

Recommendation Systeme

Gerald Kamper, Ludwig Meyer, Martin Reichart

Abstract:

Die BesucherInnen einer Konferenz bilden ein soziales Netzwerk: Manche Personen stehen in Verbindung, da sie dieselben Vorträge besucht haben. Dieses Paper stellt einen Algorithmus, sowie eine Implementierung vor, mit dessen Hilfe ausgehend von der Datenbasis der bestehenden Anmeldungen der Konferenz, Vorschläge für die Teilnahme an weiteren Vorträgen ermittelt werden können.

Zusätzlich soll verifiziert werden, ob mit Hilfe einer Analysesoftware für soziale Netzwerke visuell durch die Graphendarstellung auf die Ergebnisse mit dem im ersten Teil vorgestellten Verfahren geschlossen werden kann. Welche Gemeinsamkeiten haben Recommendation-Systeme und Werkzeuge der sozialen Netzwerkanalyse?

1 Zielsetzung

Ziel unserer Gruppe ist es, ein *Recommendation-System* zu erstellen, das BesucherInnen einer Konferenz Vorträge vorschlägt, die sie interessieren könnten. Dies soll erreicht werden, indem im ersten Schritt Personen mit ähnlichen Interessen gesucht werden. Im zweiten Schritt werden dann Vorschläge aufgrund der besuchten Vorträge dieser ähnlichen Personen erstellt.

Die grundsätzliche Frage bei Systemen wie diesen ist immer: Was ist *Ähnlichkeit* und wie kann man sie messen? Diese Frage ist sehr aktuell, wie man vor allem am Hype rund um das so genannte *Web 2.0* sieht. So verwendet Amazon einen solchen Algorithmus um BenutzerInnen Bücher vorzuschlagen, die sie interessieren könnten und Last.fm nutzt einen ähnlichen Mechanismus um NutzerInnen Songs vorzuschlagen, die ihnen gefallen könnten.

Wir werden im Zuge dieser Arbeit ein Programm entwickeln, mit dem es uns möglich ist, die Eingabedaten zu analysieren und aus diesen für jede Person die x ähnlichsten Personen zu ermitteln und jeder Person aufgrund ihrer eigenen Eigenschaften y Recommendations zu geben (x und y können beliebig gewählt werden). Im Gegenzug dazu verwenden wir in unserer Arbeit auch *Soziale Netzwerkanalyse-Tools* zur Analyse der Eingabedaten. Hauptsächlich verwenden wir hierzu das frei verfügbare Programm *Visone* [1, Visone]. Mit diesen Software-Tools ist es uns möglich, die Zusammenhänge der Eingabedaten visuell aufzubereiten und aus diesen je nach gewählter Analyseart unterschiedliche Gruppierungen und Gewichtungen abzulesen.

Ein weiteres Ziel ist, die Informationen, die wir mit unserem selbstentwickelten Programm aus den Eingabedaten gewinnen können, mit den Informationen, die wir aus den visuell aufbereiteten Daten ermitteln können, miteinander zu vergleichen. Aufgrund der Größe der Eingabedaten ist es nur möglich, strichprobenartige Vergleiche durchzuführen. Bei diesem Vergleich geht es uns darum, herauszufinden, ob wir die Informationen, die unser selbstentwickeltes Programm in sehr klarer Form liefern kann (zB.: Recommendation 1, 2, 3, 4, 5) auch aus dem visuellen Resultat, das beispielsweise Visone liefert, ähnlich deutlich ablesbar sind. Weiters ist hier interessant, ob die Netzwerkgröße und -komplexität hierauf eine Auswirkung hat und wenn ja, ab welcher Netzwerkgröße und -komplexität es möglich beziehungsweise unmöglich wird, die Informationen direkt aus den visuellen Resultaten abzulesen. Um dies herauszufinden, verändern wir die Größe unserer Eingabedaten so, dass die entstehenden visuellen Resultate unterschiedlich komplex sind und führen für jede der Eingabedateninstanzen getrennte Analysen durch.

Interessant ist außerdem die Frage, ob in einem der beiden Ergebnisarten (also den Ergebnissen unseres selbst entwickelten Programms bzw. den visuellen Ergebnissen von Sozialen Netzwerkanalyse-Tools) mehr oder gänzlich andere Informationen erkennbar sind, als in der jeweils anderen Ergebnisart.

2 Einführung

2.1 Theoretische Grundlagen: Recommendation Systeme und Machine Learning

Wie verhalten wir uns in einer Situation, in der wir viele Möglichkeiten und wenig Information haben? Wenn wir zum Beispiel in einer Videothek oder einem Büchergeschäft stehen und uns überfordert fühlen, zwischen den hunderten Möglichkeiten zu wählen? Oft fragen wir in einer solchen Situation eine Freundin oder einen Freund mit ähnlichem Geschmack nach einer Empfehlung.

Recommendation Systeme arbeiten genauso, nur dass man nicht mehr auf Leute beschränkt ist, die man persönlich kennt. Als Nachteil eines solchen Systems könnte man anführen, dass Unterhaltung und persönliche Kontakte in so einem System wegfallen. Wie wir aber sehen werden, ermöglicht unser System es, neue Leute kennen zu lernen und neue Kontakte zu knüpfen. Es soll also nicht den persönlichen Kontakt zu anderen Menschen ersetzen, sondern unterstützen.

Die Definition von Recommendation Systemen ist sehr allgemein gehalten: *Recommendation systems make suggestions about artifacts to a user* [4, Basu et. al. 1998]. Man kann zwischen zwei unterschiedlichen Arten unterscheiden:

1. Collaborative Systeme
2. Content-based Systeme

Bei unserem Projekt handelt es sich um einem *Collaborative Recommendation System* bzw. einer *Social-filtering Method* [5, Hill et. al. 1995], [7, Resnick et al. 1994], [8, Sharanand & Maes 1995]. Bei dieser Methode werden Bewertungen von anderen Menschen benutzt um die Recommendations zu erstellen.

Im Unterschied dazu gibt es noch *Content-based Recommendation Systems* [3, Balabanovic 1997], die Recommendations aufgrund einer Analyse des Inhalts, anstatt Bewertungen von BenutzerInnen für die Erstellung heranziehen. Beispielsweise können die Häufigkeiten von bestimmten Schlüsselwörtern in Texten benutzt werden um Ähnlichkeiten festzustellen. In letzter Zeit werden auch *Hybride Systeme* populär, die die beiden Ansätze kombinieren [2, Adomavicius & Tuzhilin, 2001].

Obwohl die Forschungsrichtung *Collaborative Recommendation Systems* schon Mitte der 90er Jahre entstand, ist das Thema noch immer von hohem akademischen und kommerziellen Interesse da diese Systeme ein nützliches Hilfsmittel darstellen, um mit der Informationsflut des Internets besser umzugehen.

Vor allem im Zusammenhang mit dem Boom um e-Commerce Systeme hat das Thema an Bedeutung gewonnen. Die Gründe dafür liegen auf der Hand: In einem wirklichen Geschäftslokal sieht man alle verfügbaren Produkte. Dabei stößt man eventuell auch auf Produkte, nach denen man ursprünglich nicht explizit gesucht hat. Bei e-Commerce Systemen hingegen sind die Präsentationsmöglichkeiten, schon aufgrund des kleinen Bild-

schirms, stark eingeschränkt - obwohl es üblicherweise viel mehr verfügbare Angebote als in einem realen Geschäft gibt. Für Kunden, die schon wissen, was sie suchen, stehen gute Suchfunktionen zur Verfügung. Wenn man aber nicht nach etwas Bestimmtem sucht, können Recommendation Systeme eine große Hilfe darstellen. Recommendations sind in diesem Zusammenhang also eine Form von Marketing [6, Mild et. al 2001].

Collaborative Recommendation Systeme arbeiten normalerweise in zwei Schritten:

1. Identifizierung eines *Nearest Neighbors*, also einer Person, die ähnlich ist
2. etwas vorschlagen, das diese Person hoch bewertet hat

Formal betrachtet handelt es sich bei dieser Aufgabe um ein *Learning Problem*: Das System muss eine Funktion (Utility Function) lernen, die allen möglichen Eingabewerten einen Ausgabewert zuordnet (Prediction). In der *Artificial Intelligence* existiert eine eigene Forschungsrichtung, die sich mit diesem Problem beschäftigt: *Machine Learning*.

Maschinelles Lernen ist ein Oberbegriff für die *künstliche* Generierung von Wissen aus Erfahrung: Ein künstliches System lernt aus Beispielen und kann nach Beendigung der Lernphase verallgemeinern. Das bedeutet, es lernt nicht einfach die Beispiele auswendig, sondern es *erkennt* Gesetzmäßigkeiten in den Lerndaten. So kann das System auch unbekannte Daten beurteilen [9, Wikipedia].

2.2 Soziale Netzwerk Analyse

Der Begriff des Sozialen Netzwerks kommt aus der Soziologie und beschreibt soziale Interaktionen beliebiger Art. Der Forschungsbereich der Sozialen Netzwerkanalyse versucht mit Hilfe der Methoden und Algorithmen wie sie aus der Graphen-Theorie und Netzwerk-Theorie bekannt sind, Beziehungen und Gesetzmäßigkeiten zu erkennen. Hierzu werden die Verhältnisse zwischen Entitäten, seien es nun einzelne menschliche (oder andere) Individuen, Gruppen, oder größere soziale Organisationsformen, visuell als Diagramm dargestellt.

Die Entitäten werden als Knoten dargestellt, während die untersuchten Relationen als Kanten (gerichtet, oder ungerichtet) zwischen den Knoten dargestellt werden. Die Relationen können auch mit Werten versehen werden → gewichteter Graph.

Wenn die betrachteten Knoten dieselbe Art haben, so spricht man von 1-mode Netzwerken. Sind sie jedoch von unterschiedlichen Arten (zwei, oder mehr), so nennt man diese 2- oder n-mode Netzwerke. In Kapitel 4 untersuchen wir mit Hilfe eines 1-mode Netzwerks die Beziehungen zwischen den Teilnehmern einer Konferenz, während wir mit Hilfe eines 2-mode Netzwerks die Teilnehmer in Beziehung mit den besuchten Konferenzbeiträgen analysieren.

3 Anwendungsfall

3.1 Unsere Ausgangsdaten

Die Daten, die wir für unsere Analyse verwendet haben, stammen von einer in Frankfurt stattgefundenen Konferenz, auf welcher insgesamt 53 Vorträge angeboten wurden. Nachdem alle Vorträge der Konferenz abgehalten wurden, wurden 98 der Teilnehmer bezüglich ihrer besuchten Vorträge und ihrem Interesse für Vorträge befragt. Das heißt es wurde erhoben, welche Vorträge sie besucht, und welche der Vorträge sie im Vorfeld interessiert haben; unabhängig davon, ob die Vorträge anschließend tatsächlich besucht wurden oder nicht.

Für unsere Analyse haben wir uns auf die erhobenen Daten der besuchten Vorträge der Teilnehmer konzentriert.

3.2 Use Cases

Auf Basis unserer Ausgangsdaten und unserer selbst entwickelten Analysesoftware haben wir zwei wesentliche *Use Cases* erarbeitet, welche im Folgenden beschrieben werden. Beide Use Cases sind zur Anwendung bei großen Konferenzen gedacht, bei denen vielen Besuchern eine große Anzahl an Vorträgen angeboten werden. Wir haben hier Use Cases für eine Anwendung sowohl im Vorfeld einer Konferenz als auch nach Abschluss beziehungsweise in der Endphase einer Konferenz erarbeitet. Diese Use Cases sind zum Teil durch die Beschaffenheit unserer Ausgangsdaten inspiriert, können allerdings auch für andere Anwendungsfälle als Konferenzen abgeändert werden und sollen nur die prinzipiellen Möglichkeiten unseres Recommendation Systems aufzeigen.

3.2.1 Vortragsempfehlung für Teilnehmer der Konferenz

Dieses Szenario ist gedacht, um im Vorfeld einer Konferenz den Teilnehmern eine *Empfehlung* für Vorträge zu geben, die für sie ebenfalls interessant sein könnten. Für dieses Szenario ist es wichtig, dass die Konferenzteilnehmer bereits einige Zeit vor der Konferenz eine Liste der Vorträge, die sie interessieren, an die Konferenzorganisation schicken. Außerdem ist es erst nach einer gewissen *Anzahl* von Konferenzteilnehmern, die dieses Informationen bekanntgegeben haben, möglich, Empfehlungen zu ermitteln, da davor die Eingabedaten viel zu gering sind. Die besten Empfehlungen können natürlich ermittelt werden, nachdem alle Konferenzteilnehmer die für sie interessanten Vorträge bekanntgegeben haben. Der Ablauf des Szenarios ist folgendermaßen:

1. Die Teilnehmer bekommen eine Liste der angebotenen Vorträge mit einer detaillierten Beschreibung des Inhalts der Vorträge und den genauen Zeitpunkten, zu denen die Vorträge stattfinden.
2. Jeder Teilnehmer wählt für sich Vorträge aus, die ihn aufgrund des Titels und der Beschreibung interessieren.

3. Jeder Teilnehmer gibt seine persönliche Auswahl an Vorträgen wieder an die Konferenzorganisation zurück.
4. Von der Konferenzorganisation wird für jeden Teilnehmer aufgrund seiner persönlichen Auswahl an Vorträgen eine Empfehlung für weitere Vorträge erstellt. Diese Empfehlung wird mit Hilfe unseres Recommendation Systems erstellt, das hierfür die Auswahlen aller Teilnehmer analysiert und dadurch für jeden Teilnehmer die Vorträge ermittelt, die den Teilnehmer mit hoher Wahrscheinlichkeit ebenfalls interessieren können und die der Teilnehmer noch nicht in seiner persönlichen Auswahl hat.
5. Die mit Hilfe des Recommendation Systems erstellten Vortragsempfehlungen werden an die Teilnehmer retourniert.

3.2.2 Kontaktempfehlung für Teilnehmer

Dieses Szenario ist für das Ende einer Konferenz gedacht. Ziel ist es, jedem Teilnehmer *Kontaktpersonen* zu empfehlen, die ähnliche Interessen haben und mit denen es sinnvoll sein kann, Kontakt aufzunehmen. Der Ablauf des Szenarios ist folgendermaßen:

1. Die Teilnehmer besuchen auf der Konferenz eine Menge an Vorträgen.
2. Am Ende der Konferenz wird für jeden Teilnehmer ermittelt, welche Vorträge er besucht hat. Dies kann entweder ganz am Ende der Konferenz erfolgen oder zum Beispiel durch eine sofortige Registrierung, beispielsweise durch Teilnahmelisten direkt bei der Teilnahme an den Vorträgen.
3. Von der Konferenzorganisation wird für jeden Teilnehmer aufgrund seiner besuchten Vorträge eine Empfehlung für Personen mit ähnlichen Interessen erstellt. Diese Empfehlung wird mit Hilfe unseres Recommendation Systems erstellt, welches hierfür alle Vortragsteilnahmen der Konferenzteilnehmer analysiert und für jeden Teilnehmer die Personen ermittelt, mit denen der Teilnehmer die ähnlichsten Interessen hat (in diesem Fall die meisten gemeinsam besuchten Vorträge).
4. Die mit Hilfe des Recommendation Systems erstellten Kontaktempfehlungen werden an die Teilnehmer retourniert.

4 Programmativer Ansatz

Zusätzlich zum Vorschlagen von Vorträgen soll das Programm auch noch eine weitere Funktion bieten. Es soll möglich sein, KonferenzteilnehmerInnen zu finden, die *ähnliche Vorträge* besucht haben. Im Gegensatz zum Vorschlagen ähnlicher Vorträge, ist das Finden *ähnlicher Teilnehmer* erst nach der Konferenz interessant. BesucherInnen können damit Leute mit ähnlichen Interessen finden und sich mit ihnen unterhalten und Kontakte knüpfen.

4.1 Das Programm:

4.1.1 Vorteile unseres Systems:

Collaborative Recommendation Systeme haben oft bestimmte Schwachstellen, die unser System aber nicht aufweist:

1. Die Besucherinnen und Besucher müssen keine lästigen Fragebögen ausfüllen oder ähnliches, da die Informationen, die für die Recommendations benötigt werden, direkt aus der Anmeldung für die Vorträge gewonnen werden.
2. Sensible Daten werden anderen Besucherinnen und Besuchern nicht zugänglich gemacht. In einer endgültigen Version sollte man natürlich auswählen können, ob man nach der Konferenz von anderen Leuten kontaktiert werden können soll.
3. Collaborative Recommendation Systeme haben oft ein Problem mit neuen Usern (in unserem Fall KonferenzbesucherInnen), da zu Beginn noch keine Daten für deren Beurteilung zur Verfügung stehen. Unser System hat aber schon vor der ersten Benutzung die Daten der Konferenz der jeweiligen Benutzerin und liefert daher sofort gute Ergebnisse.
4. Ein weiteres verbreitetes Problem von Recommendation Systemen ist *Sparsity*. Das bedeutet, dass ein System schlechte Ergebnisse liefert, wenn der Möglichkeitsraum sehr groß ist und es gleichzeitig nicht sehr viele Daten gibt. Unser System hat dieses Problem im Normalfall nicht, da die Anzahl der Vorträge immer verhältnismäßig gering ist und mit der Anzahl der Möglichkeiten (besuchbare Vorträge) immer auch die Anzahl der zur Verfügung stehenden Daten (Anmeldungen für Vorträge) steigt.

4.1.2 Grundlagen des Programms

Schritt 1: Umwandlung der Daten:

Die Daten werden zeilenweise aus der CSV-Datei (Comma Separated Values) ausgelesen und für jede Person wird ein *Hash* mit allen besuchten Vorträgen angelegt.

Schritt 2: Ähnlichkeit zweier Personen errechnen:

Es gibt verschiedene Möglichkeiten Ähnlichkeit zu definieren. Eine Methode ist der *Euklidische Abstand*: Zwei Merkmalsausprägungen → x- und y-Achse → Abstand mit Pythagoras.

Wir verwenden den *Pearson-Korrelationskoeffizienten*. Dieser ist ein Maß für den Grad des linearen Zusammenhangs zwischen zwei Merkmalen. Es wird versucht eine gerade Linie mit minimalem Abstand zu allen Merkmalsausprägungen durch den *Möglichkeitsraum* zu legen.

1	vollständige (lineare) Übereinstimmung
0	keine Übereinstimmung
-1	vollständige negative Übereinstimmung

Tabelle 1: Der Koeffizient liegt immer zwischen 1 und -1

Schritt 3: Die x ähnlichsten Personen finden:

Es werden für die übergebene Person die Ähnlichkeiten zu allen anderen Personen errechnet. Die Personen werden nach Ähnlichkeit sortiert und die ersten x werden zurückgegeben.

Schritt 4: x Recommendations für eine Person finden:

Für jeden Vortrag wird eine nach Ähnlichkeiten gewichtete Summe gebildet. Da es in unserem Fall nur 1 (Vortrag besucht) und 0 (nicht besucht) gibt, werden also die *Korrelationskoeffizienten* (>0) der Personen zusammengezählt, die diesen Vortrag besucht haben. Danach werden die x Vorträge zurückgeliefert, die die höchste gewichtete Summe erreicht haben und von der Person noch nicht als besucht angegeben wurden.

5 Grafische Analyse mit Hilfe eines Soziale-Netzwerkanalyse-Tools

Der zweite Teil dieses Papers beschäftigt sich mit der Verifikation, ob die Ergebnisse des Recommendation-Algorithmus bereits aus den Ausgangsdaten mit Hilfe einer *Soziale-Netzwerkanalyse* Software wie beispielsweise *Visone* [1, Visone] ermittelt werden können.

Visone arbeitet mit Algorithmen zur Graphenanalyse. Die Basisdaten die analysiert werden sollen, können unter anderem als CSV (Comma Separated Values) oder als Excel-tabelle bereitgestellt werden.

In unserem Fall haben wir zwei Merkmalsausprägungen zu behandeln:

1. Teilnehmer einer Konferenz
2. Konferenzvorträge

	Vortrag 1	Vortrag 2	Vortrag 3	Vortrag 4
Teilnehmer 1	1	1	0	0
Teilnehmer 2	0	1	0	1
Teilnehmer 3	1	0	0	1
Teilnehmer 4	0	1	1	0

Tabelle 2: Beispielhafte Inputdaten für die SNA-Software

Die Null, beziehungsweise Eins, bezeichnet, ob der Teilnehmer den Vortrag der jeweiligen Spalte besucht hat, oder nicht.

Die simple Analyse der am häufigsten besuchten Konferenzbeiträge ist in *Abbildung 1* dargestellt. Die Knoten, die die Teilnehmer repräsentieren, sind farblich nach der Teilnahmehäufigkeit gekennzeichnet; je heller, desto mehr Konferenzbeiträge hat der Teilnehmer besucht. Außerdem korreliert die Größe der Beitragsknoten mit der Zahl der Teilnehmer, die den Vortrag besuchten. Zusätzlich wurden die Konferenzbeiträge nach ihrer Beliebtheit angeordnet. In der Darstellung sind die häufiger besuchten Beiträge weiter in der Mitte des Diagramms angeordnet, als jene die weniger beliebt waren. Analog sind die Teilnehmer nach der Anzahl ihrer besuchten Konferenzbeiträge angeordnet. So ist der Konferenzbeitrag *c02* der am häufigsten besuchte Vortrag. Er befindet sich in der Abbildung in der Mitte und ist der größte Knoten. Die Beiträge *c34* und *c38* wurden nur von wenigen Teilnehmern besucht und befinden sich in der Abbildung ganz oben am Rand. Ähnlich verhält es sich mit den Knoten der Teilnehmer: *T19* hat relativ viele Beiträge besucht (er befindet sich in der Abbildung auf „3 Uhr“), während *T83* (ganz unten in der Mitte) nur sehr wenige Vorträge besucht hat.

Der Vollständigkeit halber sollte erwähnt werden, dass zur Vereinfachung der Darstellung solche Konferenzteilnehmer aus der Analyse ausgenommen wurden, die keinen der angebotenen Konferenzbeiträge besucht haben.

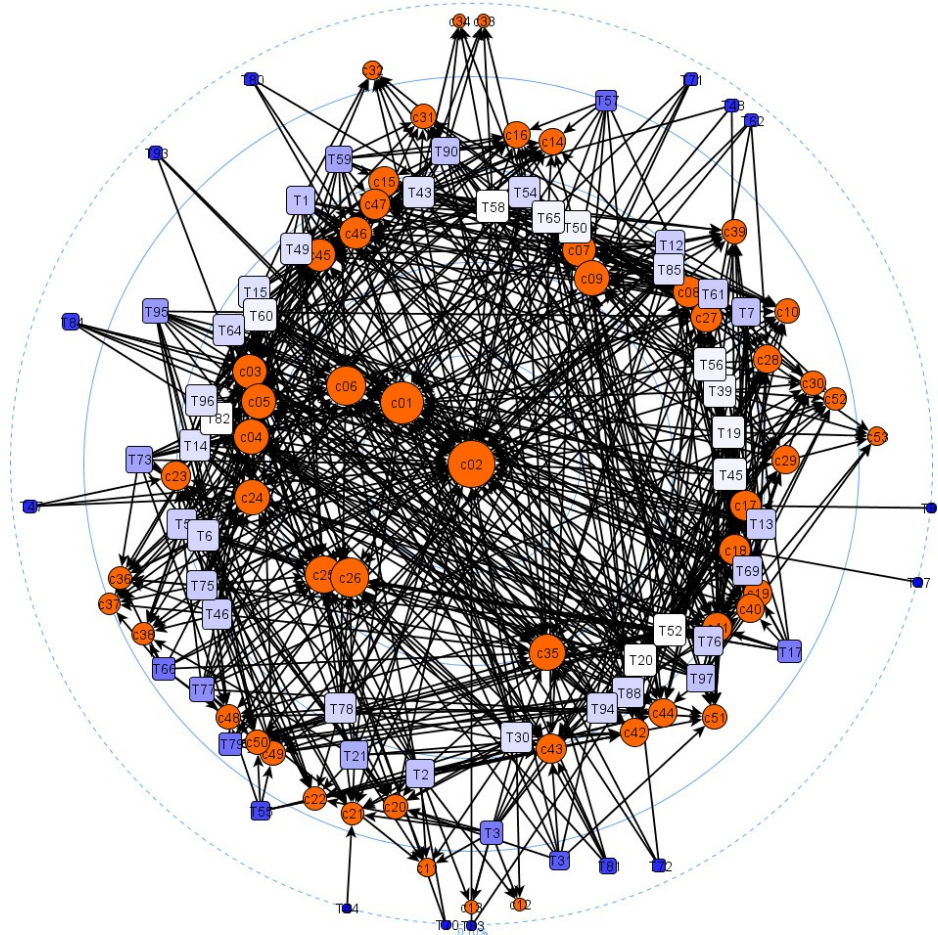


Abbildung 1: Erstes Analyse-Ergebnis mit Hilfe von Visone - Konferenzbeiträge sind in der Grafik als *cXX* und Teilnehmer als *TXX* gekennzeichnet.

In einer weiteren Analyse haben wir versucht, Empfehlungen von Konferenzteilnehmern mit ähnlichen Interessen visuell zu ermitteln, um diesen Teilnehmern am *Ende der Konferenz* die Herstellung von interessanten Kontakten zu erleichtern. Unser Ziel war es, aus der Visualisierung unserer Eingabedaten für jeden Teilnehmer ablesen zu können, welche Konferenzteilnehmer mit diesem die ähnlichsten Interessen besitzen. Die Positionierung der Teilnehmerknoten sollte unserer Vorstellung nach der folgenden Regel erfolgen:

Zwei Teilnehmer sind umso näher, je mehr gemeinsame Vorträge sie besucht haben.

Wie auch bei der vorangegangenen Analyse, haben wir wieder Visone verwendet, um eine Visualisierung zu erstellen. Unser erster Ansatz war, direkt unsere Eingabedaten zu benutzen, die bereits in einem von Visone lesbaren Format vorlagen und deren Struktur in Tabelle 2 zu sehen ist. Da die erste Visualisierung mit relativ geringer Schwierigkeit zu erstellen war, waren wir auch in diesem Fall sehr optimistisch, wurden allerdings recht schnell von Visone enttäuscht. Das größte Problem war, dass es mit Visone nicht möglich war, die individuell besuchten Vorträge eines Teilnehmers mit denen eines anderen zu vergleichen und diese Information dafür zu nutzen, um die beiden Teilnehmer in der Visualisierung zu positionieren. Stattdessen war die Positionierung der Teilnehmer eher willkürlich und hat nach manueller Kontrolle die Eingabedaten und auch die Ergebnisse, die wir mit unserer selbstentwickelten Software gewonnen haben, überhaupt nicht widerspiegelt.

Nach einigen gescheiterten Versuchen mit den unterschiedlichen Analyse- und auch Visualisierungsmöglichkeiten, die Visone bietet, haben wir uns schließlich entschlossen, das Format der Eingabedaten so zu verändern, dass sie durch Visone unserer Ansicht nach einfacher zu analysieren sein müssten.

	Teilnehmer 1	Teilnehmer 2	Teilnehmer 3	Teilnehmer 4
Teilnehmer 1	0	3	7	10
Teilnehmer 2	3	0	9	2
Teilnehmer 3	7	9	0	4
Teilnehmer 4	10	2	4	0

Tabelle 3: Beispielhafte Inputdaten für die SNA-Software nach Transformation

Im Wesentlichen ist dies eine *Adjazenzmatrix* der Teilnehmer, bei der in den jeweiligen Schnittpunkten, die Anzahl der gemeinsam besuchten Vorträge vermerkt ist. Also haben beispielsweise Teilnehmer 1 und Teilnehmer 2 gemeinsam drei Vorträge besucht.

Wir vermuteten, dass dieses Format der Eingabedaten die Arbeit für Visone wesentlich erleichtert, da die komplexe Analyse im Wesentlichen dadurch bereits erledigt ist und die Schwierigkeit nur mehr in der Positionierung der einzelnen Teilnehmer besteht. Doch auch hier erzielten wir mit Visone kein brauchbares Ergebnis. Die Visualisierungen wiesen dieselben Probleme auf, wie jene mit dem ursprünglichen Eingabeformat: Eine willkürliche Anordnung der Teilnehmer und keine Korrelation mit den aus unserer selbstgeschriebenen Software gewonnenen Erkenntnissen und nach manueller Kontrolle auch keine Widerspiegelung der Eingabedaten.

Unser letzter Versuch mittels SNA-Software eine brauchbare und den Eingabedaten entsprechende Visualisierung zu bekommen, war schließlich die Verwendung einer anderen SNA-Software - konkret von UCINET 6 [11, UCINET]. UCINET unterscheidet sich sowohl in der Bedienung als auch in der Visualisierung sehr stark von Visone und macht ganz allgemein einen wesentlichen komplexeren Eindruck.

Für die Analyse mit UCINET haben wir erneut unsere *transformierten* Eingabedaten verwendet (Tabelle 3). Als Analyseverfahren haben wir uns entschieden *Multi-Dimensional-Scaling* zu verwenden, da uns dies auch im Hinblick auf unsere gewünschte Visualisierung als die sinnvollste erschien. Zu unserer Überraschung hat es UCINET auf Anhieb geschafft, hier ein brauchbares Ergebnis zu liefern und nach einigen Verfeinerungen ist es uns gelungen, das in Abbildung 2 gezeigte Ergebnis zu erzeugen.

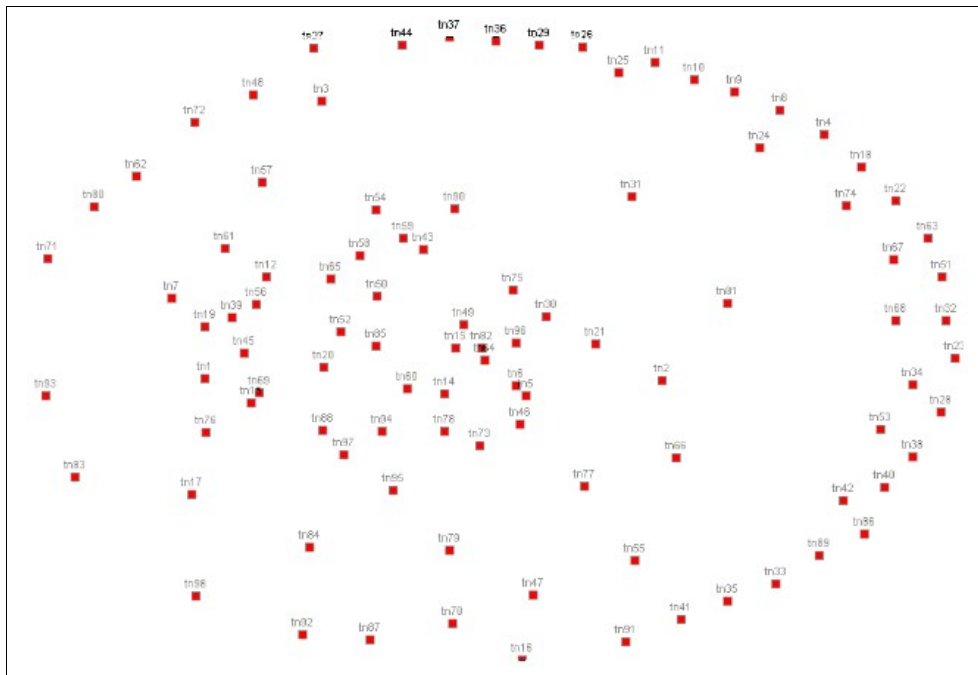


Abbildung 2: Ergebnis UCINET: Clusterbildungen bei Teilnehmern die gemeinsam Vorträge besucht haben. Je mehr gemeinsam besuchte Vorträge, desto näher beieinander befinden sich die Knoten.

Nachdem wir eine verwertbare Visualisierung durch UCINET erstellen konnten, war es uns schließlich möglich, auf eine unserer *Grundfragen* zurückzukommen: Ob die Informationen, die wir mit unserer selbstentwickelten Software gewinnen können, sich auch in der mittels SNA-Software erstellten Visualisierung abgelesen werden können. Zur Beantwortung dieser Frage haben wir stichprobenartig für einzelne Konferenzteilnehmer Empfehlungen der 5 ähnlichsten Teilnehmer erstellt und anschließend die Positionierung dieser in der Visualisierung festgestellt. Die Abbildungen 3 und 4 zeigen zwei dieser Stichproben, wobei der Konferenzteilnehmer, für den wir die Empfehlungen errechnet haben, *grün* eingekreist ist und die 5 Empfehlungen unserer selbstentwickelten Software *blau* gekennzeichnet sind.

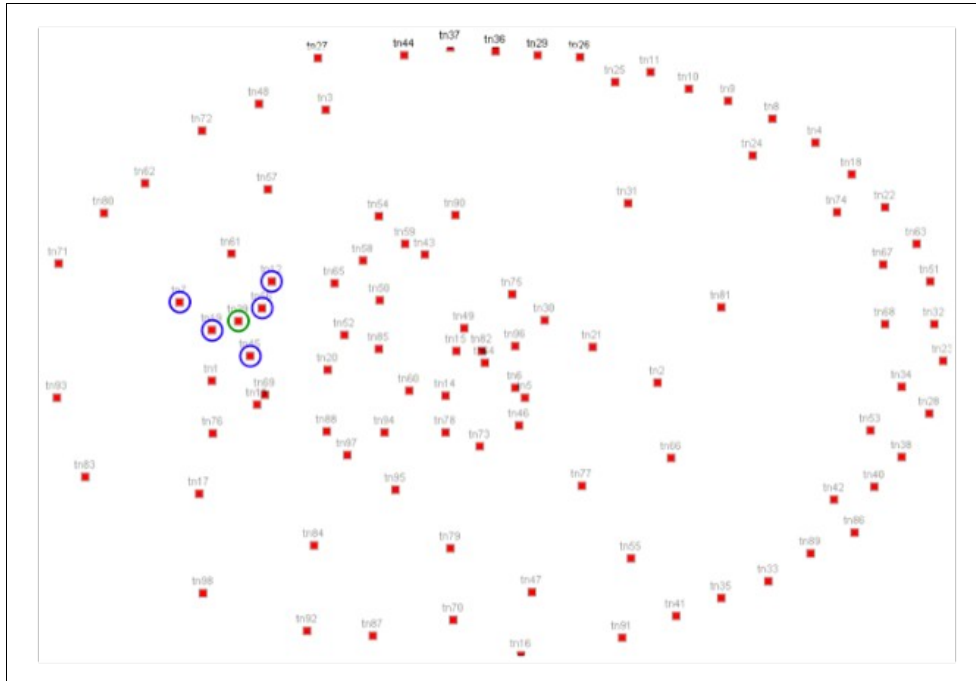


Abbildung 3: Ergebnis UCINET mit Kennzeichnung der ähnlichsten Teilnehmer (blau) für einen bestimmten Teilnehmer (grün)

Hier ist gut zu sehen, dass die SNA-Software in beiden Fällen eine sehr gute Positionierung der Teilnehmerknoten durchgeführt hat. In der ersten Abbildung stimmen die mit unserer selbst entwickelten Software ermittelten Empfehlungen zu 100% mit den nächsten Knoten des Teilnehmers überein. Auch in der zweiten Abbildung stimmen die Ergebnisse sehr gut überein und nur eine einzelne Empfehlung wurde von der SNA-Software weiter weg positioniert, als dies sein sollte.

Obwohl in den beiden gezeigten Abbildungen die Übereinstimmung der Visualisierung mit den errechneten Empfehlungen sehr hoch ist, gab es auch einzelne Fälle, in denen die Visualisierung nicht so klare Rückschlüsse ermöglichte. Unserer Meinung ist dies aber auch durch die Größe der Eingabedaten bedingt und der daraus resultierenden Schwierigkeit, so viele Teilnehmerknoten in zwei Dimensionen richtig zueinander zu positionieren. Wir vermuten, dass bei einem kleineren Eingabedatenset die Ergebnisse der Visualisierung noch besser mit den errechneten Empfehlungen übereingestimmt hätten und mit einem größeren Datenset umso schlechter werden würden. Eine genauere Analyse dieses Zusammenhangs übersteigt allerdings den Rahmen dieser Arbeit und wurde von uns deshalb im Zuge dieser nicht weiter verfolgt.

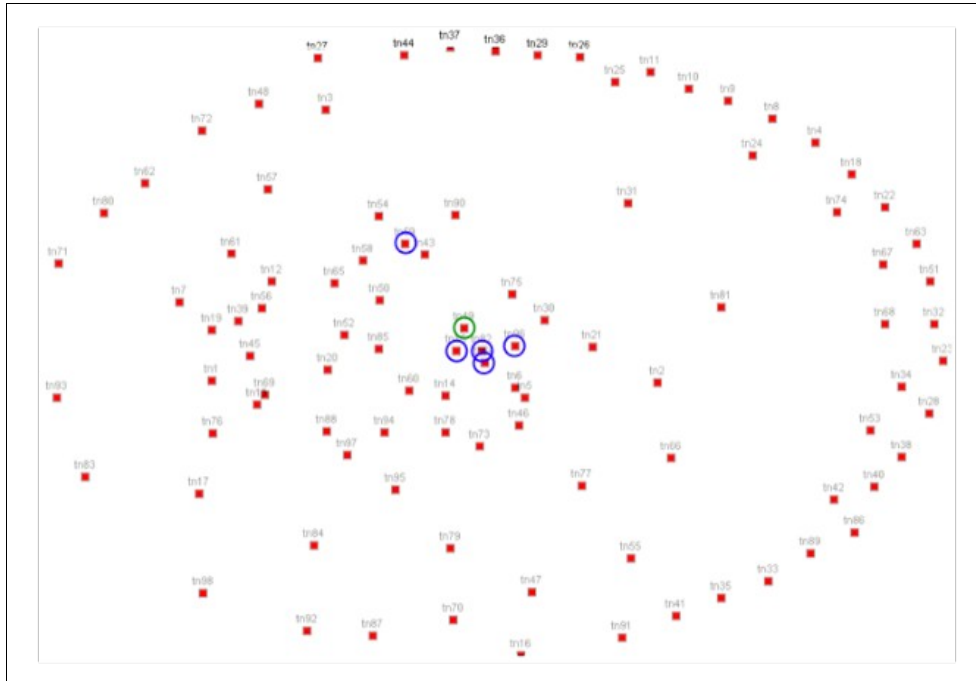


Abbildung 4: Weiteres Ergebnis UCINET mit Kennzeichnung der ähnlichsten Teilnehmer

6 Fazit

Im sogenannten Web 2.0 werden mit Hilfe von *Recommendation-Systemen* und *Machine-Learning-Algorithmen* versucht Ähnlichkeiten zu erkennen um den BenutzerInnen neue Möglichkeiten zu bieten. Unternehmen wie beispielsweise Amazon und Last.fm versuchen ihre KundInnen mit Vorschlägen zu weiteren Produkten aus ihren Katalogen unterstützen. In Zukunft werden sicherlich noch weitere Anwendungsmöglichkeiten und Dienste uns durch die Flut an Informationen und Angeboten leiten.

Wir haben in diesem Paper einen der grundlegenden Algorithmen für Recommendation-Systeme vorgestellt und in einer einfachen Implementierung vorgeführt. Am Beispiel einer Konferenz dienten die Vorträge und ihre Besucher als Datenbasis. Die Beziehungen zwischen den KonferenzteilnehmerInnen bestehen in der gemeinsamen Teilnahme an bestimmten Vorträgen. So kann das Recommendation-System sowohl bei der Anmeldung vor der Konferenz die Teilnehmer aufgrund ihrer Interessen (die sie vorher bekanntgegeben haben) weitere möglicherweise interessante Vorträge vorschlagen, als auch nach Ende der Konferenz andere Teilnehmer, die die selben Interessen, basierend auf den Besuchen der gleichen Vorträge haben, ermitteln. Letzteres könnte die Teilnehmer bei der Kontaktaufnahme unterstützen und zu weiteren Diskussionen und zum Ideenaustausch beitragen.

Im zweiten Teil des Papers haben wir untersucht, ob die Ergebnisse die der Algorithmus des Recommendation-Systems brachte, mit Hilfe eines Werkzeugs der *Sozialen Netzwerk-Analyse* ebenfalls bereits grafisch veranschaulicht werden können. Wir haben zwei Tools verwendet und nach ein paar kleineren Schwierigkeiten die Daten verifizieren können. So kann man anhand der Gruppierungen im Graphen die Teilnehmergruppen identifizieren die ähnliche Interessen besitzen, da sie teilweise die gleichen Vorträge besucht haben.

7 Quellenverweise

- [1] Visone: <http://visone.info>
- [2] Adomavicius & Tuzhilin 2001, *Recommendation Technologies: Survey of Current Methods and Possible Extensions*
- [3] Balabanovic 1997, *An Adaptive Web Page Recommendation Service*
- [4] Basu et. al. 1998, *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*
- [5] Hill et. al. 1995, *Recommending and Evaluating Choices in a Virtual Community of Use*
- [6] Mild et. al 2001, *A Critical View on Recommendation Systems*
- [7] Resnick et al. 1994, *GroupLens: An open architecture for collaborative filtering of net-news*
- [8] Shardanand & Maes 1995, *Social information filtering: Algorithms for automating 'word of mouth'*
- [9] Wikipedia: *Maschinelles Lernen* (http://de.wikipedia.org/wiki/Maschinelles_Lernen)
- [10] Ravi Kumar et. al. 1998, *Recommendation Systems: A probabilistic Analysis*
- [11] UCINet: <http://www.analytictech.com/ucinet/ucinet.htm>