Infrastruktur für Web Intelligent Systems

Thema: Business Intelligence – Teil II: Data Mining & Knowledge Discovery

von Christian Merker

Gliederung

- **Web-Intelligent-Systeme**
  - Begriffsklärung
  - Personalisiertes Web
  - Infrastruktur und Datenhaltung
- **Verfahren zur Ähnlichkeitssuche**
  - Algorithmen
  - Beispiele
- **Prefetching**
  - Begriffsklärung
  - Prefetching-Methoden
- **Zusammenfassung**

Web-Intelligent-Systeme

- Was ist Web-Intelligence ?
  - Es gibt eine Vielzahl von Definitionen, die sich in Aussage und Formulierung sehr unterscheiden.
  - Alle besitzen einen gemeinsamen Grundgedanken:

    Daten über die Besucher von Internet-Seiten zu sammeln, um daraus auf die Interessen und Vorlieben des Besuchers zu schließen bzw. bei Benutzereingaben dieses Wissen zu verwenden.

Web-Intelligent-Systeme

- Was sind Web-Intelligent-Systeme ?
  - Web-Intelligent-Systeme sind Komponenten,
    die Algorithmen zur Ähnlichkeitssuche implementieren.
- Aufgaben:
  - Sammeln von Daten über die Benutzer,
  - Analyse der Daten,
  - Erstellung von Benutzerprofilen aus analysierten Daten,
  - Bereitstellung von Funktionalitäten für andere Systeme
    z. B. in E-Commerce-Systeme.

Web-Intelligent-Systeme

- Wie werden sie eingesetzt ?
- Einsatzgebiete:
  - **Portale**
    - Anpassung des Arbeitsbereichs an individuelle Bedürfnisse.

    ➔ Schnellere Erreichbarkeit der gewünschten Informationen.
  - **Marketing-Instrument**
    - Verwendung von Wissen über Benutzervorlieben, um interessante Produkte anzubieten.

    ➔ Personalisierte Werbung!

Personalisiertes Web

- Ziel: Möglichst viele persönliche Daten über den / die Benutzer sammeln!

- **Datenerhebung:**
  - **Bewertungsfragebogen**

    ➔ Benutzer weiß, welche Daten er preisgibt.
  - **Digitale Fußspuren**

    - Besuchte Links auf einer Seite oder

    - Verweildauer auf Seite.

    ➔ Benutzer weiß nicht welche Daten er preisgibt.

Personalisiertes Web

- Realisierung
  - **Cookies:**
    - Speicherung des Cookies auf Client-Seite,
    - Eindeutige Identifikation des Benutzers im System,
    - Auslesen der Daten im Cookie beim Seitenaufruf.
  - **Registrierung:**
    - Benutzer erhält Benutzernamen,
    - Eingabe persönlicher Vorlieben (Profil),
    - Dauerhafte Verwaltung der Benutzerdaten auf Server-Seite.

Personalisiertes Web

- Verwendung der Daten:
  - Persönliches ansprechen des Benutzers,
  - Gestaltung der Seite nach Benutzerwünschen.

    (z. B. Benutzer erhält beim Anmelden im System alle Börsendaten, die er beim letzten Besuch betrachtet hat.)

- Schattenseiten:
  - Verkauf der Profile,
  - „Belästigung“ durch sog. Spam-Mails.

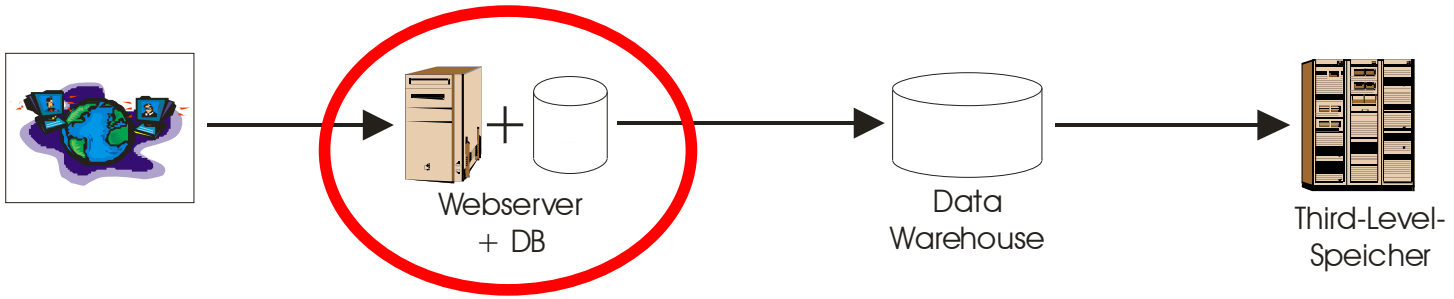
Infrastruktur und Datenhaltung

- Probleme:
  - Entstehen von großen Datenmengen durch Interaktion,

    ➔ „Click-Stream-Daten“
  - Hoher Datendurchsatz erforderlich.
- Lösung:
  - Entfernen von Rauschen,
  - Analyse der Daten (Aggregation von Informationen),
  - Verwendung einer Speicherhierarchie:
    - Web-Server mit Datenbank,
    - Data-Warehouse und
    - Third-Level-Speicher.

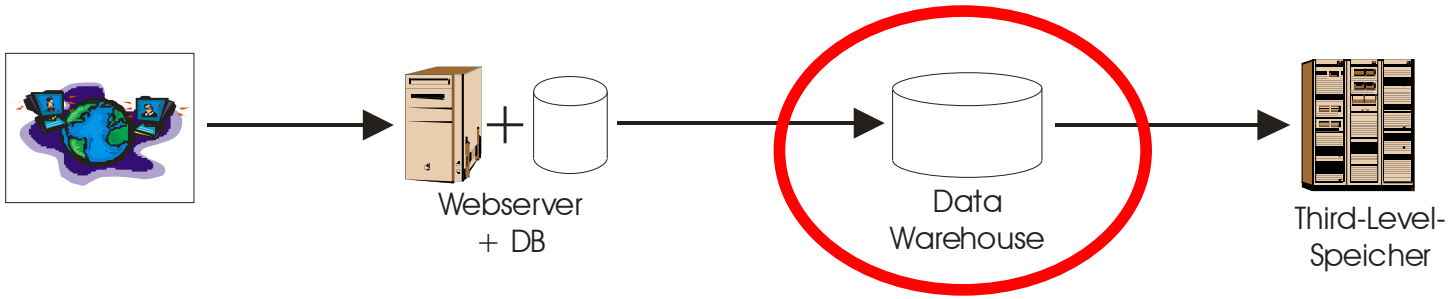
Web-Server mit Datenbank

- Aufgaben:
  - Bereitstellen der Web-Seiten,
  - Haltung der aktuellsten Benutzerdaten,
  - Herausfiltern unnötiger Daten aus „Click-Stream“,
  - Speicherung der Daten für ~24 h,
  - Priorisierung auf Durchsatz.



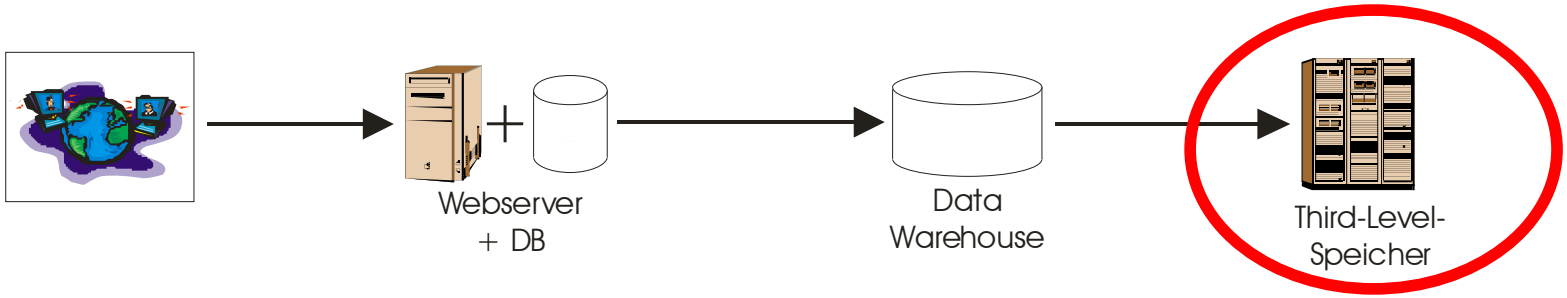
Data-Warehouse

- Aufgaben:
  - Haltung der Daten mit **mittlerer** Zugriffswahrscheinlichkeit,
  - Analyse der Daten für Marktforschungen,
  - Speicherung der Daten bis zu 12 Monaten.



Third-Level-Speicher

- Aufgaben:
  - Haltung der Daten mit **geringer** Zugriffswahrscheinlichkeit,
  - Verwendung der Daten für Langzeitanalysen,
  - Speicherung über Jahre bzw. Jahrzehnte.



Gliederung

- **Web-Intelligent-Systeme**
  - Begriffsklärung
  - Personalisiertes Web
  - Infrastruktur und Datenhaltung
- **Verfahren zur Ähnlichkeitssuche**
  - Algorithmen
  - Beispiele
- **Prefetching**
  - Begriffsklärung
  - Prefetching-Methoden
- **Zusammenfassung**

Algorithmen

- Erzeugte „Click-Stream“-Daten bestehen fast ausschließlich aus Textdaten.

  ➔ Verwendung von Methoden aus Text Retrieval!

- Einteilung der Algorithmen in 4 Klassen:
  - Kollaborative Filter,
  - Cluster-Verfahren,
  - Suchbasierte Verfahren,
  - Item-to-Item Collaborative Filtering.

Kollaborative Filter (1)

- Darstellung der Benutzerdaten als N-dim Vektor,
- Vektoreinträge enthalten positive / negative Produktbewertungen des Benutzer.

- Annahme für Ähnlichkeitssuche:
  - Ähnliche Benutzerinteressen werden durch ähnliche Produktbewertungen wiedergegeben.
    - ➔ Ähnlichkeitsvergleich der Benutzerbewertungen!

Kollaborative Filter (2)

- Bewertung:

  + Liefert sehr gute Ergebnisse!

    (Verwendung des Cosinus-Maß)

  - Vergleich sehr teuer!

    (Benutzer muss mit allen im System vorhandenen Benutzer verglichen werden.)

    ➔ Nicht geeignet bei vielen Benutzern!

Cluster-Verfahren

- Verfahren ähnlich dem Kollaborativen Filtern.

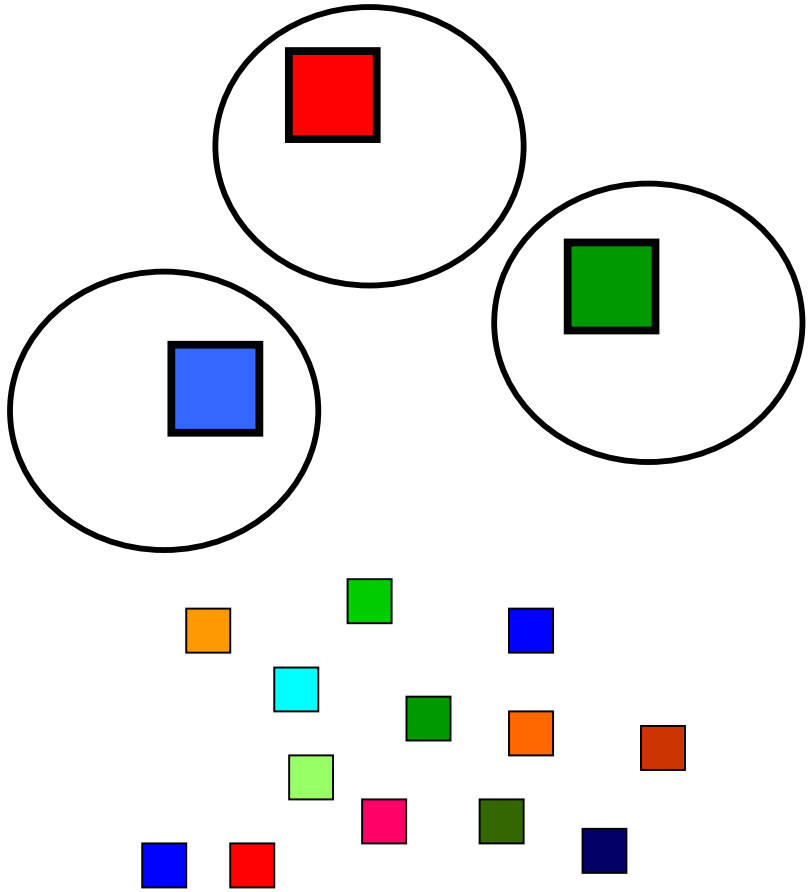
  Aber:

  - Gruppierung von ähnlichen Benutzern zu Clustern.
  - **Suche:**
    - Vergleich des Benutzers mit Repräsentanten jedes Clusters.
    - Suche nach ähnlichstem Benutzer im Cluster mit ähnlichsten Repräsentanten fortsetzen.

Cluster-Verfahren (Beispiel)

Vorbereitungsphase:

1. Wähle für jeden Cluster einen Repräsentanten (manuell).

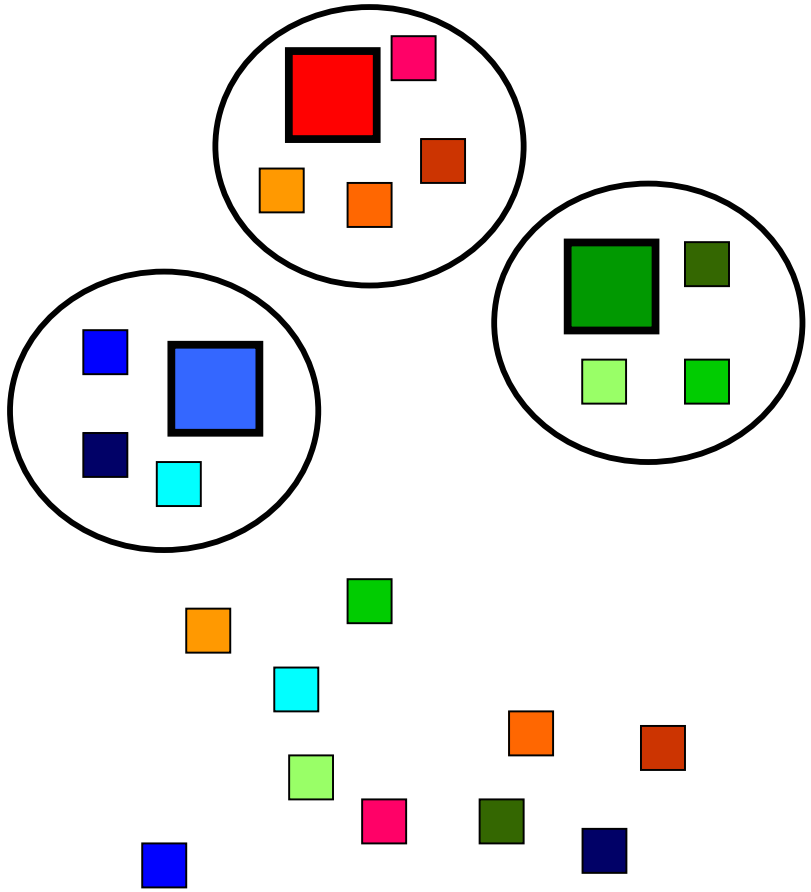


Cluster-Verfahren (Beispiel)

Vorbereitungsphase:

1. Wähle für jeden Cluster einen Repräsentanten (manuell).
2. Ordne restliche Benutzer dem ähnlichsten Repräsentanten zu.

➔ Cluster



Cluster-Verfahren (Beispiel)

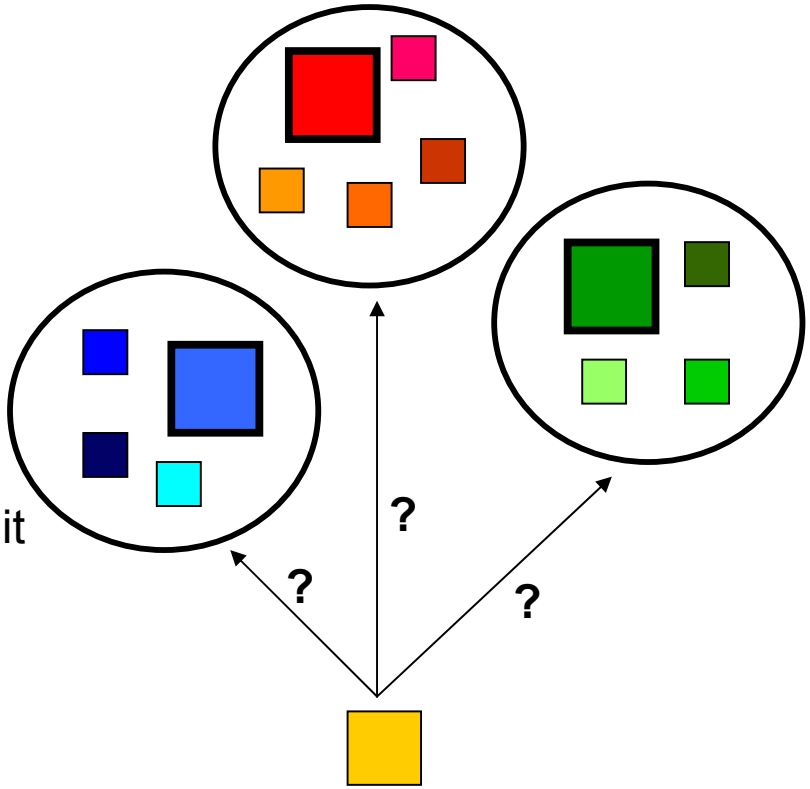
Vorbereitungsphase:

1. Wähle für jeden Cluster einen Repräsentanten (manuell).
2. Ordne restliche Benutzer dem ähnlichsten Repräsentanten zu.

➔ Cluster

Laufzeitphase:

1. Vergleiche aktuellen Benutzer mit Cluster-Repräsentanten.



Cluster-Verfahren (Beispiel)

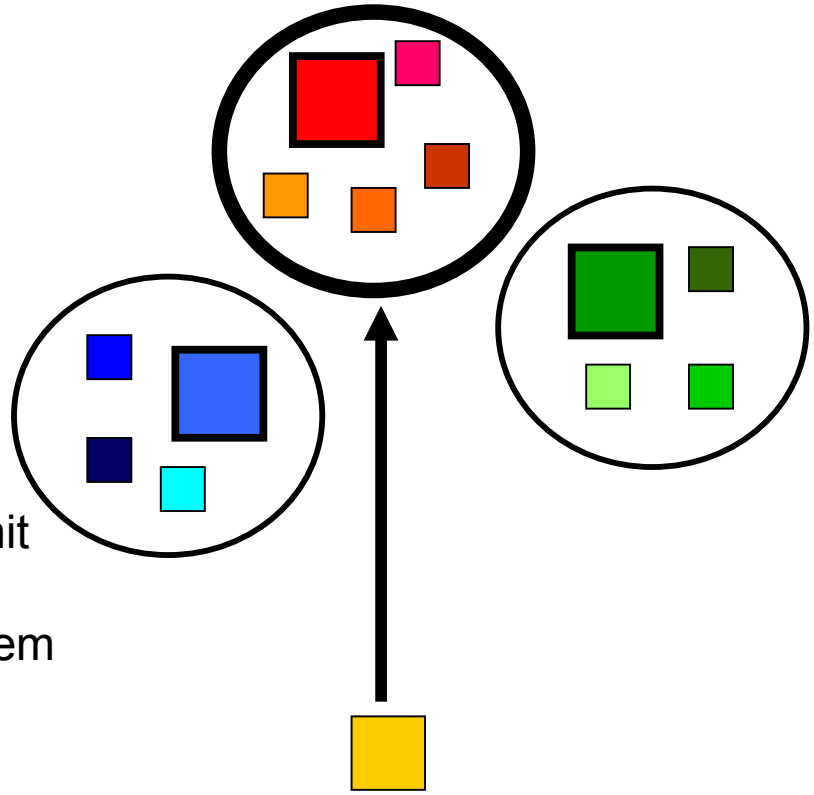
Vorbereitungsphase:

1. Wähle für jeden Cluster einen Repräsentanten (manuell).
2. Ordne restliche Benutzer dem ähnlichsten Repräsentanten zu.

➔ Cluster

Laufzeitphase:

1. Vergleiche aktuellen Benutzer mit Cluster-Repräsentanten.
2. Wähle den Cluster mit ähnlichstem Repräsentanten aus.



Cluster-Verfahren (Beispiel)

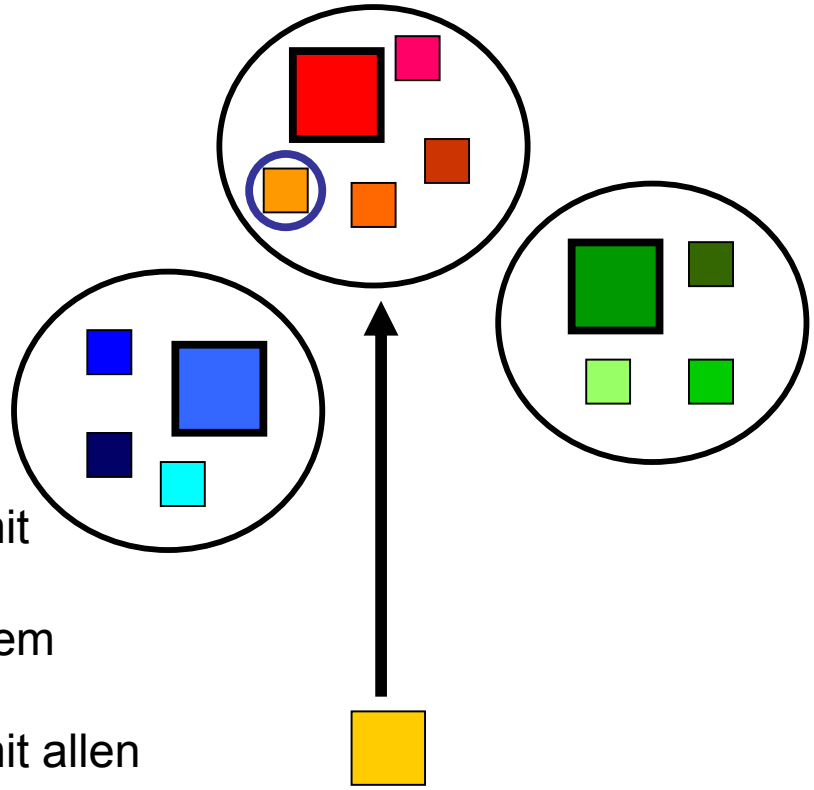
Vorbereitungsphase:

1. Wähle für jeden Cluster einen Repräsentanten (manuell).
2. Ordne restliche Benutzer dem ähnlichsten Repräsentanten zu.

➔ Cluster

Laufzeitphase:

1. Vergleiche aktuellen Benutzer mit Cluster-Repräsentanten.
2. Wähle den Cluster mit ähnlichstem Repräsentanten aus.
3. Vergleiche aktuellen Benutzer mit allen Benutzern im gewählten Cluster.



Cluster-Verfahren

- Bewertung:

  + Berechnung der Cluster nicht zur Laufzeit!

  + Weniger Vergleiche notwendig!

  - Qualität der Ergebnisse schlechter als bei Kollaborativen Filtern!

    Grund: Benutzerprofil kann in einen Cluster gelangen der nicht den ähnlichsten Benutzer enthält.

Suchbasierte Verfahren (1)

- Untersuchung der vom Benutzer gekauften Produkte,
- Suchanfrage mit Schlüsselworten aus den Produkteigenschaften,
  (z. B. Autor, Darsteller, Genre …)
- Suchanfrage kann einer SQL-Anfrage auf DB entsprechen.

Suchbasierte Verfahren (2)

- Bewertung:
  - + Sehr gute Ergebnisse bei wenigen gekauften Produkten!
  - - Anfrageergebnis sehr groß, bei vielen gekauften Produkten und disjunktiver Verknüpfung der Schlüsselworte!

Item-to-Item Collaborative Filtering

- Weiterentwicklung des suchbasierten Verfahrens,
- Matrix mit Ähnlichkeitswerten für jedes Produkt,
- Matrixberechnung geschieht nicht zur Laufzeit.

- Zur Laufzeit: Für gekaufte Produkte des Benutzers die n-ähnlichsten Produkte aus entsprechenden Matrizen auswählen.

Item-to-Item Collaborative Filtering

- Beispiel

Ähnlichkeit zu Produkt X	
A	83%
B	70%
C	68%

Ähnlichkeit zu Produkt Y	
D	81%
E	74%
B	66%

Ähnlichkeit zu Produkt Z	
C	72%
F	47%
G	32%

A **D**

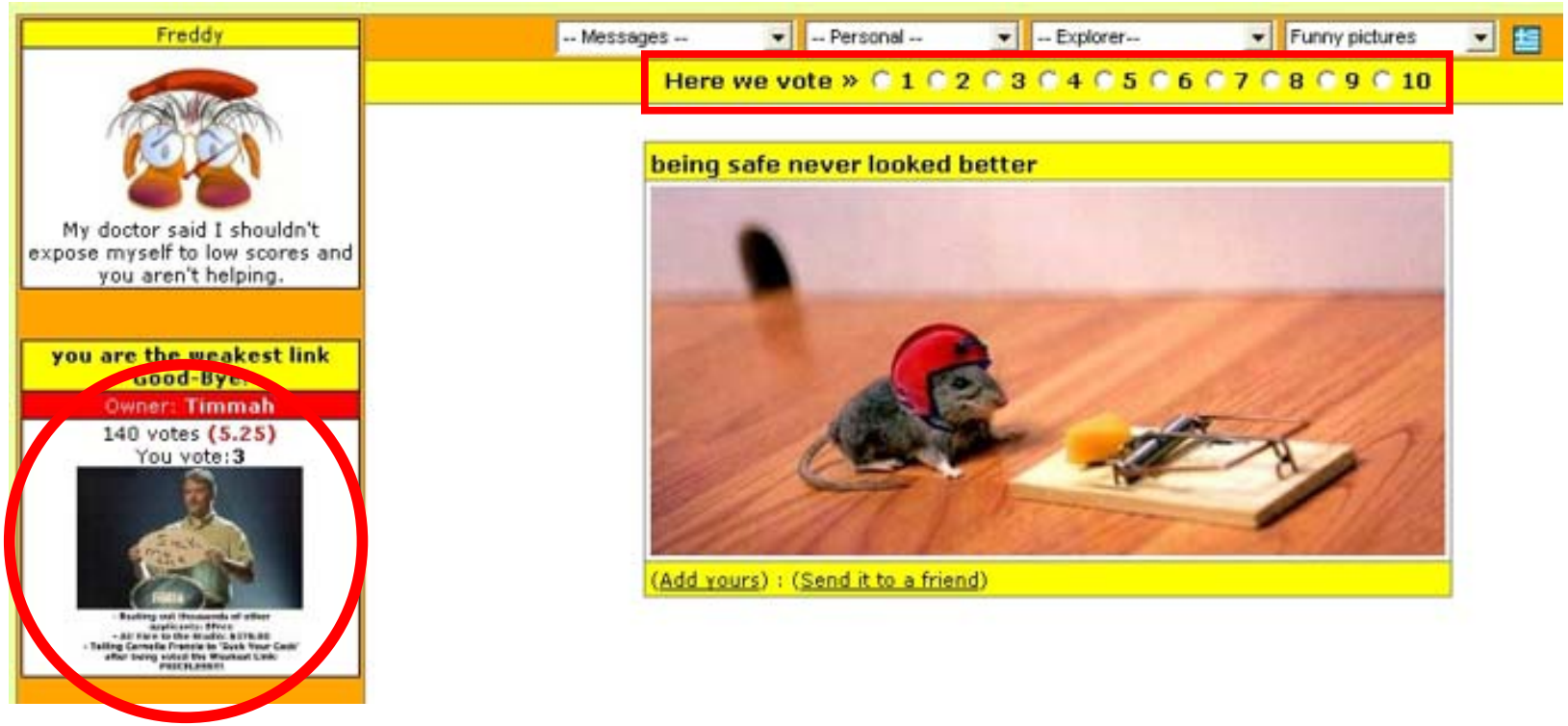
Benutzer	Gekaufte Produkte	Interessante Produkte
Fritz Meier	X, Y	**A, D**

Item-to-Item Collaborative Filtering

- Bewertung:
  - + Verfahren ist zur Laufzeit sehr schnell!
  - + Liefert sehr gute Ergebnisse!
  - + Auch bei großen Produktkatalogen effizient! (Anzahl Produkte >1.000.000)
  - - Matrixberechnung sehr aufwändig und speicherplatzintensiv!
    - ➔ **Aber:** Berechnung nicht zur Laufzeit!

Beispiele

- myFreddy.com (www.myFreddy.com)
  - Testplattform für Algorithmen,
  - Gegründet, um Testdaten zu gewinnen.



Beispiele

- PalmAgent
  - Führungssystem für Touristen,
  - Basierend auf PDA‘s,
  - PDA besitzt Agent, der Benutzerinteressen kennt,
  - Neue Daten (z. B. Veranstaltungen) durch Funk oder durch automatischen Austausch mit anderen Agenten,
  - Bewertung der Informationen nach Benutzervorlieben durch den Agenten,
  - Ähnlich dem Nexus-Projekt. (http://www.nexus.uni-stuttgart.de/)

Gliederung

- **Web-Intelligent-Systeme**
  - Begriffsklärung
  - Personalisiertes Web
  - Infrastruktur und Datenhaltung
- **Verfahren zur Ähnlichkeitssuche**
  - Algorithmen
  - Beispiele
- **Prefetching**
  - Begriffsklärung
  - Prefetching-Methoden
- **Zusammenfassung**

Prefetching

- Ziele:
  - Optimale Ausnutzung der Bandbreite zwischen Client und Server,
  - Wartezeit für den Benutzer verkürzen,
  - Reduzierung der Latenzzeit.

- Durchführungsmöglichkeiten:
  - Verhalten des Benutzers statistisch schätzen,
  - Zeitspanne zwischen 2 Aufrufen verwenden.

Prefetching

- Prefetching Methoden
  - Client-basiertes Prefetching,
  - Proxy-basiertes Prefetching,
  - Server-basiertes Prefetching,
  - Kooperatives Prefetching.

Client-basiertes Prefetching

- Analyse des Benutzerverhaltens.

  ➔ Gewohnheiten des Benutzers sehr gut ableitbar!

- Unterteilung in 2 Klassen:
  - Greedy: Alle Links auf einer Seite werden vorgeladen.

    ➔ viele unnötige Daten!
  - Non-Greedy: Aufrufhäufigkeiten werden berücksichtigt.

    ➔ häufig besuchte Seiten als interessanter einstufen!

Proxy-basiertes Prefetching

- Proxy verwaltet normalerweise eine ganze Domäne.

  ➔ Aufrufe von mehreren Benutzern sind bekannt.

- Aufrufreihenfolgen miteinander vergleichen.

  ➔ Benutzerverhalten auf Seiten vorhersagbar.

- Vergleich neuer Aufrufe mit bekannten Aufrufreihenfolgen und Folgeseiten vorladen.
- Komprimierung der Daten, um Übertragungszeit zu sparen.

Server-basiertes Prefetching

- Server besitzt größere Historie als Proxy und Client.
- Wissen über das Verhalten vieler Benutzer.

- Zwei Strategien
  - Push
    - Übertragung der z. B. 10 am häufigsten angeforderten Seiten bei jeder Anfrage.
  - Pull
    - Übergabe einer Liste der häufigsten angeforderten Seiten,
    - Auswahl von „interessanten“ Seiten durch Proxy / Client,
    - Auswahl an Server melden.

Kooperatives Prefetching

- Prefetching auf allen Ebenen.
- Kombination der Vorteile der 3 Verfahren.
- Server können interessante Seiten an Proxy schicken.
- Proxy kann Interessen der Benutzer an Server schicken.
- Clients können Verlaufsdaten an Proxy schicken.

  ➔ Hohe Kommunikation auf allen Ebenen.

  ➔ Erweiterung um Web-Intelligent-Komponenten möglich.

  ➔ Web-Intelligent-System.

Zusammenfassung

- Web-Intelligence
  - Begriff, Aufgaben, Einsatzgebiete (z. B. Portale),
  - Personalisiertes Web,
  - Infrastruktur (3-stufige Speicherhierarchie).
- Algorithmen zum Vergleich von Profilen
  - Kollaborative Filter,
  - Cluster-Verfahren,
  - Suchbasierte Verfahren,
  - Item-to-Item Collaborative Filtering.
- Prefetching
  - Client-, Proxy-, Server-basiertes Prefetching,
  - Kooperatives Prefetching.