinweis darauf keine *Perceived* Affordance, aber eine vielleicht gelernte Affordanz.
  - e) Chatbot: zunächst nur eine unklare Affordanz, denn man weiß nicht, was die Maschine verstehen und antworten kann.
7. Die Kurve ist im Regelfall monoton wachsend: Je mehr der wirklich Positiven als positiv erkannt werden sollen, desto mehr *False* Positives muss man ertragen. Versuchen wir, den Klassifikator zu verbessern: Dann könnte bei gleich bleibender Rate der falsch Positiven die Rate der korrekt Positiven höher als bisher sein, also der Bauch der Kurve nach oben wandern<sup>1</sup>; oder es könnte bei gleich bleibender Rate der korrekt Positiven die Rate der *False* Positives kleiner als bisher sein, also der Bauch der Kurve nach links wandern. Die Fläche rechts unterhalb der Kurve, also die AUC, wird in beiden Fällen größer.
8. *Zum Beispiel:*
  - Die Trainingsdaten spiegeln die Ungerechtigkeit der Welt wieder. Ein so trainiertes System liefert wieder ungerechte Ergebnisse.
  - Maschinenlernen etwa in der Bilderkennung erlaubt eine umfassende Nachverfolgung von Personen. (Oder umgekehrt kann der feste Glaube

---

<sup>1</sup>Sicherheitshalber angemerkt: nicht *zu* weit nach oben; sie muss immer noch monoton wachsend bleiben.

---

an diese Möglichkeit dazu führen, dass man Systeme installiert, die viele falsch-positive Verdächtigungen produzieren.)

- Dass in allen Situationen ein verlässliches oder zumindest ungefährliches Resultat erreicht wird, ist kaum nachweisbar.
- Konzentration von Marktmacht: Viele führende Modelle sind in der Hand von großen Unternehmen oder lassen sich wegen ihrer Komplexität nur in den Clouds der großen Unternehmen auswerten.
- Das Training großer Modelle benötigt viel Hardware und viel Energie.

9. *Zum Beispiel:* Es geht um die (wahrscheinlich inkrementelle) Verbesserung eines existierenden Produkts, also quasi eine weitere Iterationsrunde in einem spiralförmigen Prozess. Also wird man zunächst den Status quo zu erfassen versuchen, zum Beispiel so:

- Über Fokusgruppen könnte man genauer zu erfahren versuchen, was Anwender\*innen denken und was ihnen widerfahren ist.
- Mit dem Think Aloud Protocol könnte man sich anhören, wo und wie welche Gedanken (und welche Ärgernisse?) bei der Bedienung auftreten.
- Man könnte die Elektronik mitzählen lassen, welche Bedienschritte wann wie oft ausgeführt oder abgebrochen worden sind.

Auf Basis der so gewonnenen Erkenntnisse implementiert man eine neue Version (vielleicht erst nur als Mockup) und testet sie.

10. Wenn die Nullhypothese gilt, treten  $p$ -Werte zwischen 0,00 und 0,05 (die Werte der Teststatistik liegen in den entferntesten 5 % der Verteilung, unter der Annahme, dass die Nullhypothese gilt) genauso häufig auf wie  $p$ -Werte zwischen 0,80 und 0,85 (die Werte der Teststatistik liegen zwischen 80 und 85 % der Verteilung, unter der Annahme, dass die Nullhypothese gilt). Wenn die Nullhypothese *nicht* gilt, wird die Verteilung der Teststatistik extremer<sup>2</sup>, so dass  $p$ -Werte im niedrigen Bereich häufiger auftreten.

---

<sup>2</sup>Anmerkung jenseits des Prüfungsumfangs dieser Klausur: Wenn nicht nur die Nullhypothese nicht gilt, sondern man zum Beispiel die Bedingungen der jeweiligen Teststatistik – etwa, dass die Werte normalverteilt sind – verletzt, ist das nicht mehr gewährleistet.