92,7,22 Data mining جلسه 1

OLAP چیزی نیست جز جدول، جدولی که شاید مثلاً ۵۰ سال پیش توسط آماری ها ابداع شد و الان توی داده کاوی از آن استفاده میشود، و به عنوان یک روش خیلی توصیفی مهم از آن یاد می‌شود، خیلی چیز ساده ای است و با چند مثال توضیح میدهیم. کاری که OLAP انجام می دهد در واقع جلا دادن داده ها یا Database آن توی جدول است، توی جدول چند بعدی، مشکلی که ما داریم، اسم جدول چند بعدی بررسی طلف، کار راحتی نیست، تنها کاری که این روش انجام می دهد این است که می آید، قضیه ها پیاده را می سنگین تر می کند و آنها را روی یک صفحه نشان می دهد. قبلاً ما حرفی اینکه این کار را انجام دهیم یک نرم افزار خاص داشتیم، الان نه، تمام نرم افزارها این کار را میکنند. توسط بعضی شرکت ها ابداع شده

OLAP

- On-Line Analytical Processing (OLAP) was proposed by E. F. Codd, the father of the relational database.
- Relational databases put data into tables, while OLAP uses a multidimensional array representation.
  - Such representations of data previously existed in statistics and other fields
- There are a number of data analysis and data exploration operations that are easier with such a data representation.

مثلاً spss یا elementin این نرم افزارهم یه جاهایی توی اسم این جور جداول دارد و خیلی حرفهای توی این جداول کار میشه. مثلاً در پزشکی و علوم سرطان بسیار ۲ هم دیگر ربط دارند یا ربط ندارند، این ربط سرطان و سیگار چه جوری است؟! ما یکسری داده داریم که نسبتی بودند که سیگار می کشیدند سرطان گرفتند یا نه و نه که سیگار می کشیدند و سرطان گرفتند و کسانی که سیگار می کشیدند و سرطان گرفتند و آدم هایی که سیگار نمی کشیدند سرطان گرفتند یا نه!! در واقع ما یک جدول داریم، با این جدول چه کار می کنیم؟ سرطان، نه یا بله، سیگار (معتاد) و معتاد نیست، ۴ حالت داریم، حالت اول ما از این جدول چه استفاده می کنیم؟ OLAP می گوید: این جدول برای داده های مجموعه سفارشی درست هستند. برای داده های پیوسته هم یک درستش کرده. جدول توافقی، یعنی جداول توافقی که قبلاً در مورد آن صحبت کردیم، بعضی مثلاً Contigensi table

Creating a Multidimensional Array

- Two key steps in converting tabular data into a multidimensional array.
  - First, identify which attributes are to be the dimensions and which attribute is to be the target attribute whose values appear as entries in the multidimensional array.
    - The attributes used as dimensions must have discrete values
    - The target value is typically a count or continuous value, e.g., the cost of an item
    - Can have no target variable at all except the count of objects that have the same set of attribute values
  - Second, find the value of each entry in the multidimensional array by summing the values (of the target attribute) or count of all objects that have the attribute values corresponding to that entry.

یعنی جداول توافقی، بعضی هاهم در همین حد جدول ضربدری هستند که در OLAP جای دهد. جدول OLAP را از دو بعدی میتوانیم شروع کنیم و ببریم جلو ولی OLAP واقعاً برای دو بعدی هم نیست جای ۲ بعدی است، یعنی ما داده های که داریم، آن داده ها، همگروه اش ممکن است چند تا شود، چند تا متغیر دیگر دخیل باشد و اینها را مجبوریم تقسیم بندی کنیم.

غالباً در حالت ۳ بعدی ما میتوانیم این را به صورت مکعبی رسم کنیم و روش های جدیدی هم در سال های اخیر توسط نرم افزارها ابداع شده برای نمایش دادن این جداول، مثلاً سنتاخ جدید هم نرم افزارهای جدیدی ایجاد کرده، ما خیلی خودمان هم با این نمودارهای جدید آشنا شیم. پس برای اینکه ما یک Multi dimensional Array را درست کنیم، یک آرایم چند بعدی درست کنیم ما باید جدول چند بعدی درست کنیم، باید به چند نکته ما توجه کنیم، پس اگر بردن یک نمودار چند بعدی مثل اینکه ما داده ها را در یک آرایه بریزیم، حالا ما می رسیم یک آرایه ۲ بعدی است، ولی ممکن است چند بعد هم داشته باشیم که همش آرایه چند بعدی میشود. اولین کاری که ما باید انجام دهیم:

WELDER QUALIFICATION MATRIX

Issue : 5
Revision : 0
Date :20 01 2011

WR No	WELDER NAME	WPS No	MATERIALS	PROCESS	F Nos. Qualified	POS	THICKNESS QUALIFIED	DIA QUALIFIED
							GTAW - 8mm max/ 14.14mm max SMAW	
W 8002	Talari Raju	QME 01	Carbon Steel - ASME IX	GTAW/SMAW	All F No.6	6G		OD 25mm & above
W 8002	Talari Raju	QME 06	CuNi -ASME IX	GTAW	F No.34 & All F No.41 through F No.46	6G	6.0 mm maximum	OD 73 mm & above
W 8005	Santhosh Kumar P	QME 17	Carbon Steel Group I & II of AWS D1.1	FCAW	All F No.6	3G	3mm to Unlimited	24" OD & Above
W 8005	Santhosh Kumar Y	QME 17	Carbon Steel Group I & II of AWS D1.1	FCAW	All F No.6	3G	3mm to Unlimited	24" OD & Above
W 8005	Kanakaraju	GMS/WPS/MS-25	Carbon Steel Group I & II of AWS D1.1	SMAW	F No 1 to 4	3G	5mm to Unlimited	24" OD & Above
W 8007	Satheesh Kumar	QME 03	Carbon Steel Group I & II of AWS D1.1	SMAW	F No.1 to 4	3G	5mm to Unlimited	24" OD & Above
W 8007	Musharaf Ali	QME 16	Carbon Steel - ASME IX	FCAW	All F No.6	3G	Unlimited	24" OD & Above
W 8007	Musharaf Ali	QME 16	Carbon Steel - ASME IX	FCAW	All F No.6	6G	Unlimited	27/8" OD & Above
W 8007	Musharaf Ali	QME 17	Carbon Steel Group I & II of AWS D1.1	FCAW	All F No.6	6G	Unlimited	27/8" OD & Above
W 8009	Murugan Rajendran	QME 16	Carbon Steel - ASME IX	FCAW	All F No.6	3G	Unlimited	24" OD & Above
W 8009	Murugan Rajendran	QME 03	Carbon Steel Group I & II of AWS D1.1	SMAW	F No 1 to 4	6G	22mm max	4" OD & Above
W 8009	Murugan Rajendran	QME 01	Carbon Steel - ASME IX	SMAW	F No 1 to 4	6G	16mm max	27/8" OD & Above
W 8002	Murugan Rajendran	QME 17	Carbon Steel Group I & II of AWS D1.1	FCAW	All F No 6	3G	Unlimited	24" OD & Above
W 8012	Reji George	QME 03	Carbon Steel Group I & II of AWS D1.1	SMAW	F No 1 to 4	3G	5mm to Unlimited	24" OD & Above
W 8012	Davidson Sair	QME 16	Carbon Steel - ASME IX	FCAW	All F No.6	3G	Unlimited	24" OD & Above
W 8016	Habib Rahman	QME 16	Carbon Steel - ASME IX	FCAW	All F No.6	3G	Unlimited	24" OD & Above
W 8016	Sivanesan Nadar	QME 05	Stainless Steel -ASME IX	GTAW	All F No.6	6G	11.0mm maximum	OD 25mm & above
W 8016	Sivanesan Nadar	QME 05	CuNI -ASME IX	GTAW	F No.34 & All F No.41 through F No.46	6G	6.0 mm maximum	OD 73 mm & above
W 8016	B Nageswara Rao	QME 03	Carbon Steel - ASME IX	GTAW/SMAW	All F No 6 / All F 1 to F4	6G	GTAW - 12mm max/ SMAW-Unlimited	27/8" OD & Above

حالا فرض کنید داده های Iris را بخواهیم برایشان یک جدول بکشیم!! آن وقت خواهیم شنوی یک جدول بکشیم. یک جدول است و جدول خیلی جالب نیست، میتوانیم دستشو ترهم بکنیم، همان نرم افزار spss میتواند بهمش بریزیم متغیرهایش کدامش را در سطر بگذارد و کدام هایش را در ستون، تا مشت ترشده. مثلاً اینجا ما ۳ تا متغیر داریم، متغیر اولمان ۳ تا حالت دارد، متغیر دوم سال ۳ تا حالت دارد و متغیر سوم مان هم ۳ تا حالت دارد، حالا براساس متغیر اول می آیم ۹ تا کم و متوسط و زیاد مینویسم، توی متغیر دوم مان دوباره کم، ممکن است کم باشد، برای کم ممکن است medium باشد، زیاد ندارد اصلاً، یعنی یک حالت اصلاً وجود ندارد. متغیر سوم آن را هم بنویسم اما حذفی شود و همینطوری بیایم پایین و این هم متغیر سوم مان و دوباره برای بار دوم الیم آن بدترین بار ترین نوع نمایش این داده هاست.

Example: Iris data

- We show how the attributes, petal length, petal width, and species type can be converted to a multidimensional array
  - First, we discretized the petal width and length to have categorical values: *low*, *medium*, and *high*
  - We get the following table - note the count attribute

Petal Length	Petal Width	Species Type	Count
low	low	Setosa	46
low	medium	Setosa	2
medium	low	Setosa	2
medium	medium	Versicolour	43
medium	high	Versicolour	3
medium	high	Virginica	3
high	medium	Versicolour	2
high	medium	Virginica	3
high	high	Versicolour	2
high	high	Virginica	44

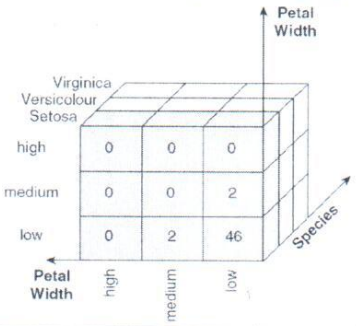
OLAP

پس این همه افزارهای ما است

همین کار بالا را حاضر است دوتا یک طلب بکنیم که این صفرهای توی بین جاهایی است که در جدول بالا اصلا نشانش ندادیم ولی اینجا نمیتوانیم از زیرش دربریم چون طلب خالی نیست، پس ارتباطی که وجود ندارند باید جای تعدادشان صفر بگذاریم و نشان دهیم ولی هم داده ها با این روش قابل خواندن نیستند.

Example: Iris data (continued)

- Each unique tuple of petal width, petal length, and species type identifies one element of the array.
- This element is assigned the corresponding count value.
- The figure illustrates the result.
- All non-specified tuples are 0.



درنرم افزار قابل خواندن هستند اما دوست ندارم. راه دومی که قبلا هم گفتیم میتوانیم بیاییم شروع کنیم روی یکی از متغیرها آنرا شروع کنیم روی یکی از متغیرها، آن وقت میتوانیم آن را به رشته تبدیل کنیم، یعنی شروع کنیم روی یک متغیر و حالا هم جدول دوبعدی رسم میکنیم. این جدول دوبعدی سمت راست و اینجا تعداد صفرهای مازادی موجود، آن هایی که اصلا وجود نداشتند. پس برای داده هایی به صورت سه بعدی هستند حالا میتوانیم یک همچین نموداری رسم کنیم (اسلاید 35) و هنری که ما به خرج می دهیم این است که شروع کنیم روی چه متغیر؟! بعضی وقتها مجبور میشوم تمام شکل ها را بکشم، این جداول مارا (وقتی نرم افزار می گویم یا شروع کدام متغیر را، از خواص برای ما میماند لازم نیست ما به همه این ها بپردازیم چطور این جداول را بکشیم؛ ما فقط متغیری که شروع کنیم و بشکنیمش را باید بدانیم. هرکدام که بهتر بود دوباره تمرین را بسته مربوط انتخاب میکنیم برای گزارش خودمان.

Example: Iris data (continued)

- Slices of the multidimensional array are shown by the following cross-tabulations
- What do these tables tell us?

Length \ Width	low	medium	high
low	46	2	0
medium	2	0	0
high	0	0	0

Length \ Width	low	medium	high
low	0	0	0
medium	0	43	3
high	0	2	2

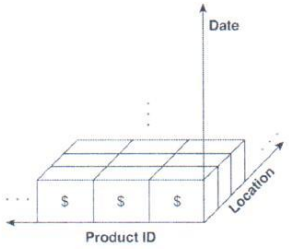
Length \ Width	low	medium	high
low	0	0	0
medium	0	0	3
high	0	3	44

OLAP Operations: Data Cube

- The key operation of a OLAP is the formation of a data cube
- A data cube is a multidimensional representation of data, together with all possible aggregates.
- By all possible aggregates, we mean the aggregates that result by selecting a proper subset of the dimensions and summing over all remaining dimensions.
- For example, if we choose the species type dimension of the Iris data and sum over all other dimensions, the result will be a one-dimensional entry with three entries, each of which gives the number of flowers of each type.

Data Cube Example

- Consider a data set that records the sales of products at a number of company stores at various dates.
- This data can be represented as a 3 dimensional array
- There are 3 two-dimensional aggregates (3 choose 2), 3 one-dimensional aggregates, and 1 zero-dimensional aggregate (the overall total)



Data Cube Example (continued)

- The following figure table shows one of the two dimensional aggregates, along with two of the one-dimensional aggregates, and the overall total

	date				
product ID	Jan 1, 2004	Jan 2, 2004	...	Dec 31, 2004	total
1	$1,001	$987	...	$891	$370,000
⋮	⋮			⋮	⋮
27	$10,265	$10,225	...	$9,325	$3,800,020
⋮	⋮			⋮	⋮
total	$527,362	$532,953	...	$631,221	$227,352,127

OLAP Operations: Slicing and Dicing

- Slicing is selecting a group of cells from the entire multidimensional array by specifying a specific value for one or more dimensions.
- Dicing involves selecting a subset of cells by specifying a range of attribute values.
  - This is equivalent to defining a subarray from the complete array.
- In practice, both operations can also be accompanied by aggregation over some dimensions.

OLAP Operations: Roll-up and Drill-down

- Attribute values often have a hierarchical structure.
  - Each date is associated with a year, month, and week.
  - A location is associated with a continent, country, state (province, etc.), and city.
  - Products can be divided into various categories, such as clothing, electronics, and furniture.
- Note that these categories often nest and form a tree or lattice
  - A year contains months which contains day
  - A country contains a state which contains a city

تقسیم
OLAP اصلاً کاربرد داده‌ها نداره‌م باید تبدیلشان کنیم، نمایشان کنیم، داده‌هایی هستند، معمولاً داده‌های
حد مرکزی نمایشان می‌کنیم حالت عادی اصلاً نمایش نیست. OLAP باید تنها خلاصه می‌کند داده‌ها را و بعد حالا
از این خلاصه‌ها چه استفاده‌ای می‌شود؟ ما با استفاده از این خلاصه‌سازی می‌توانیم تست‌هایی ارائه دهیم که رابطه‌ای بین
متغیرها را در بیاوریم مثلاً وقتی بچه‌ها دوبیدی است مایک تعلق می‌توانیم بدهیم آیا سرطان به سیگار وابسته هست یا خیر؟!
مستقل اند یا مستقل نیستند؛ ولی اگر جدول ما 7 بعدی شود، 24000 حرف داریم برای جدول 7 بعدی، وقتی جدول ابعادی

OLAP Operations: Roll-up and Drill-down

- This hierarchical structure gives rise to the roll-up and drill-down operations.
  - For sales data, we can aggregate (roll up) the sales across all the dates in a month.
  - Conversely, given a view of the data where the time dimension is broken into months, we could split the monthly sales totals (drill down) into daily sales totals.
  - Likewise, we can drill down or roll up on the location or product ID attributes.

است ، باید مدل میتوانیم ارزش بدهیم بر روی داده‌هایی ، و لی آخر 7 بعدی شد ما میتوانیم 24000 خرمنیرا آزمون کنیم
(مورد مدل 7 بعدی . یعنی وقتی بعد بالا می رود تعداد خرمنیم های که ما میتوانیم مطرح کنیم خیلی زیاد است ، اینجا داده‌های
(مورد داده‌ها
بیک روش های چند متغیره کنیم مطرح میشود.

Data mining ادامه جلسه ۶

در جلسه قبل در مورد خوشه بندی صحبت کردیم، کلستر کلستر یک روش unsupervised است که هدفش قرار دادن داده های همگون توی یک خوشه است. با قدم شروع کردیم، گفتیم که درختی و منهوا هم فاصله دوتا داده را از هم حساب کنیم، مثل صحبتهای که در اسلاید های درس کردیم، حالا ما آیم با یک متر فاصله آنها را اندازه گیری میکنیم، اگر داده هایمان یک بعدی بودند، این متر ها خیلی ساده بود، همان متر معمولی، فاصله ی بین دو تا نقطه را چگونه حساب می کردیم؟! از هم کم شان میکردیم، یعنی ۲ تا را میگرفتیم بعد جدرش میگرفتیم و میگفتیم این فاصله ی دو نقطه شده. ولی دقتی برداری بودند، یعنی نقطه ما از یک بعد بیشتر داشت که مختصات داده ها ی ما این چنین هستند، گفتیم باید بیاییم مولفه و ابزار این کار را پیدا کنیم، یعنی مولفه های x دو نقطه را از هم کم کنیم، توان ۲ بسانیم + مولفه های y هم همین کار، اگر مولفه z یا t و ... هم داشت باشند، همین کار را برای بقیه هم انجام دهیم و بعد بیاییم توان ۲ بسانیم و بعد با هم جمع شان کنیم و در نهایت جدرش بگیریم، این برای p بعدی است، پس فاصله ی دو نقطه x و y که هر کدام p بعد دارند، اینجوری حساب میشود، این فاصله اقلیدسی است. ارها norm minkusvky میخواستیم این بعدم، minkusvkiy با پارامتر ۲ (چون ۲ فرض کردیم) دیگر جذرهم نمی خواست بگیریم، بعد میگفتیم با استفاده از این ابزارها میتوانیم کارهای خیلی خوبی هم انجام می دهیم، البته بعد متر های دیگر تعریف کردیم، که انواع و اقسام متر ها را گفتیم، براس متر اقلیدسی اگر ما n تا نقطه p بعدی داشته باشیم، می توانیم ماتریس فواصلشان را حساب کنیم البته در وقتی با نرم افزار elementin این ماتریس وجود ندارد، به جای ماتریس چون فقط باید از این ماتریس که فرضا ۹ تا عضو دارد این سه تا پایین را آنالیز کنیم، بصورت برداری ما این را ۹ تایی می دهیم یعنی بردار فاصله را به ما ۹ می دهد نرم افزار ماتریس فاصله ها، حتما نیم باید تبدیل این بردار را به ماتریس تبدیل کنیم، این کار خیلی نیست. ماتریس فاصله ها چه دردی مان میخورد؟! وقتی میخواهیم خوشه بندی کنیم، این خوشه بندی ما اگر جمع شونده باشد، یعنی بیاییم داده ها (صفحه 474 هستیم) .....

clustering با استفاده از روش های سلسله مراتبی انجام بدهیم، ۲ راه عمده داریم: ۱- بیاییم تمام داده ها را به صورت چند تا خوشه در نظر بگیریم یعنی به اندازه داده ها خوشه تعریف کنیم بعد سعی کنیم با Merg کردن این داده ها، نهایتا به یک خوشه برسیم، باین میگیم روش جمع شونده یا اینکه برعکس همه داده ها را یک خوشه بگیریم و بعد سعی کنیم داده ها را از هم دسته جدا کنیم، به این روش میگوییم روش تقسیم شونده یا (divi sin). پس در روش سلسله مراتبی دقت کنید روش ما supervise نیست، ما فقط داریم چند تا خوشه با یه داده های ما تبدیل میشود، ما اطلاعی نداریم که ظاهرهش می آیم تمام حالتها را در نظر میگیریم و بدنبالش میگیم با استفاده از یک روشی بهترین خوشه را جدا کنیم، ولی این کار را باید انجام دهیم یعنی تمام داده ها به اندازه تمام داده ها n تا داده داریم که n خوشه، نهایتا باید برسیم به یک خوشه، یا یک خوشه داریم برسیم به n خوشه و نهایتا جایی که خوشه ایمان را از هم بگیریم (۱)

کلاستر شخص داد که کجا بهترین خوشه را با ما می‌دهد مشخص کنیم، خاطر همین بهش میگویند clustering. اگر بعد از خوشه‌ها توی روش segmentation مشخص باشد، دیگه نیاز به [illegible] ما می‌آیم سراغ خیلی که با بایاس، یک ضرب در ۲، مرحله مرحله می‌آید، برای روش تقسیم شونده راهه داره. ما داریم یک خوشه، بعد یک خوشه را تقسیم میکنیم به دو، بعد دو خوشه، بعد ۳ خوشه، بعد ۴ خوشه و الی آخر تا n خوشه که n تا داده‌ها ما است. محاسبات باید برای همه داده‌ها، انجام شود (این روش پارتیشنینگ نیست و متفاوت است) این کار را که انجام می‌دهیم، داده‌ها را تقسیم می‌کنیم، محاسبات وحشتناکی داریم چون در هر مرحله باید فاصله‌ی همه‌ی داده‌ها را از هم آنها حساب کنیم، توی مرحله اول تقسیم می‌کنیم به آنها فاصله‌ی داده‌ها را از هم‌دیگر حساب می‌کنیم، فاصله نقاط از هم‌دیگر فرض کنید روش ما جمع‌شونده باشد، اگر روش ما جمع‌شونده باشد کافیه هر یک از داده‌ها را یک خوشه در نظر بگیریم، پس این شد n تا خوشه، منتهی هم دو تا از این‌ها را با هم Merg کنیم، توی هر مرحله یکی از تعداد خوشه‌ها یکی کم میشود، خاطر همین محاسبات این روش خیلی سنگین است. توی مرحله اول می‌آیم داده‌ها را فاصله‌هایشان را با هم حساب میکنیم، توی ماتریس D ذخیره میکنیم، آن دو تا داده‌ای که از هم کمترین فاصله‌ها را دارند Merg شان می‌کنیم، میشن یک خوشه، در مرحله بعد منتهی هم بیایم یک خوشه داریم که دو تا عضو دارد، بقیه خوشه‌ها یا یک عضو دارند، منتهی هم فاصله اینها را حساب کنیم، ما تا الان بلد بودیم فاصله نقاط را حساب کنیم، فاصله نقطه از مجموعه را بلد نیستیم حساب کنیم، پس نیاز به یک تعریف جدید داریم، سنگین فاصله یک نقطه از مجموعه را شما تعریف Unik ای ندارید، چندین تعریف داریم، یکی از تعاریف، تعریف این است که فاصله‌ی یک نقطه از خوشه را ما تعریف کنیم: کمترین فاصله‌ی نقطه تا اعضای آن خوشه، یعنی بیایم فاصله‌ی تمام اعضایی که بیرون از خوشه هستند را با داخل خوشه‌ها حساب کنیم، اگر که مینیمم‌شان را بگیریم، این را به عنوان فاصله تعریف میکنیم، پس فاصله‌ی دو تا مجموعه را، یا یک نقطه از یک مجموعه را، تعریف می‌کنیم: فاصله‌ی تمام نقاط‌شان از هم را مینیمم‌شان را به عنوان فاصله در نظر میگیریم، پس یک مجموعه یک محاسبات سنگین‌ها اینجا دارم، تمام ۷ها یی که در ۸ هستند و تمام ۷ها یی که در ۷ هستند، ۷ تا دو تا مجموعه است، دو تا خوشه است، حالا در مرحله دوم، ۸ ما یک عضو بیشتر ندارد، پس وقتی آمدم فاصله عضو ۸ را با تمام فاصله‌ی اعضای ۷ حساب میکنم، مینیمم‌اش را می‌گیرم، کمترین مقداری که پیدا است آن را به عنوان فاصله میگذاریم و به این میگیم minimum Distance، یا روش‌مان میشود single linkage. پس ما روش‌های مختلفی داریم برای این‌که فاصله را چی تعریف کنیم، اسم روش‌مان عوض می‌شود، پس single linkage یا nearist nigbor دنیای اش یکی است، به روشی میگیم فاصله‌ی نقاط، توی دو تا مجموعه را حساب میکنیم به عنوان فاصله تعریف میکنیم. (صفحه 477 کتاب مثال خوانده شود)

دندوگرام چیست؟ دندوگرام همان مراحل خوشه بندی است، سوال این ماخوشه را کجا بکشیم؟! این سوال اصلی در خوشه بندی است

single linkage شاید ساده ترین روش باشد اما بهترین روش در کل نیست، این روش مشکلاتی دارد

چه جوری بهترین خوشه را انتخاب کنیم؟ گفتیم ۲ راه دارد، یکی از قوانین آمار دان، آقای موجانا یک قاعده داده، شیم ما براساس داده هایمان، ۲ تا پارامتر را محاسبه کنید و دید عدد ضرب کنید و بعد فاصله از کجا نمودار خوشه بندی را ببریم مشخص است.

فاصله این خوشه ها را براساس minimum Distance ها است، ولی عقل سلیم میگه خوشه خوب خوشه ای است که واریانس درون خوشه cluster کم باشد، فاصله بین خوشه ها زیاد باشد، پس ما باید واریانس نقاط داخل خوشه ها را حساب کنیم، و واریانس بین خوشه ها را هم حساب کنیم، آنرا بر مجموع، فاصله ی داخلی ها کم باشد و بیرونی ها زیاد باشد، آن خوشه بهترین خوشه است. ما خوشه ای را انتخاب می کنیم که بیشترین واریانس بین خوشه ها داشته باشد و کمترین واریانس در داخل خوشه ها، حالا این روش این است که این واریانس ها دوباره یک توزیع آماری دارند، میتوانیم براساس آن توزیع مثلاً F است، آزمون فرض انجام دهیم و خوشه ها را انتخاب کنیم. آخر کار ما این کار را انجام می دهیم. تو امتحان حتماً یک سوال می دهم که با این روش خوشه بندی کنیم (تنگ نگیند) آماری که

بعضی وقتها ما میتوانیم فاصله را طوری دیگه حساب کنیم، که بهش میگوییم farther neighbor (complete linkage) آنوقت برای دو مجموعه فاصله شان را به صورت ماکزیمم تعریف کنیم، ماکزیمم اعضایشان را، از این روش بخواهم حالا بروم، روش ان حالا روش جمع شونده است، به این روش میگوییم complete linkage. پس این روش براساس ماکزیمم حساب میکند فاصله را. خیلی وقتها که از این دو روش استفاده میکنیم، باید جواب بدیم، همان مثال قبلی، دقت کنید در حالت اول که ماتریس Pistance ها را حساب میکنیم، در complete linkage, single linkage عیناً مثل هم میشود هیچ فرقی نمیکند در مرحله بعد در مرحله دوم تا اینجا هیچ فرقی نکرده است مثل مثال قبلی است، ولی از اینجا به بعد فواصل با هم فرق می کنند و فاصله ها حساب می کنیم باید max باشد، max نقاط را بگیریم و max ها را به عنوان فاصله ها در نظر بگیریم. (61) پس اول مرحله min می گیریم و بعد مرحله max می گیریم، مثالی هم در امتحان می دهم، مثال جمع جوری خواهد بود و با دست می توانیم حسابش کنیم.

عقل سلیم حکم می کند که نه min خوب است نه max، می گویند ما بیایم متوسط فاصله ها را بگیریم به عنوان فاصله A و B بیایم فاصله نقاط را با هم حساب کنیم، بعد متوسط اش را به عنوان فاصله بگیریم، به بهش میگوییم avrage linkage که این با عقل سلیم بیشتر جور در می آید و بعضی اوقات ها واقعاً توزیع داده ها نرمال باشد، توزیع معقولی داشته باشند و داده ی پرتی وجود نداشته باشد، این روش از دو روش قبلی جواب بهتری می دهد، به k-means نزدیک است و با آن قبلی فرق میکند، شاید فاصله اش به k-means باشد، k-means یک الگوریتم ریاضی است، برعکس EM جلو می رود، جوابش در یک حالت خاص با k-means یکی میشود، ولی با k-means در حالت کلی تفاوت است، یعنی میشود که یکی فرض ها را

روی k-means گذاشت که با جواز average linkage بی جواب دادیم بزرگ بود ولی در حالت کلی بزرگ نیست.

حالا محاسباتی که ما در اینجا داریم (average linkage) یه محاسبات وحشتناکی است، بخاطر همین بعضی‌ها به جای این روش می‌آیند از Centroid استفاده می‌کنند، که در این روش به جای اینکه میانگین فواصل رو حساب کنند، فاصله میانگین‌ها را می‌گیرند. این دو تا با هم کاملاً متفاوت است. پس در Centroid داده‌ها را component wise میانگین حساب می‌کنیم، میدانیم در یک بردار، امین مؤلفه $\bar{y}_A$ (مثلاً y تا نقطه داره) بعد میایم تمام نقاطی که توی یک مجموعه A قرار گرفتند، مؤلفه‌ی اولشان را با هم جمع می‌کنیم، بخش بر تعدادشان می‌کنیم، تعدادشان n_A تا است، پس عضو امین میانگین می‌گیریم. اسمش می‌ذاریم $\bar{y}_A$ (صفحه 403 کتاب) و برای $\bar{y}_B$ هم، p تا میانگین می‌گیریم از n^2 تا داده، مثلاً برای A n_1 تا داده داریم برای B n_2 تا داده داریم و حالا می‌آیم فاصله میانگین‌ها را حساب می‌کنیم که از روش قبلی محاسبات خیلی سبک‌تر است، که فرمول‌ها به این صورت هستند:

$$D(A,B) = d(\bar{y}_A, \bar{y}_B)$$

$$\bar{y}_{AB} = \frac{n_A \bar{y}_A + n_B \bar{y}_B}{n_A + n_B} \quad , \quad \bar{y}_A = \sum_{i=1}^{n_A} \frac{y_i}{n_A}$$

دقت کنید بخاطر این سبک‌تر است که میانگین را میانگین وزنی است، وقتی اگر خواستید میانگین A و B را حساب کنید نزدید میانگین را از اول بگیرید چون میانگین یک میانگین وزنی است، ما بیایم تعداد اعضا‌ها را در میانگین ضرب کنیم بعد اضافه تعداد اعضا‌ها را میانگین بخش بر تعداد کل این کنیم، این بردار جدید به دست می‌آورد.

میانگین یک بدی داشت که آن این است که نسبت به داده‌های پرت حساس است به جای اینکه از میانگین بیایم استفاده کنیم خیلی وقتا بخاطر این بدی این از میانه استفاده کنیم. در صفحه 486 منظور از $\bar{y}_A$ و $\bar{y}_B$ میانه است نه میانگین.

Ward's Method چیست؟ در این متد آمدیم SSE را تعریف کردیم، SSEA و داریاس که می‌خواهیم حساب کنیم، واریانس یعنی پراکندگی داده‌ها حول میانگین، فرمول واریانس را همه بلدیم مثلاً

$$S_2 = \frac{1}{(n-1)\sum (x_i - \bar{x})^2}$$

این برای وقتی است که داده‌های ما اسکالر هستند، اگر داده‌های ما بردار باشند، چطور می‌کنیم؟ می‌آییم $(x_i-\bar{x})$ را $(x_i-\bar{x})'$ ضرب می‌کنیم، کاری که اینجا هم انجام داده هستیم است، (ص 486) پس اگر بخواهیم واریانس را تعاریف کنیم مجموعه A را حساب کنیم، باید بیایم آن مجموعه را، آن بردار را صفحه‌ای حساب کنیم که این میانگین، میانگین مؤلفه به مؤلفه است، پس SSEA، SSE_{13} و SSE_{24} را به همین صورت بدست می‌آوریم، حالا ما می‌توانیم SSE_{AB} را هم تعریف کنیم، یعنی دوتای اولی SSE واریانس درون کلاس‌ها هستند، اگر تقسیم بر تعداد اون کنیم میشه واریانس، SSE_B یعنی توان دوم خطاهای مجموعه‌ای B، این هم واریانس داخل شاخه است، حالا می‌توانیم SSE_{AB} را پیدا کنیم، یعنی بین A و B و حقیقتا هم اختلاف دارند

$$SSE_A = \sum_{i=1}^{nA} (y_i - \bar{y}_A)'(y_i - \bar{y}_A)$$

$$SSE_B = \sum_{i=1}^{nB} (y_i - \bar{y}_B)'(y_i - \bar{y}_B)$$

$$SSE_{AB} = \sum_{i=1}^{nAB} (y_i - \bar{y}_{AB})'(y_i - \bar{y}_{AB})$$

Transpose

میانگین‌گیری

$$\bar{y}_{AB} = \frac{n_A \bar{y}_A + n_B \bar{y}_B}{n_A + n_B}, \quad n_{AB} = n_A + n_B \leftarrow$$ مجموع اعضای B, A

روش پیشنهادی ward می‌گفتیم وقتی ما دوتا کلاستر را با هم Merg می‌کنیم اگر SS بینشان کم باشد یعنی شبیه هم هستند یا به عبارتی واریانس بین A, B کم باشد. پس I_{AB} را به صورت زیر تعریف می‌کنیم و می‌گفتیم ما I_{AB} را کمینه می‌کنیم.

$$I_{AB} = SSE_{AB} - (SSE_A + SSE_B)$$

اگر I_{AB} کوچک شد یعنی این دوتا (A, B) باید با هم Merg شوند. با این روش می‌توانیم همین کار را انجام دهیم و این دندروگرام حاصل می‌شود. I_{AB} را می‌توان ساده کرد و به این روابط رسید که البته در اینجا اثبات روابط ساده و کم‌پیچیده‌تر خواهد بود اما در نرم‌افزارها به راحتی است. می‌توان نشان داد که فرمول ساده شده‌ای از صفحه 488 دوی اشتراس هست که البته اثبات شده است که درست است این روش‌ها فقط است، در واقع همان روش centroid و منظور ما از square error lice است و آنرا به صورت اقلیدسی استفاده می‌کنیم.