

Phylogenetische Bäume

Patrick Schillinger
patrick.schillinger@uni-ulm.de

Universität Ulm – Fakultät für Informatik
Hauptseminar Bioinformatik SS 2006

2. August 2006

Inhaltsverzeichnis

1	Einführung	2
1.1	Motivation	2
1.2	verschiedene Arten	2
1.2.1	Merkmals basiert	2
1.2.2	Wahrscheinlichkeits basiert	2
1.2.3	Distanz basiert	3
2	Perfekte Phylogenie	3
2.1	Existenz	4
2.2	Prüfung der Existenz eines phylogenetischen Baumes in $O(nm)$	5
2.3	Erstellen eines phylogenetischen Baumes in $O(nm)$	6
3	Kompatibilität	7
3.1	$O(n)$ -Algorithmus	7
4	Fazit	8
5	Literatur	8

1 Einführung

1.1 Motivation

Phylogenetische Bäume dienen dazu, die Evolution verschiedener Spezies graphisch darzustellen. In einem phylogenetischen Baum soll man auf einen Blick sehen welche Arten wie stark miteinander verwandt sind, durch welche Mutationen sie sich im einzelnen unterscheiden und welche einen gemeinsamen Vorfahren haben.

1.2 verschiedene Arten

Es gibt im wesentlichen drei Hauptarten phylogenetischer Bäume:

- Merkmals basiert
- Wahrscheinlichkeits basiert
- Distanz basiert

1.2.1 Merkmals basiert

Die Blätter des Baumes sind die Spezies, die Knoten sind gemeinsame Vorfahren und die Kanten sind die Merkmale. Eine Spezies besitzt also alle Merkmale, die auf dem Weg von der Wurzel zum entsprechenden Blatt liegen. Es gibt drei Arten von Merkmalen:

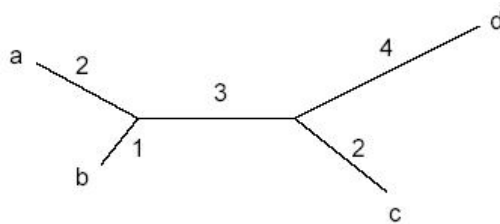
- binäre Merkmale (z.B Säugetier ja=1/nein=0)
- numerische Merkmale (z.B Anzahl Beine)
- zeichenreihige Merkmale (z.B bestimmte DNA-Teilsequenzen)

1.2.2 Wahrscheinlichkeits basiert

Hierbei werden für verschiedene Mutationen , verschiedene Wahrscheinlichkeiten bestimmt. Anschliessend werden alle möglichen Bäume berechnet (inklusive ihrer Wahrscheinlichkeit) und der Baum mit der höchsten Wahrscheinlichkeit wird als Lösung genommen. Diese Methode ist natürlich sehr rechenaufwendig.

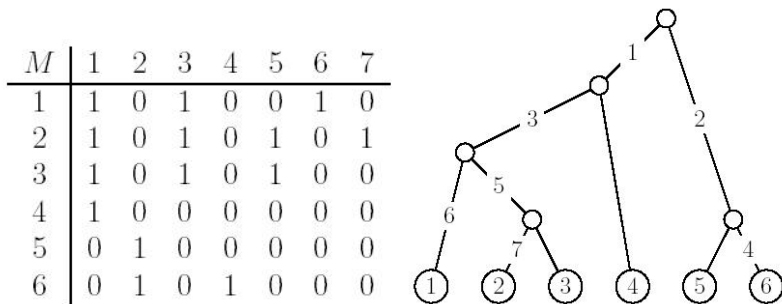
1.2.3 Distanz basiert

Hierbei werden die Distanzen der verschiedenen Spezies zueinander bestimmt. Je größer die Distanz, desto mehr unterscheiden sie sich. An den Kanten steht die Distanz von einem Blatt (lebende Spezies) zu einem Knoten (Vorfahr), oder von einem Knoten zu einem anderen Knoten. Addiert man die Distanzen auf den Kanten, die auf dem Weg von einem Blatt zum anderen liegen, erhält man die Distanz zwischen zwei lebenden Spezies. (z.B. hat Spezies a zu Spezies c die Distanz 7)



2 Perfekte Phylogenie

Bei der perfekten Phylogenie hat man eine binäre Eigenschaftsmatrix gegeben. In ihr entsprechen die Spalten den Eigenschaften und die Zeilen den Spezies. Besitzt also Spezies i die Eigenschaft j, dann steht in der Zelle (i,j) eine 1; anderenfalls eine 0. Daraus wird der phylogenetische Baum berechnet. In ihm sind dann die Blätter die Spezies und die Kanten die Eigenschaften. Wichtig ist, dass eine Eigenschaft genau eine Kante ist. D.h., dass eine neue Eigenschaft nur einmal hinzu kommen kann und auch nicht mehr verschwindet. (siehe Fazit)



2.1 Existenz

Natürlich kann man nicht aus jeder binären Matrix einen phylogenetischen Baum erstellen. Das Hauptproblem hierbei ist, dass eine Mutation nicht in 2 verschiedenen Zweigen gleichzeitig auftreten kann.

Theorem:

Sei O_k die Menge der Zellen mit Wert 1 in Spalte k. Eine Matrix besitzt einen phylogenetischen Baum genau dann wenn für jedes Spaltenpaar i,j gilt:

$$O_i \subseteq O_j \text{ oder } O_j \subseteq O_i \text{ oder } O_i \cap O_j = \emptyset$$

Beweis:

” \Rightarrow ”

Seien k_i und k_j zwei Kanten des Baums, an denen Merkmal i bzw. j zum ersten mal auftritt (also den Wert 1 annimmt).

- Fall1: $k_i = k_j \Rightarrow O_i = O_j$
- Fall2: eine Kante liegt auf dem Pfad von der Wurzel zur anderen $\Rightarrow O_i \subset O_j$ bzw. $O_j \subset O_i$
- Fall3: Die Pfade zu den Kanten verzweigen, bevor eine der beiden Kanten erreicht wurde. $\Rightarrow O_i \cap O_j = \emptyset$

” \Leftarrow ”

Seien s_1 und s_2 zwei Spezies. Sei k die größte Eigenschaft, die beide Spezies besitzen (größte bedeutet hier: am weitesten rechts in einer sortierten Matrix). Sei i eine beliebige Eigenschaft von s_1 , die jedoch kleiner als k ist. s_1 besitzt Merkmal i und k

$$\Rightarrow O_i \cap O_k \neq \emptyset, \text{ k größer (weiter rechts)}$$

$$\Rightarrow O_k \subseteq O_i$$

$$\Rightarrow s_2 \text{ besitzt ebenfalls Merkmal i.}$$

\Rightarrow Werden nun für die einzelnen Spezies Strings zusammengestellt, die aus den Bezeichnungen der Merkmale der jeweiligen Spezies und einem abschließenden Terminalsymbol bestehen (z.B.: Spezies a hat Merkmale 1,5,6 und 9. der String hierzu wäre 1569\$), erstellt, dann sind die Strings zweier Spezies bis auf eine Eigenschaft identisch und danach haben sie keine Gemeinsamkeiten mehr.

⇒ aus diesen Strings kann ein Keywordtree erstellt werden, der eine perfekte Phylogenie darstellt.

q.e.d

Beispiel:

M	1	2	3	4	5	6	7
1	1	0	1	0	0	1	0
2	1	0	1	0	1	0	1
3	1	0	1	0	1	0	0
4	1	0	0	0	0	0	0
5	0	1	0	0	0	0	0
6	0	1	0	1	0	0	0

M	1	2	3	4	5	6	7
1	1	0	1	0	0	1	0
2	1	0	1	0	1	0	1
3	1	0	1	1	1	0	0
4	1	0	0	0	0	0	0
5	0	1	0	0	0	0	0
6	0	1	0	1	0	0	0

⇒ phylogenetischer Baum existiert! ⇒ phylogenetischer Baum existiert nicht,
da Spalte 4 wegen der zusätzlichen 1
die Bedingungen nicht mehr erfüllt.

2.2 Prüfung der Existenz eines phylogenetischen Baumes in $O(nm)$

1. Spalten der Größe nach sortieren (da die Matrix binär ist, kann man die Spalten als Binärzahlen interpretieren. Die Sortierung funktioniert mittels Radixsort in $O(nm)$)
2. Lösche alle Spalten, die identisch mit der Spalte zu ihrer Rechten sind.
3. Sei O die Menge der Zellen mit Wert 1.

Für jede Zelle $(i,j) \in O$ setze $L(i,j)$ dem größten Index $k < j$, so dass $M'(i,k) \in O$. Setze $L(i,j)=0$, falls ein solcher Index nicht existiert. Für jede Spalte j sei $L(j)=$ dem größten $L(i,j)$.

4. Falls alle $L(i,j)=L(j)$ so dass $(i,j) \in O \Rightarrow$ existent

Es wird also nach der letzten gemeinsamen Eigenschaft von zwei Spezies gesucht und geprüft, ob alle Eigenschaften vor dieser übereinstimmen.

2.3 Erstellen eines phylogenetischen Baumes in $O(nm)$

1. Spalten der Größe nach sortieren (mittels Radixsort)
2. String zu jeder Zeile konstruieren (Der String besteht aus den Bezeichnungen der Eigenschaften der Spezies in dieser Zeile gefolgt von einem Terminalsymbol. Z.B.: Spezies a hat Merkmale 1,5,6 und 9. der String hierzu wäre 1569\$)
3. Keyword-Tree aus den Strings erstellen
4. Teste, ob T eine perfekte Phylogenie von M ist (trivial, da falls die Existenz positiv überprüft wurde und ein Keywordtree erstellt wurde, muss es sich bereits um eine perfekte Phylogenie handeln)

Beispiel:

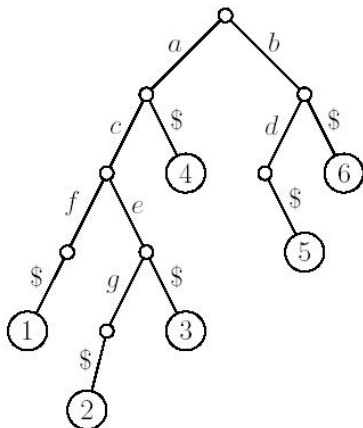
Eigenschaftenmatrix:							
M	a	b	c	d	e	f	g
1	1	0	1	0	0	1	0
2	1	0	1	0	1	0	1
3	1	0	1	0	1	0	0
4	1	0	0	0	0	0	0
5	0	1	0	0	0	0	0
6	0	1	0	1	0	0	0

Sortierte Eigenschaftenmatrix:							
M'	a	c	f	e	g	b	d
1	1	1	1	0	0	0	0
2	1	1	0	1	1	0	0
3	1	1	0	1	0	0	0
4	1	0	0	0	0	0	0
5	0	0	0	0	0	1	0
6	0	0	0	0	0	1	1

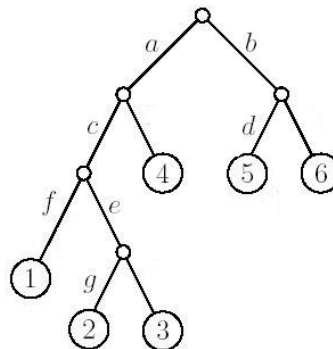
Strings zu den Zeilen:

acf\$ aceg\$ ace\$ a\$ b\$ bd\$

Baum erstellen:



Terminalsymbol löschen:



3 Kompatibilität

Oft gibt es verschiedene Datensätze, in denen jedoch teilweise die gleichen Spezies auftreten. Z.B. kann ein Datensatz genauer sein, also mehr Merkmale betrachten, oder mehr Spezies enthalten. Das Problem hierbei ist die Kompatibilität der Bäume. D.h.: beschreiben zwei Bäume den gleichen evolutionären Vorgang? Also kann man sie zu einem gemeinsamen Baum kombinieren?

Def.: Ein Baum T_i ist eine Verfeinerung eines Baumes T_j , wenn man T_j durch Kantenreduktion aus T_i erhält.

Theorem: Zwei Bäume T_1 und T_2 sind kompatibel, wenn es einen Baum T_3 gibt, der eine Verfeinerung von T_1 und T_2 ist.

3.1 $O(n)$ -Algorithmus

1. Während eines Tiefendurchlaufs durch Baum T_1 :

- Von jedem Objekt i zu jedem Blatt $V_1(i)$ in dem i enthalten ist einen Pointer setzen.
- Sei $N_1(i)$ die Anzahl der Elemente von $V_1(i)$.
- Speicher für jeden Knoten x Die Anzahl der Objekte unter x .
- Das Selbe für T_2 und anschliessend alle Objekte aktiv setzen

2.

For $j=1$ to 2

if $j=1$, then $k=2$, else $k=1$

while (es gibt ein aktives Objekt i , so dass $N_j(i)$ größer $N_k(i)$)

do

Gehe von $V_k(i)$ zur Wurzel von T_k bis ein Knoten x erreicht wurde, der mindestens $N_j(i)$ Objekte in den Blättern seines Teilbaums $T(x)$ hat. Prüfe für alle Objekte in den Blättern von $T(x)$, ob es die selben sind, die in $V_j(i)$ sind.

if not

break T_1 und T_2 nicht kompatibel

else

Ersetze den Knoten $V_j(i)$ in T_j durch $T(x)$ Alle Objekte in $T(x)$ inaktive setzen.

Der Algorithmus erstellt also zuerst eine Tabelle, in der für jedes Element (Spezies) gespeichert wird, in welchem Blatt es enthalten ist und wieviele Elemente in diesem Blatt enthalten sind. Danach sucht man das erste Element, das im ersten Baum mehr Nachbarn hat, als im zweiten. Das bedeutet, dass der zweite Baum an dieser Stelle feiner ist, falls sie kompatibel sind. Dies ist er aber nur, wenn die Elemente in diesem Blatt (im ersten Baum) gleich den Elementen in dieser Verfeinerung (im zweiten Baum) sind. Falls ja, wird das Blatt durch die Verfeinerung ersetzt. Das wird so lange wiederholt, bis keine weiteren Elemente mehr gefunden werden, die im ersten Baum mehr Nachbarn haben, als im zweiten. Anschliessend wird das ganze aus der Sicht vom zweiten Baum wiederholt.

4 Fazit

Die hier vorgestellten Methoden werden wahrscheinlich nicht genügen, um die wahren evolutionären Hintergründe aller Spezies zu bestimmen. Das Hauptproblem ist, dass man das Auftreten der gleichen Mutation in zwei verschiedenen Zweigen und das wieder Verschwinden eines Merkmals ausschliesst. Dabei gibt es in der Natur viele Beispiele, dass dem nicht so ist. Z.B. waren Wale Landsäugetiere, die wieder ins Wasser zurück sind, oder Fledermäuse und Adler haben Flügel, obwohl sie keinen gemeinsamen fliegenden Vorfahren haben. (Als Säugetiere wären Fledermäuse stärker mit dem Menschen als mit den Vögeln verwandt)

5 Literatur

[1] Dan Gusfield- Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, 1997

[2] Dan Gusfield- Efficient Algorithms for Inferring Evolutionary Trees, University of California - Davis, 1991

[3] Volker Heun- Skriptum zur Vorlesung Algorithmische Bioinformatik: Bäume und Graphen, LMU München, 29. Juni 2006 Version 0.24