Erzeugung zufälliger Graphen und Bayes-Netze

Proseminar „Algorithmen auf Graphen“

Georg Lukas, IF2000
2002-07-09

E-Mail: georg@op-co.de
Folien: http://op-co.de/bayes/

Gliederung

1. Einleitung
2. einfache Graphen-Konstruktion
3. Rekursiver Aufbau von Graphen
4. „artiger“ Einpass-Algorithmus
5. Bayes-Netze, Erzeugung zufälliger Attribute

Gründe zur Beschäftigung mit zufälligen Graphen

Einsatzgebiete:

- Modellierung von realen Netzen (z.B. Computer-Netze, Verkehrsnetze)
- Testen von Graphen-Algorithmen
  - Layouting von Graphen/Bäumen
  - Lernalgorithmen für Bayes-Netze

Einteilung der Graphen:

- echt zufällige Graphen (leicht erzeugbar)
- oft bestimmte Eigenschaften erforderlich
  - begrenzte maximale Cliquen
  - Verhältnis Kanten/Knoten (Dichte des Graphen)
  - zusammenhängend
  - Attribute (statistische/zufällige Daten)

Zufällige Adjazenzmatrix

Anschauliches Vorgehen:

1. leeren Graphen mit n Knoten erzeugen
2. für jede mögliche Kante entscheiden, ob sie gesetzt werden soll

Adjazenzmatrix erzeugen:

1. $n \times n$-Matrix erzeugen, mit Nullen füllen
2. für jedes Element der oberen Dreiecksmatrix (ohne Hauptdiagonale):
   - Zufallszahl $0 \leq x < 1$ erzeugen
   - Wenn $x < p$ Matrix-Element auf 1 setzen

- erzeugt echt zufällige Graphen
- Anzahl Kanten über die Wahrscheinlichkeit p beeinflussbar, jedoch nicht direkt
- nicht immer zusammenhängend
- keine feste Struktur (Eltern-Anzahl, Kreise)
- für Bayes-Netze bedingt brauchbar
- Komplexität $O(n^2)$

Kanten zufällig anordnen

- leeren Graphen mit n Knoten erzeugen
- Anzahl Kanten e zufällig aus $0..\frac{n(n-1)}{2}$ wählen
- für i von 1 bis e:
  - p und q zufällig aus $1..n$ wählen, so dass $p < q$ gilt
  - Gerichtete Kante von v_p nach v_q erzeugen

- explizites Setzen von $e \Rightarrow$ feste Kanten-Anzahl
- geeignete Wahl von p und $q \Rightarrow$ max. Eltern- und Kinderanzahl beeinflussbar
- nicht immer zusammenhängend
- für Bayes-Netze bedingt brauchbar
- Laufzeit von e abhängig
- Komplexität bis zu $O(n^2)$

Rekursive Erzeugung - Grundlagen

- Grundlage: Formel für die Anzahl möglicher Graphen von R.W. Robinson:
$$a_n^{(k)} = \sum_{s=1}^{n-k} \binom{n}{k} (2^k - 1)^s 2^{k(n-k-s)} a_{n-k}^{(s)}$$
- $a_n^{(k)}$: Anzahl der Graphen mit n Knoten (davon k Wurzeln)

- Wiederholte rekursive Zerlegung des Graphen in Wurzeln und Rest-Knoten
- Für alle möglichen Wurzel-Anzahlen s des Teilgraphen alle Kombinationen aufsummieren

Rekursive Erzeugung - Algorithmus

Umkehrung der Berechnungsformel zur Erzeugung eines Graphen (erfordert Knotenmenge V und Anzahl Wurzelknoten k):

1. Zufällig k Wurzeln wählen: $K \subseteq V$, $|K| = k$
2. Zufällig s aus $1..(n-k)$ wählen, dabei beachten:
$$P(s) = \binom{n}{k}(2^k - 1)^s 2^{k(n-k-s)} a_{n-k}^{(s)} / a_n^{(k)}$$
s für die Erzeugung von $S \subseteq V \setminus K$ im nächsten Rekursionsschritt benötigt
3. Wähle zufällig und unabhängig voneinander s nichtleere Teilmengen aus $K \Rightarrow$ Eltern für jedes Element aus S
4. Wähle zufällig und unabhängig voneinander $(n-k-s)$ Teilmengen (die auch leer sein dürfen) aus $K \Rightarrow$ Eltern für Elemente aus $V \setminus K \setminus S$
5. Benutze diesen Algorithmus rekursiv um aus $V \setminus K$ als Knotenmenge und s als Anzahl der Wurzeln einen azyklischen Graphen mit den Wurzel-Knoten S zu erzeugen.

Bei zufälliger Bestimmung von k muss $P(k) = a_n^{(k)} / a_n$ gelten

Rekursive Erzeugung - Vor- und Nachteile

Vor- und Nachteile:

- $a_n^{(k)}$ müssen für alle Kombinationen von k und n berechnet und zwischengespeichert werden ($a_1 = 1; a_2 = 3; a_3 = 25;$ ... $a_7 = 1.138.779.265$)
- nicht immer zusammenhängend
- Variation von k und $s \Rightarrow$ breiter/hoher Graph
- rekursive topologische Aufspaltung $\Rightarrow$ einfacheres Ermitteln von Attribut-Abhängigkeiten
- Komplexität $O(n^2)$

Ein-Pass-Algorithmus - Voraussetzungen [dag99]

Algorithmus, der in einem Durchgang einen zusammenhängenden Graphen mit gegebener Anzahl an Knoten (n), Wurzeln (r) und limitiertem maximalen Eingangsgrad m erzeugt.

Notwendige und hinreichende Bedingungen für den Algorithmus:

- $n \geq 2$ (ein sinnvoller Graph muss mindestens zwei Knoten haben)
- $1 \leq r < n$ (mindestens eine Wurzel, aber nicht alle Knoten)
- $1 \leq m < n$ (notwendig für einen zusammenhängenden Graphen)
- Einschränkung von (n, r, m):
  - Wenn $r \geq m$ dann muss $m(n - r) \geq n - 1$ gelten,
  - sonst $m(n - m) + \frac{m(m-1)-r(r-1)}{2} \geq n - 1$

Eingabe: (n, r, m)
Ausgabe: zusammenhängender azyklischer gerichteter Graph G
Komplexität: $O(mn)$

Ein-Pass-Algorithmus

1. maximale Kantenanzahl berechnen (wenn $r \geq m$ dann $maxKa = m(n-r)$ sonst $maxKa = m(n-m) + 0.5(m(m-1) - r(r-1)))$
2. tatsächliche Kantenanzahl e zufällig aus $[n-1, maxKa]$ bestimmen
3. für alle Wurzel-Knoten $(v_0..v_{r-1})$ Hingrad $d^-(v_i) = 0$ setzen
4. für restliche Knoten: $d^-(v_i) \Leftarrow [1, min(i,m)]$; $\sum_{i=r}^{n-1} d^-(v_i) = e$ beachten
5. für alle Knoten Anzahl zu verbindender Eltern $p(v_i) = d^-(v_i)$ setzen
6. für i von r bis $n-1$
   - Kante von v_j zu v_i setzen $(j \Leftarrow [r-1, i-1])$
7. für i von 0 bis $r-2$
   - Kante von v_i zu x setzen $(x \Leftarrow \{v_j | p(v_j) \geq 1\})$
8. für i von r bis $n-1$
   - Solange $p(v_i) \geq 1$ wiederhole:
     - Kante von x zu v_i setzen $(x \Leftarrow \{v_j | 0 \leq j \leq i-1, (v_j, v_i) \notin E\})$

Bayessche Netze

Bayessches Netz:

- gerichteter azyklischer Graph
- jeder Knoten besitzt eine Variable mit Angabe über ihre statistische Verteilung.
- i.A. zusammenhängend

Erweitern eines Graphen zum Bayesschen Netz:

1. jedem Knoten v_i eine Variable V_i zuordnen
2. Anzahl möglicher Zustände für V_i bestimmen ($N_i = |V_i|$)
3. Für jeden Wurzelknoten: N_i Zufallswerte erzeugen, zu $\sum = 1$ normalisieren
4. Für alle anderen Knoten in topologischer Reihenfolge: für jede Belegung der Elternknoten N_i Zufallswerte erzeugen, zu $\sum = 1$ normalisieren
5. bei Bedarf beliebig viele Einzelbelegungen des Netzes entsprechend den Verteilungen erzeugen

Komplexität von Elternknoten-Anzahlen abhängig, $\approx O(mn)$

Zusammenfassung

Vorgestellte Algorithmen:

- zufällige Adjazenzmatrix
  - kaum Variationsmöglichkeiten im Algorithmus
  - Gleichverteilung über alle möglichen Graphen
- zufällige Kantenanordnung
  - erzeugte Graphen bedingt über die Parameter anpassbar
- Rekursive Erzeugung
  - grundlegende Struktur variierbar
  - komplexer Algorithmus, hohe Laufzeit
- Ein-Pass-Algorithmus
  - zusammenhängend, obwohl nur wenige Kanten
  - gut für Bayessche Netze geeignet
- Graphen $\to$ Bayes-Netze
  - kombinierbar mit o.g. Algorithmen
  - ermöglicht Erzeugung pseudo-statistischer Daten

Quellen

1. UAI Mailing list: http://cs.oregonstate.edu/~dambrosi/uai-archive/
2. G. Melancon, I. Dutour und M. Bousquet-Melou: „Random generation of dags for graph drawing.“; 2000, INS-R0005, ISSN 1386-3681; http://www.cwi.nl/ftp/CWIreports/INS/INS-R0005.ps.Z
3. Y. Xiang and T. Miller: „A Well-Behaved Algorithm for Simulating Dependence Structures of Bayesian Networks“; 1999; http://snowhite.cis.uoguelph.ca/faculty_info/yxiang/ ([dag99])
4. Fabio Gagliardi Cozman, Bayes-Netz-Generator in JAVA http://www.pmr.poli.usp.br/ltd/Software/BNGenerator/