**دانشکده مهندسی کامپیوتر**
**دانشگاه صنعتی شریف**

# ژنومیک محاسباتی

**مطهری و فروغمند**
پاییز ۱۴۰۰

# درخت تبارزایی (۳)

جلسه ششم

**نگارنده: حسین ولی‌شیرین**

## ۱ مروری بر مباحث گذشته

در جلسات گذشته مطرح شد درخت‌های تبارزایی را می‌توان توسط دو مدل بازسازی کرد:

- **ورودی:** ویژگی‌ها، که مفصلا به شرح آن پرداخته شد.
- **ورودی:** ماتریس فاصله‌ها، که در این جلسه به آن پرداخته خواهد شد.

## ۲ بازسازی فاصله‌مبنای درخت تبارزایی

- **ورودی** : ماتریس $M$ فواصل دوبه‌دو می‌باشد.

$$\text{تعریف فاصله} = \begin{cases} M_{ij} = M_{ji} \quad and \quad M_{ii} = 0 \quad \forall i,j & \text{متقارن} \\ M_{ij} + M_{jk} \geq M_{ik}, \quad \forall i,j,k & \text{نامساوی مثلثی} \end{cases}$$

- **خروجی:** درخت فیلوژنی.

**تذکر ۱.** در اینجا چون جهتی وجود ندارد، ریشه خیلی معنایی ندارد و درخت بی‌ریشه است.

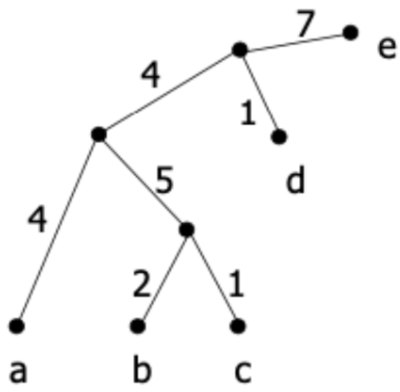
**تذکر ۲.** می‌دانیم که در درخت بی‌ریشه، رئوس میانی درجه ۳ بوده و برگ‌ها درجه یک اند.

در مواجه با این مسئله، دو حالت خاص وجود دارد که در ادامه آن می‌پردازیم.

## ۱.۲ ماتریس فاصله جمعی

**تعریف ۳.** ماتریس $M$، ماتریس فاصله جمعی است اگر و تنها اگر یک درخت با وزن یال‌های مثبت برای آن وجود داشته باشد و فاصله دو برگ برابر با جمع فاصله یال‌های بین آن‌ها باشد.

| $M$ | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | 0 | 11 | 10 | 9 | 15 |
| $b$ | 11 | 0 | 3 | 12 | 18 |
| $c$ | 10 | 3 | 0 | 11 | 17 |
| $d$ | 9 | 12 | 11 | 0 | 8 |
| $e$ | 15 | 18 | 17 | 8 | 0 |



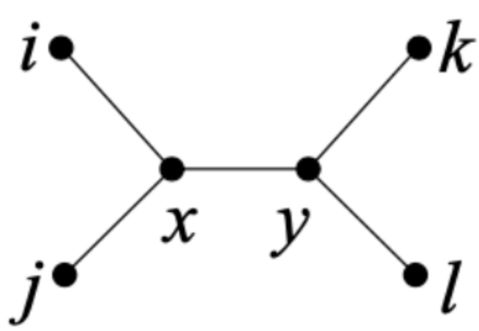

شکل ۱: یک نمونه درخت فاصله جمعی همراه با ماتریس آن

از جمله مسائلی که در این زمینه ممکن است با آن مواجه شویم، تشخیص فاصله جمعی بودن یک ماتریس و یافتن درخت فاصله جمعی از روی ماتریس است.

**قضیه ۴** (شرط چهار نقطه). ماتریس $M$، برای درخت یکتای $T$ فاصله جمعی است، اگر و تنها اگر به ازای $\forall i, j, k, l$ بتوان نام‌گذاری‌شان را تغییر داد که داشته باشیم:

$$M_{ik} + M_{jl} = M_{il} + M_{jk} \geq M_{ij} + M_{kl}$$

**اثبات.** • اگر فواصل جمعی باشد آن‌گاه، شرط ۴ نقطه برقرار است.
برای اثبات آن باید ابتدا تمام برگ‌ها غیر از چهار برگ $i, j, k, l$ حذف شوند. در این مسیر ممکن است راس‌های غیر برگ به برگ تبدیل شوند که در این صورت آن‌ها نیز حذف می‌شوند. این عمل تا جایی ادامه پیدا خواهد کرد که تمامی رئوس باقی‌مانده یا برگ باشند $(i, j, k, l)$ و یا رئوس میانی با درجه دو یا سه. در ادامه رئوس درجه دو را نیز حذف می‌کنیم و تنها رئوس درجه سه و برگ‌های مذکور باقی می‌مانند. در شکل ۲ نمونه‌ای از این درخت را مشاهده می‌کنید.




شکل ۲: یک نمونه درخت

**مشاهده ۵.** اگر گراف راسی با درجه سه نداشت، چهار راس $i, j, k, l$ همبند نمی‌شدند؛ مگر آن‌که راسی از درجه چهار از آن‌ها وجود داشت که این فرض باطل است و بیشترین درجه رئوس سه می‌باشد.

با توجه به شکل۲، روابط زیر برقرارند:

$$\begin{cases} M_{il} = & M_{ix} + M_{xy} + M_{yl} \\ M_{jk} = & M_{jx} + M_{xy} + M_{yk} \\ M_{ij} = & M_{jx} + M_{ix} \\ M_{kl} = & M_{yl} + M_{yk} \end{cases}$$

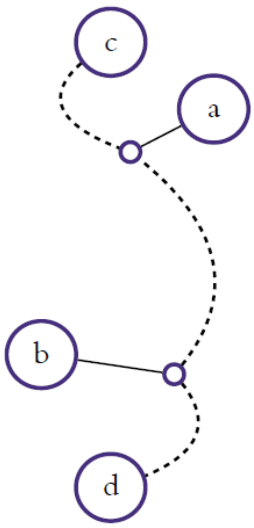
در روابط بالا $M_{xy}$ دو بار تکرار شده در حالی که در فاصله $M_{ij}$ و $M_{kl}$ اصلا یال $M_{xy}$ دخالتی ندارد. در نتیجه شرط چهار نقطه برقرار است.

- اگر شرط چهار نقطه برقرار باشد آن‌گاه، درختی موجود است که ماتریس فاصله‌ها از روی آن ساخته شده است.

یک گزاره قوی‌تر را اثبات میکنیم.

**گزاره ۶.** اگر شرط چهار نقطه برقرار باشد آن‌گاه، درختی یکتا موجود است که ماتریس فاصله‌ها از روی آن ساخته شده است.

فرض می‌کنیم برای تمامی ماتریس‌ها در صورت صدق شرط جهار نقطه، با کمتر از n عضو یک درخت یکتا موجود است. قصد داریم با استقرا حکم را برای n راس اثبات کنیم.
به شکل ۳ توجه کنید.




شکل ۳: یک نمونه درخت

دو برگ $a$ و $b$ را از درخت بالا (دو سطر و ستون از ماتریس) در نظر می‌گیریم. یک بار سطر و ستون $a$ را حذف کرده و ماتریس جدیدی ایجاد می‌شود که شرط چهار نقطه برای آن برقرار است. به طور مشابه این عمل برای برگ $b$ نیز تکرار می‌شود.

اگر $S$ مجموعه تمامی راس‌ها باشد، $T$درخت یکتایی است که از حذف راس $a$ از $S$به دست آمده است. همچنین $T'$درخت یکتایی‌ست که از حذف راس $b$ حاصل شده است.

باید اثبات شود دو درخت $T$ و $T'$ به جز دو راس $a$ و $b$ بر هم منطبق‌اند. برای اثبات آن، اگر از $T'$ برگ $a$ نیز حذف شود، راس درجه سه متصل به $a$ به درجه دو تبدیل می‌شود. پس از ساده‌سازی، درخت $T'$ دو برگ $a$ و $b$ را نخواهد داشت و ماتریس فواصلی که ایجاد می‌شود سطر و ستون $a$ و $b$ را نخواهد داشت و درخت حاصل طبق فرض استقرا یکتاست.

به طور مشابه از درخت $T$، برگ $b$ حذف شده و روند قبلی تکرار می‌شود و در نهایت باید به درختی مشابه با $T'$ تبدیل شود. حال با توجه به یکسان بودن درخت‌های نهایی، دو برگ $a$ و$b$ را به محلی که از درخت جدا شده بودند مجددا متصل می‌کنیم.

حال باید اثبات شود فواصل موجود در درخت همان فواصل ماتریس $M$ است. اگر فاصله هر دو راسی غیر از $a$ و $b$ در درخت در نظر گرفته شود، یا هر دو راس در درخت $T$ قرار دارند یا در درخت $T'$ . چون فواصل دو درخت در ماتریس $M$وجود داشت در نتیجه فواصل هر دو نقطه نیز

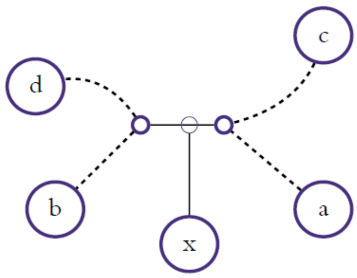
در ماتریس قرار دارد. تنها فاصله‌ای که در این دو درخت نیست فاصله $a$ و $b$ است که باید ثابت کنیم این مقدار برابر $M_{ab}$ است. می‌توان از رابطه زیر با شرط برقراری شرط چهار نقطه، استفاده کرد؛ یعنی نوشت:

$$M_{ad} + M_{bc} - M_{cd} = M_{ab}$$

از طرفی، حالت $M_{ad} + M_{bc} \geq M_{cd} + M_{ab}$ نیز به دلیل آن‌که $c$ و $d$ در فاصله دورتری از $a$ و$b$ قرار دارد، اتفاق نمی‌افتد. پس فواصل در درخت و ماتریس با هم برابر هستند. حکم استقرا ثابت شد. □

### ۱.۱.۲ الگوریتم ماتریس جمعی

اگر سه‌تایی از گونه‌ها داشته باشیم که تنها یک درخت بتوان برای آن ساخت، طول یال‌ها را می‌توان به صورت یکتا بدست آورد. به همین ترتیب راس چهارم هم مطابق شکل ۴ اضافه می‌شود.




شکل ۴: یک نمونه درخت

حال اگر راس جدیدی به نام $x$ را به درخت اضافه کنیم، به دلیل یکتا بودن درخت، گونه جدید را تنها به یک نقطه می‌توان اضافه کرد. همچنین می‌دانیم با حذف $x$ درخت مجدد یکتا خواهد شد. باید یالی که گونه $x$ به آن متصل می‌شود را پیدا کنیم. یال مدنظر این خاصیت را دارد که اگر زیر درخت $bdc$ را در نظر بگیریم، یال مذکور با خاصیت مورد نظر به صورت یکتا یافت خواهد شد.

در نتیجه، برای هر یال، زیر درخت سه‌تایی آن را ساخته و سپس یالی که گونه $x$ به آن متصل می‌گردد، یافت می‌شود.

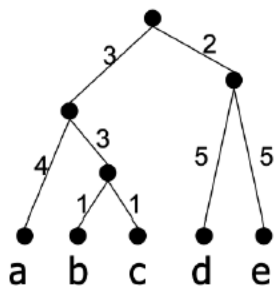
زمان اجرا: $O(n^2)$ (درخت تعداد $n$ یال دارد و به همین تعداد الگوریتم تکرار خواهد شد.)

## ۲.۲ ماتریس ابرمتریک

**تعریف ۷.**
- $M$ جمعی باشد.
- درخت ریشه‌دار $T$ با وزن‌های مثبت
- فاصله تمامی برگ‌ها با ریشه برابر است.

در شکل ۵ یک نمونه ماتریس و درخت ابرمتریک را ملاحظه می‌نمایید.

| $M$ | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | 0 | 8 | 8 | 14 | 14 |
| $b$ | 8 | 0 | 2 | 14 | 14 |
| $c$ | 8 | 2 | 0 | 14 | 14 |
| $d$ | 14 | 14 | 14 | 0 | 10 |
| $e$ | 14 | 14 | 14 | 10 | 0 |




شکل ۵: یک نمونه درخت و ماتریس ابرمتریک

**قضیه ۸.** اگر دو گونه $i, j$ را داشته باشیم و $v$ کوچک‌ترین جد مشترک آن‌ها باشد، آن‌گاه:

$$d(i, v) = d(j, v) = \frac{M_{ij}}{2}$$

فاصله بین ریشه و راس $v$ برای هر دو راس مشترک است در نتیجه دو قسمت بعدی نیز باید با هم برابر باشد.

$$d(i, v) = d(j, v)$$

**قضیه ۹** (شرط سه نقطه). ماتریس $M$ ابرمتریک است اگر و تنها اگر به ازای هر سه گونه $i, j, k$ بتوان نام‌گذاریشان را تغییر داد که داشته باشیم:

$$M_{ik} = M_{jk} \geq M_{ij}$$

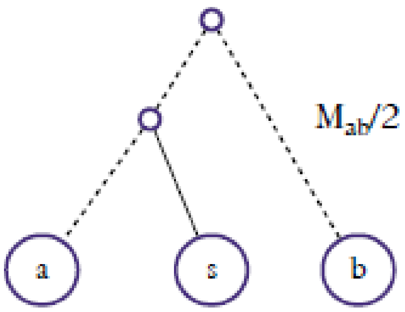
**اثبات.** • اگر ماتریس ابر متریک باشد، شرط سه نقطه برقرار است.

اگر هر سه گونه‌ای را به صورت رندوم در نظر بگیریم و برگ‌ها را ساده کنیم، جد مشترک هر سه بالاتر قرار می‌گیرد. همچنین دو گونه به یک راس درجه سه متصل‌اند.

• اگر شرط سه نقطه برقرار باشد، ماتریس ابرمتریک است.

اگر شرط سه نقطه برقرار باشد، آن‌گاه شرط چهار نقطه نیز برقرار است. (تمرین) پس درخت بی‌جهت یکتا از ورودی ماتریس $M$ را می‌توان ایجاد کرد. در این راستا دو مورد مهم باید تعیین شود:

۱. محل قرارگیری ریشه درخت: ریشه در وسط بلندترین مسیر درخت قرار می‌گیرد.




شکل ۶: درخت مذکور در روند اثبات قضیه۹

۲.اثبات شود فاصله ریشه تا تمامی برگ‌ها یکسان است: به شکل۶ توجه کنید. فاصله ریشه تا $a$ و فاصله ریشه تا گونه $b$ برابر است. حال می‌توان یک راس جدید مانند $s$ را در نظر گرفت که به گونه $a$ نزدیک‌تر از گونه $b$ باشد. آنگاه رابطه زیر برقرار است:

$$M_{ab} = M_{sb} \geq M_{sa}$$

در نتیجه حکم اثبات می‌شود.

□

## ۳.۲ الگوریتم‌ها

### ۱.۳.۲ الگوریتم UPGMA

**UPGMA algorithm**

Set $\mathcal{C} = \{\{c_1\},\{c_2\},\ldots,\{c_n\}\}$ where $height(\{c_i\}) = 0$ for $i \in \{1,\ldots,n\}$;
For all $\{c_i\},\{c_j\} \in \mathcal{C}$, set $dist(\{c_i\},\{c_j\}) = M_{ij}$;
**for** $i = 2$ to $n$ **do**
  Determine clusters $C_i, C_j \in \mathcal{C}$ such that $dist(C_i, C_j)$ is minimized;
  Let $C_k$ be a cluster formed by connecting $C_i$ and $C_j$ to the same root;
  Let $height(C_k) = dist(C_i, C_j)/2$;
  Let $d(C_k, C_i) = height(C_k) - height(C_i)$;
  Let $d(C_k, C_j) = height(C_k) - height(C_j)$;
  $\mathcal{C} = \mathcal{C} - \{C_i, C_j\} \cup \{C_k\}$;
  For all $C_x \in C - \{C_k\}$, define $dist(C_x, C_k) = dist(C_k, C_x) = \frac{|C_i| dist(C_i,C_x) + |C_j| dist(C_j,C_x)}{(|C_i|+|C_j|)}$
**end for**

$$dist(C_1, C_2) = \frac{\sum_{i\in C_1, j\in C_2} M_{ij}}{|C_1| \times |C_2|}$$

شکل ۷: الگوریتم UPGMA

این الگوریتم اگر ماتریس ابرمتریک را به عنوان ورودی بگیرد، می‌توان درخت ماتریس را بسازد.

در مرحله اول، همه رئوس را می‌گیرد و هر راس را به یک کلاستر تبدیل می‌کند. ماتریس دو به دو هر کلاستر ساخته می‌شود. سپس، ارتفاعی برای کلاسترها در نظر می‌گیرد که این مقدار در ابتدا برابر صفر می‌باشد.

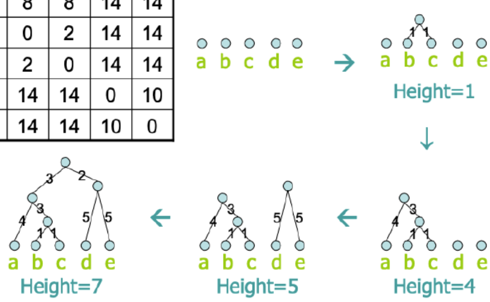
هر دو راسی که فاصله بین کلاسترهایشان از بقیه کمتر باشد، مانند $C_i, C_j$ را با هم ادغام می‌کند. بدین صورت که یک کلاستر رو مثلا $C_k$ را بالای آن‌ها قرار داده و دو کلاستر را به آن وصل می‌کند. ارتفاع $C_k$ برابر با نصف فاصله بین دو کلاستر $C_i, C_j$ قرار می‌دهد.

همچنین، فاصله بین $C_k$ و هرکدام از دو کلاستر دیگر با توجه به اختلاف ارتفاع $C_k$ و آن دو بدست می‌آید.

**نکته.** در ابتدای کار الگوریتم، فاصله واقعی بین هر دو راس مشخص و موجود است. اما فاصله واقعی رئوس ایجاد شده میانی مشخص نیست. فاصله بین رئوس میانی و باقی رئوس به صورت میانگین فاصله دو به دو کلاسترها با هم تقسیم بر ضرب تعدادشان، بدست می‌آید.

**مثال ۱۰.** ساخت درخت برای گونه‌های $a, b, c, d, e$

| M | a | b | c | d | e |
|---|---|---|---|---|---|
| a | 0 | 8 | 8 | 14 | 14 |
| b | 8 | 0 | 2 | 14 | 14 |
| c | 8 | 2 | 0 | 14 | 14 |
| d | 14 | 14 | 14 | 0 | 10 |
| e | 14 | 14 | 14 | 10 | 0 |




شکل ۸: مثالی از کاربرد الگوریتم UPGNA

دو راس نزدیک به هم با توجه به ماتریس فاصله‌ها دو راس $c, b$ هستند. برای ادغام آن‌ها یک راس بالاتر با ارتفاع نصف فاصله آن دو یعنی یک قرار می‌گیرد. برای ادامه کار فاصله کلاستر جدید و باقی نقاط محاسبه می‌شود که در این بین فاصله آن با راس $a$ مطابق رابطه زیر، دارای کمترین مقدار می‌باشد.

$$\frac{ca + cb}{۲} = \frac{۸ + ۸}{۲} = ۸$$

در نتیجه نقطه میانی جدیدی بالای آن با ارتفاع ۴ قرار می‌گیرد و این روند ادامه پیدا خواهد کرد.

**قضیه ۱۱.** الگوریتم $UPGMA$ روی ماتریس‌های ابرمتریک درست کار می‌کند.

**نکته.** اگر ماتریس ما با ماتریس ابرمتریک ان‌قدر دارای اختلاف کمی باشد که در تصمیمات الگوریتم بی‌اثر باشد، یعنی فواصل با نویز همرار باشد اما کمترین فاصله تغییری نکند، در آن صورت خروجی الگوریتم UPGMA با درخت بدون نویز برابر است.

### ۲.۳.۲ الگوریتم Neighbor-Joining

**Neighbor-Joining algorithm**

Let $Z = \{\{1\}, \{2\}, \cdots, \{n\}\}$ be the set of initial clusters;
For all $\{i\}, \{j\} \in Z$, set $dist(\{i\}, \{j\}) = M_{ij}$;
**for** $i = 2$ to $n$ **do**
  For every cluster $A \in Z$, set $u_A = \frac{1}{n-2} \sum_{D \in Z} dist(D, A)$;
  Find two clusters $A, B \in Z$ which minimizes $dist(A, B) - u_A - u_B$;
  Let $C$ be a new cluster formed by connecting $A$ and $B$ to the same root $r$. Let $r_A$ and $r_B$ be the roots of $A$ and $B$. The edge weights of $(r, r_A)$ and $(r, r_B)$ are $\frac{1}{2}dist(A, B) + \frac{1}{2}(u_A - u_B)$ and $\frac{1}{2}dist(A, B) + \frac{1}{2}(u_B - u_A)$, respectively;
  Set $Z = Z \cup \{C\} - \{A, B\}$;
  For any $D \in Z - \{C\}$, define $dist(D, C) = dist(C, D) = \frac{1}{2}(dist(A, D) + dist(B, D) - dist(A, B))$;
**end for**

شکل ۹: الگوریتم Neighbor-Joining

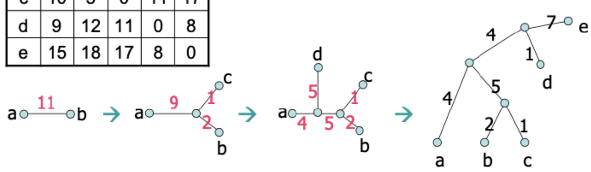
در ابتدا مشابه با الگوریتم قبلی عمل می‌کند اما ارتفاع را لحاظ نمی‌کند. برای هر کلاستر، میانگین فاصله تا باقی کلاسترها را محاسبه کرده و دو کلاستری را با هم مرج می‌کند که دو ویژگی به طور همزمان داشته باشد:
۱.فاصله از یک‌دیگر کمترین باشد.
۲. فاصله دو کلاستر با باقی کلاسترها حداکثر باشد.
به طور مثال اگر دو کلاستر $A, B$ دارای این شروط بوده و مرج شوند در نقطه $D$ ، فاصله این راس تا باقی کلاسترها به صورت زیر تعریف می‌شود:

$$\frac{۱}{۲}(dis(A, D) + dis(B, D) - dis(A, B))$$

**قضیه ۱۲.** الگوریتم *Neighbor-Joining* روی متریک‌های جمعی درست کار می‌کند.

مثال:

| M | a | b | c | d | e |
|---|---|---|---|---|---|
| a | 0 | 11 | 10 | 9 | 15 |
| b | 11 | 0 | 3 | 12 | 18 |
| c | 10 | 3 | 0 | 11 | 17 |
| d | 9 | 12 | 11 | 0 | 8 |
| e | 15 | 18 | 17 | 8 | 0 |




شکل ۱۰: مثالی از الگوریتم Neighbor-Joining

# مراجع

[WK09] Sung Wing-Kin. *Algorithms in bioinformatics: A practical introduction*. Chapman Hall/CRC Computational Biology Series, 2009.