CS(STAT)5525 : Data Analytics
Lecture #5

Reza Jafari, Ph.D

Collegiate Associate Professor
rjafari@vt.edu

COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH.

Feature Transformation

- A feature transformation refers to a transformation that is applied to all values of a feature.

Feature Transformation

- A feature transformation refers to a transformation that is applied to all values of a feature.
- E.g, if only magnitude of a variable is important, then the values of feature can be transformed by taking absolute value.

Feature Transformation

- A feature transformation refers to a transformation that is applied to all values of a feature.
- E.g, if only magnitude of a variable is important, then the values of feature can be transformed by taking absolute value.
- Three important feature transformations are:

Transformation

Feature Transformation

- A feature transformation refers to a transformation that is applied to all values of a feature.
- E.g, if only magnitude of a variable is important, then the values of feature can be transformed by taking absolute value.
- Three important feature transformations are:

Transformation

1. Simple Functional transformation

Feature Transformation

- A feature transformation refers to a transformation that is applied to all values of a feature.
- E.g, if only magnitude of a variable is important, then the values of feature can be transformed by taking absolute value.
- Three important feature transformations are:

Transformation

1. Simple Functional transformation
2. Standardization

Feature Transformation

- A feature transformation refers to a transformation that is applied to all values of a feature.
- E.g, if only magnitude of a variable is important, then the values of feature can be transformed by taking absolute value.
- Three important feature transformations are:

Transformation

1. Simple Functional transformation
2. Standardization
3. Normalization

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.
- Variable transformation should be applied with caution because they change the nature of the data.

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.
- Variable transformation should be applied with caution because they change the nature of the data.
- E.g, $\frac{1}{x}$ reduces the magnitude of values that are 1 or larger but reduces the magnitude of values between 0 and 1.

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.
- Variable transformation should be applied with caution because they change the nature of the data.
- E.g, $\frac{1}{x}$ reduces the magnitude of values that are 1 or larger but reduces the magnitude of values between 0 and 1.
- Important questions to ask:

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.
- Variable transformation should be applied with caution because they change the nature of the data.
- E.g, $\frac{1}{x}$ reduces the magnitude of values that are 1 or larger but reduces the magnitude of values between 0 and 1.
- Important questions to ask:
  1. What is the desired property of transformed attribute?

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.
- Variable transformation should be applied with caution because they change the nature of the data.
- E.g, $\frac{1}{x}$ reduces the magnitude of values that are 1 or larger but reduces the magnitude of values between 0 and 1.
- Important questions to ask:
  1. What is the desired property of transformed attribute?
  2. Does the order need to be maintained?

Simple Functions

1- Simple functions

- Simple mathematical function applied to each value individually: x^k, $\log x$, $\exp^x$, $\sqrt{x}$, $\frac{1}{x}$
- Suppose the feature of interest is the number of data bytes ranges from 1 to 1 billion. The log transformation is advantageous to compress the range from 0 to 9.
- Variable transformation should be applied with caution because they change the nature of the data.
- E.g, $\frac{1}{x}$ reduces the magnitude of values that are 1 or larger but reduces the magnitude of values between 0 and 1.
- Important questions to ask:
  1. What is the desired property of transformed attribute?
  2. Does the order need to be maintained?
  3. Does the transformation apply to all values especially negative values and 0?

Standardization or Normalization

- Consider comparing two features: people age & income.

Standardization or Normalization

- Consider comparing two features: people age & income.
- For any two people, the difference in income will likely be much higher than age.

Standardization or Normalization

- Consider comparing two features: people age & income.
- For any two people, the difference in income will likely be much higher than age.
- Without considering the differences in the range of values of age and income, the comparison between people will be dominated by income.

Standardization or Normalization

- Consider comparing two features: people age & income.
- For any two people, the difference in income will likely be much higher than age.
- Without considering the differences in the range of values of age and income, the comparison between people will be dominated by income.
- If similarity or dissimilarity of two people is calculated, the income values will be dominated.

Standardization or Normalization

- Consider comparing two features: people age & income.
- For any two people, the difference in income will likely be much higher than age.
- Without considering the differences in the range of values of age and income, the comparison between people will be dominated by income.
- If similarity or dissimilarity of two people is calculated, the income values will be dominated.
- There are two approaches for the feature rescaling:

2- Standardization

3- Normalization

Standardization

- Standardization is a feature scaling technique that rescales dataset to the zero mean and variance of 1.

Standardization

- Standardization is a feature scaling technique that rescales dataset to the zero mean and variance of 1.
- Several machine learning algorithms i.e. RBF kernel of Support Vector machine or L1 and L2 regularizer of linear models require standard normally distributed data.

Standardization

Standardization

- Standardization is a feature scaling technique that rescales dataset to the zero mean and variance of 1.
- Several machine learning algorithms i.e. RBF kernel of Support Vector machine or L1 and L2 regularizer of linear models require standard normally distributed data.

Standardization

- Standardization rescales a dataset to have a mean of 0 and standard deviation of 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - \overline{x}}{\sigma_x}$$

Standardization

- Standardization is a feature scaling technique that rescales dataset to the zero mean and variance of 1.
- Several machine learning algorithms i.e. RBF kernel of Support Vector machine or L1 and L2 regularizer of linear models require standard normally distributed data.

Standardization

- Standardization rescales a dataset to have a mean of 0 and standard deviation of 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - \overline{x}}{\sigma_x}$$

- x_i is the i^{th} value in the dataset.

Standardization

- Standardization is a feature scaling technique that rescales dataset to the zero mean and variance of 1.
- Several machine learning algorithms i.e. RBF kernel of Support Vector machine or L1 and L2 regularizer of linear models require standard normally distributed data.

Standardization

- Standardization rescales a dataset to have a mean of 0 and standard deviation of 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - \overline{x}}{\sigma_x}$$

- x_i is the i^{th} value in the dataset.
- $\overline{x}$ is the sample mean.

Standardization

- Standardization is a feature scaling technique that rescales dataset to the zero mean and variance of 1.
- Several machine learning algorithms i.e. RBF kernel of Support Vector machine or L1 and L2 regularizer of linear models require standard normally distributed data.

Standardization

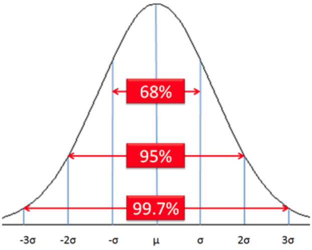
- Standardization rescales a dataset to have a mean of 0 and standard deviation of 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - \overline{x}}{\sigma_x}$$

- x_i is the i^{th} value in the dataset.
- $\overline{x}$ is the sample mean.
- σ_x is the sample standard deviation .

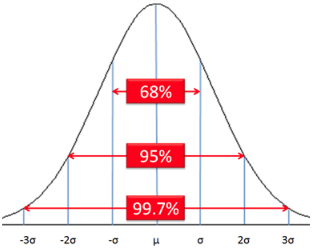
Standardization

- Any normal distribution with any value of mean (μ) and standard deviation (σ) can be transformed into the standard normal distribution, where the mean of zero and a standard deviation of 1. This is also called standardization.



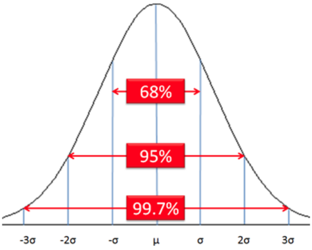
Standardization

- Any normal distribution with any value of mean (μ) and standard deviation (σ) can be transformed into the standard normal distribution, where the mean of zero and a standard deviation of 1. This is also called standardization.
- 68, 95, 99 rule:



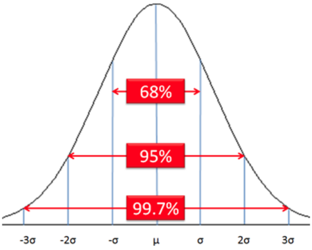
Standardization

- Any normal distribution with any value of mean (μ) and standard deviation (σ) can be transformed into the standard normal distribution, where the mean of zero and a standard deviation of 1. This is also called standardization.
- 68, 95, 99 rule:
  1. 68% of data within 1 std



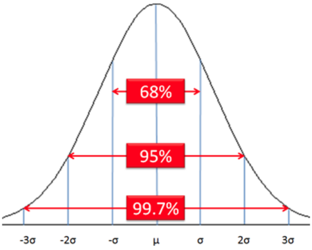
Standardization

- Any normal distribution with any value of mean (μ) and standard deviation (σ) can be transformed into the standard normal distribution, where the mean of zero and a standard deviation of 1. This is also called standardization.
- 68, 95, 99 rule:
  1. 68% of data within 1 std
  2. 95% of data within 2 std



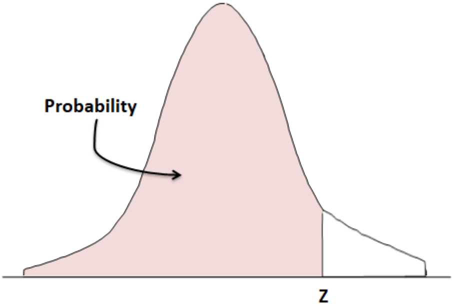
Standardization

- Any normal distribution with any value of mean (μ) and standard deviation (σ) can be transformed into the standard normal distribution, where the mean of zero and a standard deviation of 1. This is also called standardization.
- 68, 95, 99 rule:
  1. 68% of data within 1 std
  2. 95% of data within 2 std
  3. 99% of data within 3 std



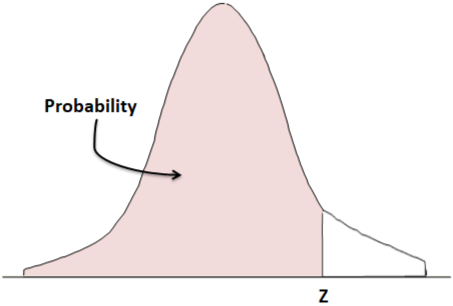
Z-score Table Interpretation

- Most importantly, Z-score helps to calculate how much area that specific Z-score is associated with.



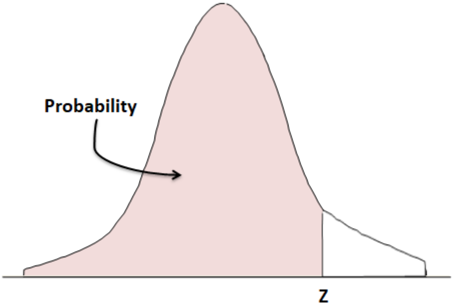
Z-score Table Interpretation

- Most importantly, Z-score helps to calculate how much area that specific Z-score is associated with.
- A Z-score table is also known as a standard normal table used to find the exact area.



Z-score Table Interpretation

- Most importantly, Z-score helps to calculate how much area that specific Z-score is associated with.
- A Z-score table is also known as a standard normal table used to find the exact area.
- Z-score table tells the total quantity of area contained to the left side of any score or value (x).



Z-score Table - Positive z-score

- E.g, z-score $=+1.03$

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
+0	.50000	.50399	.50798	.51197	.51595	.51994	.52392	.52790	.53188	.53586
+0.1	.53983	.54380	.54776	.55172	.55567	.55966	.56360	.56749	.57142	.57535
+0.2	.57926	.58317	.58706	.59095	.59483	.59871	.60257	.60642	.61026	.61409
+0.3	.61791	.62172	.62552	.62930	.63307	.63683	.64058	.64431	.64803	.65173
+0.4	.65542	.65910	.66276	.66640	.67003	.67364	.67724	.68082	.68439	.68793
+0.5	.69146	.69497	.69847	.70194	.70540	.70884	.71226	.71566	.71904	.72240
+0.6	.72575	.72907	.73237	.73565	.73891	.74215	.74537	.74857	.75175	.75490
+0.7	.75804	.76115	.76424	.76730	.77035	.77337	.77637	.77935	.78230	.78524
+0.8	.78814	.79103	.79389	.79673	.79955	.80234	.80511	.80785	.81057	.81327
+0.9	.81594	.81859	.82121	.82381	.82639	.82894	.83147	.83398	.83646	.83891
+1	.84134	.84375	.84614	.84849	.85083	.85314	.85543	.85769	.85993	.86214
+1.1	.86433	.86650	.86864	.87076	.87286	.87493	.87698	.87900	.88100	.88298
+1.2	.88493	.88686	.88877	.89065	.89251	.89435	.89617	.89796	.89973	.90147
+1.3	.90320	.90490	.90658	.90824	.90988	.91149	.91308	.91466	.91621	.91774
+1.4	.91924	.92073	.92220	.92364	.92507	.92647	.92785	.92922	.93056	.93189
+1.5	.93319	.93448	.93574	.93699	.93822	.93943	.94062	.94179	.94295	.94408
+1.6	.94520	.94630	.94738	.94845	.94950	.95053	.95154	.95254	.95352	.95449
+1.7	.95543	.95637	.95728	.95818	.95907	.95994	.96080	.96164	.96246	.96327
+1.8	.96407	.96485	.96562	.96638	.96712	.96784	.96856	.96926	.96995	.97062
+1.9	.97128	.97193	.97257	.97320	.97381	.97441	.97500	.97558	.97615	.97670
+2	.97725	.97778	.97831	.97882	.97932	.97982	.98030	.98077	.98124	.98169
+2.1	.98214	.98257	.98300	.98341	.98382	.98422	.98461	.98500	.98537	.98574
+2.2	.98610	.98645	.98679	.98713	.98745	.98778	.98809	.98840	.98870	.98899
+2.3	.98928	.98956	.98983	.99010	.99036	.99061	.99086	.99111	.99134	.99158
+2.4	.99180	.99202	.99224	.99245	.99266	.99286	.99305	.99324	.99343	.99361
+2.5	.99379	.99396	.99413	.99430	.99446	.99461	.99477	.99492	.99506	.99520
+2.6	.99534	.99547	.99560	.99573	.99585	.99598	.99609	.99621	.99632	.99643
+2.7	.99653	.99664	.99674	.99683	.99693	.99702	.99711	.99720	.99728	.99736
+2.8	.99744	.99752	.99760	.99767	.99774	.99781	.99788	.99795	.99801	.99807
+2.9	.99813	.99819	.99825	.99831	.99836	.99841	.99846	.99851	.99856	.99861
+3	.99865	.99869	.99874	.99878	.99882	.99886	.99889	.99893	.99896	.99900

Z-score Table - Positive z-score

- E.g, z-score =+1.03
- 85% of the data is bellow the number corresponding to above z-score.

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
+0	.50000	.50399	.50798	.51197	.51595	.51994	.52392	.52790	.53188	.53586
+0.1	.53983	.54380	.54776	.55172	.55567	.55966	.56360	.56749	.57142	.57535
+0.2	.57926	.58317	.58706	.59095	.59483	.59871	.60257	.60642	.61026	.61409
+0.3	.61791	.62172	.62552	.62930	.63307	.63683	.64058	.64431	.64803	.65173
+0.4	.65542	.65910	.66276	.66640	.67003	.67364	.67724	.68082	.68439	.68793
+0.5	.69146	.69497	.69847	.70194	.70540	.70884	.71226	.71566	.71904	.72240
+0.6	.72575	.72907	.73237	.73565	.73891	.74215	.74537	.74857	.75175	.75490
+0.7	.75804	.76115	.76424	.76730	.77035	.77337	.77637	.77935	.78230	.78524
+0.8	.78814	.79103	.79389	.79673	.79955	.80234	.80511	.80785	.81057	.81327
+0.9	.81594	.81859	.82121	.82381	.82639	.82894	.83147	.83398	.83646	.83891
+1	.84134	.84375	.84614	.84849	.85083	.85314	.85543	.85769	.85993	.86214
+1.1	.86433	.86650	.86864	.87076	.87286	.87493	.87698	.87900	.88100	.88298
+1.2	.88493	.88686	.88877	.89065	.89251	.89435	.89617	.89796	.89973	.90147
+1.3	.90320	.90490	.90658	.90824	.90988	.91149	.91308	.91466	.91621	.91774
+1.4	.91924	.92073	.92220	.92364	.92507	.92647	.92785	.92922	.93056	.93189
+1.5	.93319	.93448	.93574	.93699	.93822	.93943	.94062	.94179	.94295	.94408
+1.6	.94520	.94630	.94738	.94845	.94950	.95053	.95154	.95254	.95352	.95449
+1.7	.95543	.95637	.95728	.95818	.95907	.95994	.96080	.96164	.96246	.96327
+1.8	.96407	.96485	.96562	.96638	.96712	.96784	.96856	.96926	.96995	.97062
+1.9	.97128	.97193	.97257	.97320	.97381	.97441	.97500	.97558	.97615	.97670
+2	.97725	.97778	.97831	.97882	.97932	.97982	.98030	.98077	.98124	.98169
+2.1	.98214	.98257	.98300	.98341	.98382	.98422	.98461	.98500	.98537	.98574
+2.2	.98610	.98645	.98679	.98713	.98745	.98778	.98809	.98840	.98870	.98899
+2.3	.98928	.98956	.98983	.99010	.99036	.99061	.99086	.99111	.99134	.99158
+2.4	.99180	.99202	.99224	.99245	.99266	.99286	.99305	.99324	.99343	.99361
+2.5	.99379	.99396	.99413	.99430	.99446	.99461	.99477	.99492	.99506	.99520
+2.6	.99534	.99547	.99560	.99573	.99585	.99598	.99609	.99621	.99632	.99643
+2.7	.99653	.99664	.99674	.99683	.99693	.99702	.99711	.99720	.99728	.99736
+2.8	.99744	.99752	.99760	.99767	.99774	.99781	.99788	.99795	.99801	.99807
+2.9	.99813	.99819	.99825	.99831	.99836	.99841	.99846	.99851	.99856	.99861
+3	.99865	.99869	.99874	.99878	.99882	.99886	.99889	.99893	.99896	.99900

Z-score Table - Negative z-score

- E.g, z-score $= -1.03$

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-0	.50000	.49601	.49202	.48803	.48405	.48006	.47608	.47210	.46812	.46414
-0.1	.46017	.45620	.45224	.44828	.44433	.44034	.43640	.43251	.42858	.42465
-0.2	.42074	.41683	.41294	.40905	.40517	.40129	.39743	.39358	.38974	.38591
-0.3	.38209	.37828	.37448	.37070	.36693	.36317	.35942	.35569	.35197	.34827
-0.4	.34458	.34090	.33724	.33360	.32997	.32636	.32276	.31918	.31561	.31207
-0.5	.30854	.30503	.30153	.29806	.29460	.29116	.28774	.28434	.28096	.27760
-0.6	.27425	.27093	.26763	.26435	.26109	.25785	.25463	.25143	.24825	.24510
-0.7	.24196	.23885	.23576	.23270	.22965	.22663	.22363	.22065	.21770	.21476
-0.8	.21186	.20897	.20611	.20327	.20045	.19766	.19489	.19215	.18943	.18673
-0.9	.18406	.18141	.17879	.17619	.17361	.17106	.16853	.16602	.16354	.16109
-1	.15866	.15625	.15386	.15151	.14917	.14686	.14457	.14231	.14007	.13786
-1.1	.13567	.13350	.13136	.12924	.12714	.12507	.12302	.12100	.11900	.11702
-1.2	.11507	.11314	.11123	.10935	.10749	.10565	.10383	.10204	.10027	.09853
-1.3	.09680	.09510	.09342	.09176	.09012	.08851	.08692	.08534	.08379	.08226
-1.4	.08076	.07927	.07780	.07636	.07493	.07353	.07215	.07078	.06944	.06811
-1.5	.06681	.06552	.06426	.06301	.06178	.06057	.05938	.05821	.05705	.05592
-1.6	.05480	.05370	.05262	.05155	.05050	.04947	.04846	.04746	.04648	.04551
-1.7	.04457	.04363	.04272	.04182	.04093	.04006	.03920	.03836	.03754	.03673
-1.8	.03593	.03515	.03438	.03362	.03288	.03216	.03144	.03074	.03005	.02938
-1.9	.02872	.02807	.02743	.02680	.02619	.02559	.02500	.02442	.02385	.02330
-2	.02275	.02222	.02169	.02118	.02068	.02018	.01970	.01923	.01876	.01831
-2.1	.01786	.01743	.01700	.01659	.01618	.01578	.01539	.01500	.01463	.01426
-2.2	.01390	.01355	.01321	.01287	.01255	.01222	.01191	.01160	.01130	.01101
-2.3	.01072	.01044	.01017	.00990	.00964	.00939	.00914	.00889	.00866	.00842
-2.4	.00820	.00798	.00776	.00755	.00734	.00714	.00695	.00676	.00657	.00639
-2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
-2.6	.00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
-2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.00272	.00264
-2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
-2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139
-3	.00135	.00131	.00126	.00122	.00118	.00114	.00111	.00107	.00104	.00100

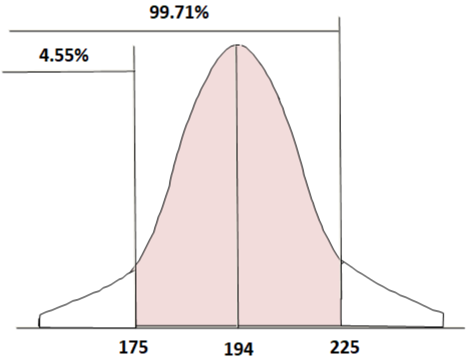
Z-score Table - Negative z-score

- E.g, z-score $= -1.03$
- 15% of the data is bellow the number corresponding to above z-score.

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-0	.50000	.49601	.49202	.48803	.48405	.48006	.47608	.47210	.46812	.46414
-0.1	.46017	.45620	.45224	.44828	.44433	.44034	.43640	.43251	.42858	.42465
-0.2	.42074	.41683	.41294	.40905	.40517	.40129	.39743	.39358	.38974	.38591
-0.3	.38209	.37828	.37448	.37070	.36693	.36317	.35942	.35569	.35197	.34827
-0.4	.34458	.34090	.33724	.33360	.32997	.32636	.32276	.31918	.31561	.31207
-0.5	.30854	.30503	.30153	.29806	.29460	.29116	.28774	.28434	.28096	.27760
-0.6	.27425	.27093	.26763	.26435	.26109	.25785	.25463	.25143	.24825	.24510
-0.7	.24196	.23885	.23576	.23270	.22965	.22663	.22363	.22065	.21770	.21476
-0.8	.21186	.20897	.20611	.20327	.20045	.19766	.19489	.19215	.18943	.18673
-0.9	.18406	.18141	.17879	.17619	.17361	.17106	.16853	.16602	.16354	.16109
-1	.15866	.15625	.15386	.15151	.14917	.14686	.14457	.14231	.14007	.13786
-1.1	.13567	.13350	.13136	.12924	.12714	.12507	.12302	.12100	.11900	.11702
-1.2	.11507	.11314	.11123	.10935	.10749	.10565	.10383	.10204	.10027	.09853
-1.3	.09680	.09510	.09342	.09176	.09012	.08851	.08692	.08534	.08379	.08226
-1.4	.08076	.07927	.07780	.07636	.07493	.07353	.07215	.07078	.06944	.06811
-1.5	.06681	.06552	.06426	.06301	.06178	.06057	.05938	.05821	.05705	.05592
-1.6	.05480	.05370	.05262	.05155	.05050	.04947	.04846	.04746	.04648	.04551
-1.7	.04457	.04363	.04272	.04182	.04093	.04006	.03920	.03836	.03754	.03673
-1.8	.03593	.03515	.03438	.03362	.03288	.03216	.03144	.03074	.03005	.02938
-1.9	.02872	.02807	.02743	.02680	.02619	.02559	.02500	.02442	.02385	.02330
-2	.02275	.02222	.02169	.02118	.02068	.02018	.01970	.01923	.01876	.01831
-2.1	.01786	.01743	.01700	.01659	.01618	.01578	.01539	.01500	.01463	.01426
-2.2	.01390	.01355	.01321	.01287	.01255	.01222	.01191	.01160	.01130	.01101
-2.3	.01072	.01044	.01017	.00990	.00964	.00939	.00914	.00889	.00866	.00842
-2.4	.00820	.00798	.00776	.00755	.00734	.00714	.00695	.00676	.00657	.00639
-2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
-2.6	.00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
-2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.00272	.00264
-2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
-2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139
-3	.00135	.00131	.00126	.00122	.00118	.00114	.00111	.00107	.00104	.00100

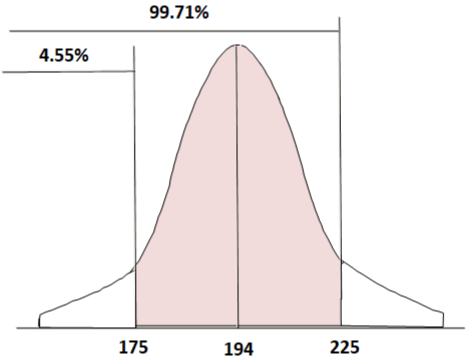
Example

- The weights of 500 American men were taken and the sample mean was found to be 194 pounds with a standard deviation of 11.2 pounds. What percentages of men have weights between 175 and 225 pounds?



Example

- The weights of 500 American men were taken and the sample mean was found to be 194 pounds with a standard deviation of 11.2 pounds. What percentages of men have weights between 175 and 225 pounds?
- Answer : 95%



Standardization - sklearn

- From the sklearn package the .preprocessing.StandardScaler() function can be used to implement the standardization transformation.

```
import numpy as np
from sklearn.preprocessing import StandardScaler

x = np.array([13,16,19,22,23,38,47,56,58,63,65,70,71])

scalar = StandardScaler()

scalar.fit(x.reshape(-1,1))
scalar_transform = scalar.transform(x.reshape(-1,1))
print(f'Standardized data \n : {np.round(scalar_transform,2)}')
```

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

- Normalization rescales a dataset so that each value falls between 0 and 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

- Normalization rescales a dataset so that each value falls between 0 and 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

- x_i is the i^{th} value in the dataset.

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

- Normalization rescales a dataset so that each value falls between 0 and 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

- x_i is the i^{th} value in the dataset.
- x_{min} is the minimum value of x.

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

- Normalization rescales a dataset so that each value falls between 0 and 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

- x_i is the i^{th} value in the dataset.
- x_{min} is the minimum value of x.
- x_{max} is the maximum value of x.

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

- Normalization rescales a dataset so that each value falls between 0 and 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

- x_i is the i^{th} value in the dataset.
- x_{min} is the minimum value of x.
- x_{max} is the maximum value of x.

Normalization

- The goal of Normalization is another way to rescale dataset.

Normalization

- Normalization rescales a dataset so that each value falls between 0 and 1. To accomplish the transformation, the following formula is used:

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

- x_i is the i^{th} value in the dataset.
- x_{min} is the minimum value of x.
- x_{max} is the maximum value of x.

- Question : Create a function in python that normalizes the above synthetic dataset

Normalization - sklearn

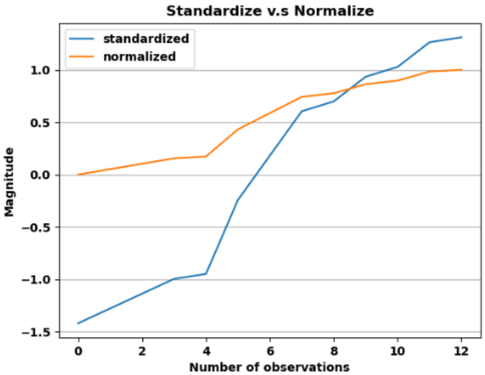
- From the sklearn package the .preprocessing.MinMaxScaler() function can be used to implement the normalization transformation.

```
import numpy as np
from sklearn.preprocessing import MinMaxScaler

x = np.array([13,16,19,22,23,38,47,56,58,63,65,70,71])

scalar = MinMaxScaler()

scalar.fit(x.reshape(-1,1))
scalar_transform = scalar.transform(x.reshape(-1,1))
print(f'Standardized data \n : {np.round(scalar_transform,2)}')
```

Standardization & Normalization

- Create a function in python that standardizes & normalizes the following synthetic dataset and plot the result in one graph.

Data	13	16	19	22	23	38	47	56	58	63	65	70	71
Standardized	-1.42	-1.28	-1.14	-0.99	-0.95	-0.24	0.18	0.60	0.70	0.93	1.03	1.26	1.31
Normalized	0	.05	0.1	0.16	0.17	0.43	0.59	0.74	0.78	0.86	0.9	0.98	1



Scaling to median and quantiles

- Scaling using median and quantiles consists of subtracting the median to all the observations and then dividing by the interquartile difference.

$$x_{scaled} = \frac{x - median(x)}{75thQuantile(x) - 25thQuantile(x)}$$

- It Scales features using statistics that are robust to outliers.

Interquartile

The interquartile difference is the difference between the 75th and 25th quantile:

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.
- Informally, the similarity between two objects is numerical measure of the degree to which two objects are alike.

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.
- Informally, the similarity between two objects is numerical measure of the degree to which two objects are alike.
- Similarities are usually non-negative number between 0 (no similarity) and 1 (complete similarity).

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.
- Informally, the similarity between two objects is numerical measure of the degree to which two objects are alike.
- Similarities are usually non-negative number between 0 (no similarity) and 1 (complete similarity).
- The dissimilarity(distance) between two objects is numerical measure of the degree to which two objects are different.

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.
- Informally, the similarity between two objects is numerical measure of the degree to which two objects are alike.
- Similarities are usually non-negative number between 0 (no similarity) and 1 (complete similarity).
- The dissimilarity(distance) between two objects is numerical measure of the degree to which two objects are different.

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.
- Informally, the similarity between two objects is numerical measure of the degree to which two objects are alike.
- Similarities are usually non-negative number between 0 (no similarity) and 1 (complete similarity).
- The dissimilarity(distance) between two objects is numerical measure of the degree to which two objects are different.
- Dissimilarities sometimes fall in the interval [0,1], but it is also common for them to range from 0 to ∞.

Measure of Similarity and Dissimilarity

- Similarity and dissimilarity are important because they are used by number of data mining techniques, such as clustering, nearest neighborhood and anomaly detection.
- Informally, the similarity between two objects is numerical measure of the degree to which two objects are alike.
- Similarities are usually non-negative number between 0 (no similarity) and 1 (complete similarity).
- The dissimilarity(distance) between two objects is numerical measure of the degree to which two objects are different.
- Dissimilarities sometimes fall in the interval [0,1], but it is also common for them to range from 0 to ∞.
- Proximity refers to similarity or dissimilarity.

Single attribute similarity/dissimilarity measures

Attribute Type	Dissimilarity	Similarity
Nominal	d $= \begin{cases} 0 & x = y \\ 1, & x \neq y \end{cases}$	s$=\begin{cases} 1 & x = y \\ 0, & x \neq y \end{cases}$
Ordinal	d$=\frac{\lVert x-y \rVert}{n-1}$ n number of values	$s = 1 - d$
Interval or Ratio	d$=\lVert x - y \rVert$	$s = \frac{1}{1+d}$ $s = 1 - \frac{d - min_d}{max_d - min_d}$

- **Nominal** is binary if two values are equal or not.

Single attribute similarity/dissimilarity measures

Attribute Type	Dissimilarity	Similarity
Nominal	d $= \begin{cases} 0 & x = y \\ 1, & x \neq y \end{cases}$	s$=\begin{cases} 1 & x = y \\ 0, & x \neq y \end{cases}$
Ordinal	d$=\frac{\lVert x-y \rVert}{n-1}$ n number of values	$s = 1 - d$
Interval or Ratio	d$=\lVert x - y \rVert$	$s = \frac{1}{1+d}$ $s = 1 - \frac{d-min_d}{max_d-min_d}$

- **Nominal** is binary if two values are equal or not.
- **Ordinal** is the difference between two values, normalized by maximum distance.

Single attribute similarity/dissimilarity measures

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & x = y \\ 1, & x \neq y \end{cases}$	$s = \begin{cases} 1 & x = y \\ 0, & x \neq y \end{cases}$
Ordinal	$d = \frac{\lVert x-y \rVert}{n-1}$ n number of values	$s = 1 - d$
Interval or Ratio	$d = \lVert x - y \rVert$	$s = \frac{1}{1+d}$ $s = 1 - \frac{d - min_d}{max_d - min_d}$

- **Nominal** is binary if two values are equal or not.
- **Ordinal** is the difference between two values, normalized by maximum distance.
- **Quantitative** dissimilarity is just a distance between, similarity attempts to scale the distance to [0,1]

Distance Properties

- Distance that satisfies the following well-known properties is called metric.

$$
\begin{aligned}
\textit{Positivity} &: d(x,y) \geq 0 \; \forall \; x,y \in \mathbb{R}^n \;\;, d(x,y) = 0 \textit{ iff } x = y \\
\textit{Symmetry} &: d(x,y) = d(y,x) \; \forall \; x,y \in \mathbb{R}^n \\
\textit{Traiangle Inequality} &: d(x,r) \leq d(x,y) + d(y,r) \; \forall \; x,y,r \in \mathbb{R}^n
\end{aligned}
$$

Euclidean Distance

Minkowski Distance

Hamming Distance

Mahalanobis Distance

Euclidean Distance

Euclidean Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Euclidean Distance is given by :

$$d(x,y) = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}$$

Euclidean Distance

Euclidean Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Euclidean Distance is given by :

$$d(x,y) = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}$$

- where n is the number of dimensions and x_k & y_k are the k^{th} components of x & y.

Euclidean Distance

Euclidean Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Euclidean Distance is given by :

$$d(x, y) = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}$$

- where n is the number of dimensions and x_k & y_k are the k^{th} components of x & y.
- If scales of the attributes differ substantially, standardization is necessary.

Minkowski Distance

- The Minkowski distance is a generalization of the Euclidean distance.

Minkowski Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Minkowski Distance is given by :

$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

Minkowski Distance

- The Minkowski distance is a generalization of the Euclidean distance.

Minkowski Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Minkowski Distance is given by :

$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

- where $r \geq 1$. It is also called the L_r metric.

Minkowski Distance

- The Minkowski distance is a generalization of the Euclidean distance.

Minkowski Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Minkowski Distance is given by :

$$d(x,y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

- where $r \geq 1$. It is also called the L_r metric.
- $r = 1$: L_1 metric, Manhattan or Hamming distance.

Minkowski Distance

- The Minkowski distance is a generalization of the Euclidean distance.

Minkowski Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Minkowski Distance is given by :

$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

- where $r \geq 1$. It is also called the L_r metric.
- $r = 1$: L_1 metric, Manhattan or Hamming distance.
- $r = 2$: L_2 metric, Euclidean distance.

Minkowski Distance

- The Minkowski distance is a generalization of the Euclidean distance.

Minkowski Distance

- Assume that we have measurements x & $y \in \mathbb{R}^n$. The Minkowski Distance is given by :

$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

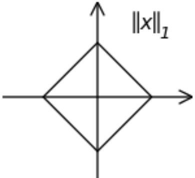
- where $r \geq 1$. It is also called the L_r metric.
- $r = 1$: L_1 metric, Manhattan or Hamming distance.
- $r = 2$: L_2 metric, Euclidean distance.
- $r \longrightarrow \infty$: L_∞ metric, Supremum distance.

$$\max(|x_1 - y_1|, ..., |x_n - y_n|)$$

Minkowski Distance

L_1 norm (Manhattan Distance)

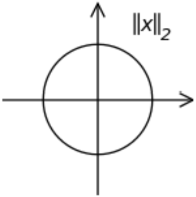
- The L_1 norm (commonly referred to as the taxicab or Manhattan distance) is formally defined as the sum of the absolute value of the difference in each coordinate between two vectors.
- Manhattan distance is the distance it would take you to walk along that grid from one point to another.
- A taxicab would drive through Manhattan from one point to another.



Minkowski Distance

L_2 norm (Euclidean Distance)

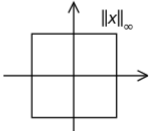
- Euclidean distance allows us to take straight-line paths from point to point, allowing us to reach further into the corners of the L_1 diamond.
- A circle $\rightarrow$ Euclidean distance.
- Since Euclidean distance is most common in the real world.



Minkowski Distance

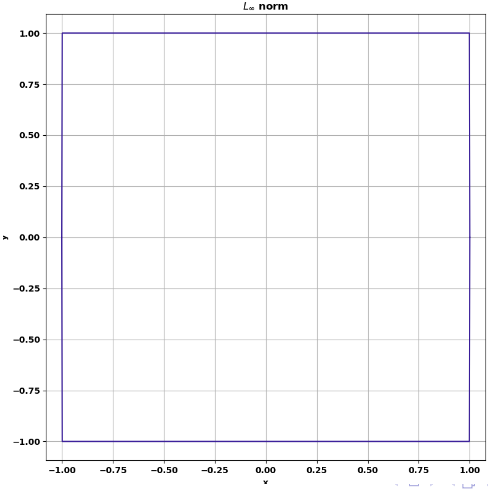
L_∞ norm (Chebyshev Distance)

- The L_∞ norm is equivalent to the maximum absolute dimension in the distance between two points.
- We wanted to construct a circle, where every point is equal Chebyshev's distance to the center, we would get a square
- Let's say that distance is 1. When compared to the origin, Chebyshev's distance will pick the highest absolute coordinate: x or y.
- So all points of distance 1 from the origin will have either x=±1, y=±1, or both, but never more



Minkowski Distance - r-norms

- r-norms with $r < 1$ squeeze in the corners, and travel further along the axis



In class Activity

- Let two vectors x & y to be defined as below.

point	x	y
p_1	0	2
p_2	2	0
p_3	3	1
p_4	5	1

- Calculate L_1, L_2 & L_∞ distances.

L_1	p_1	p_2	p_3	p_4
p_1				
p_2				
p_3				
p_4				

L_2	p_1	p_2	p_3	p_4
p_1				
p_2				
p_3				
p_4				

L_∞	p_1	p_2	p_3	p_4
p_1				
p_2				
p_3				
p_4				

Mahalanobis Distance

- Let $X \in \mathbb{R}^{n \times p}$. The i^{th} row of X is:

$$x_i^T = (x_{i1}, \ldots, x_{ip})$$

Mahalanobis Distance

Mahalanobis Distance

- Let $X \in \mathbb{R}^{n \times p}$. The i^{th} row of X is:

$$x_i^T = (x_{i1}, ..., x_{ip})$$

Mahalanobis Distance

- The Mahalanobis distance is

$$d(i,j) = ((x_i - x_j)^T \Sigma^{-1} (x_i - x_j))$$

Mahalanobis Distance

- Let $X \in \mathbb{R}^{n \times p}$. The i^{th} row of X is:

$$x_i^T = (x_{i1}, ..., x_{ip})$$

Mahalanobis Distance

- The Mahalanobis distance is

$$d(i,j) = ((x_i - x_j)^T \Sigma^{-1} (x_i - x_j)))$$

- where Σ is the $p \times p$ sample covariance matrix.

Variance & Covariance

- When dealing with problems on statistics and machine learning one of the most frequently encountered terms is covariance.
- Most of us know that variance represents the variation on a single variable but we may not sure what is covariance.

Covariance

- Covariance can provide way more information on solving multivariate problems.
- Most of the methods for preprocessing or predictive analysis depend on the covariance.
- Multivariate outlier detection, dimensionality reduction, and regression can be given as examples.

Variance

- It would be better to go over the variance to understand the covariance.
- The variance explains how the values vary in a variable.
- It depends on how the values far from each other.

Population is known

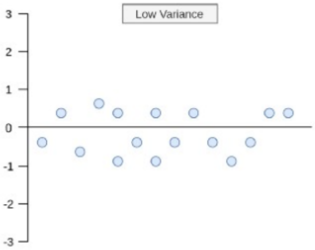
$$var(x) = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}$$

Population is unknown

$$var(x) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}$$

Different variances

- Difference between high and low variance.



Covariance

Covariance

- Unlike the variance, covariance is calculated between <u>two</u> different variables.
- Its purpose is to find the value that indicates how these two variables <u>vary</u> together.
- In the covariance formula, the values of both variables are multiplied by taking the difference from the mean.

Population is known

$$cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{n}$$

Population is unknown

$$cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n - 1}$$

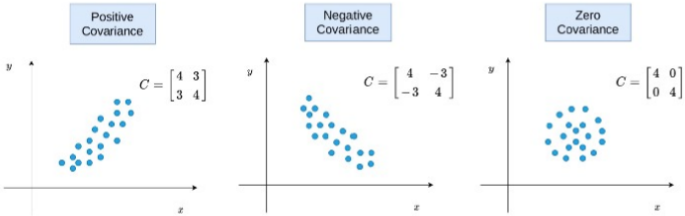
Covariance Matrix

- Because covariance can only be calculated between two variables, covariance matrices stand for representing covariance values of each pair of variables in multivariate data.
- The covariance between the same variables equals variance, so, the diagonal shows the variance of each variable.
- The values in the covariance matrix shows the distribution magnitude and direction of multivariate data in multidimensional space.

$$
\begin{array}{cc}
 & \begin{array}{cc} \mathrm{x} & \mathrm{y} \end{array} \\
\begin{array}{c} \mathrm{x} \\ \mathrm{y} \end{array} &
\begin{bmatrix} var(x) & cov(x,y) \\ cov(x,y) & var(y) \end{bmatrix}
\end{array}
\qquad
\begin{array}{cc}
 & \begin{array}{ccc} \mathrm{x} & \mathrm{y} & \mathrm{z} \end{array} \\
\begin{array}{c} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{array} &
\begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(x,y) & var(y) & cov(y,z) \\ cov(x,z) & cov(y,z) & var(z) \end{bmatrix}
\end{array}
$$

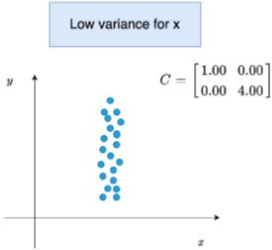
Covariance Matrix

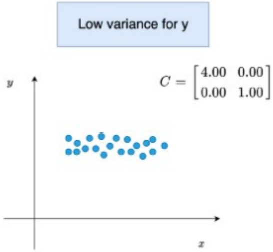
- If the sum of these values is positive, covariance gets found as positive.
- It means variable X and variable Y variate in the same direction.
- If there is a negative covariance, this is interpreted right as theopposite.
- The covariance can only be zero when the sum of products of $x_i - \overline{x}$ and $y_i - \overline{y}$ is zero.



Covariance Matrix

- when the covariance is near zero and the variance of variables are different.





Eigenvalues and Eigenvectors of Covariance Matrix

- Eigenvalues and Eigenvectors are the essential part of the covariance matrix.
- Eigenvalue eigenvector finds the magnitude and direction of the data points.
- Eigenvalues $\rightarrow$ magnitude of the spread in the direction of the principal components in PCA.
- When the covariance is zero,eigenvalues will be directly equal to the variance values

Mathematics of Eigenvalues and Eigenvectors

- For a square matrix A, an eigenvalue λ and eigenvector v make this equation true.

$$Av = \lambda v$$

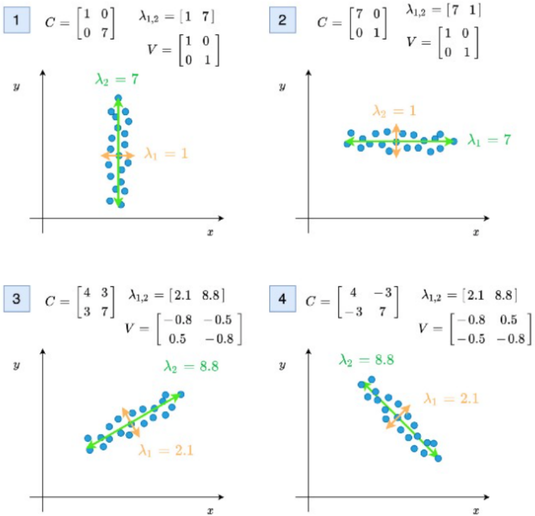
How to find eigen things?

$$\begin{aligned} Av &= \lambda v \\ Av &= \lambda I v \\ Av - \lambda I v &= 0 \end{aligned}$$

- If v is non-zero (hopefully) solve for λ using the determinant

$$|A - \lambda I| = 0$$

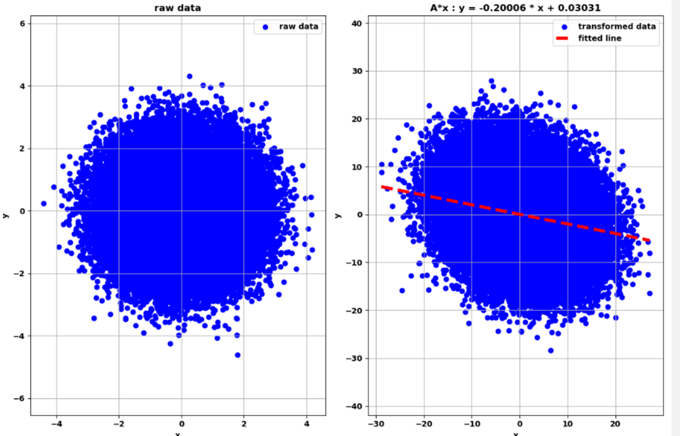
Eigenvalues and Eigenvectors of Covariance Matrix



λ = eigenvalues $\qquad V$ = eigenvectors

In class Activity

- Calculate the eigenvalue and eigenvector associated with the matrix A= $\begin{pmatrix} -6 & 3 \\ 4 & 5 \end{pmatrix}$
- Answer : $\lambda_1 = 6$ and $\lambda_2 = -7$



Similarities between Data Objects

- For similarities, the triangle property topically does not hold but the symmetry & positivity typically do.

$$s(x,y) = 1 \; \textit{only if} \; x = y. \; (0 \leq s \leq 1)$$
$$s(x,y) = s(y,x) \; \forall \; x, y \in \mathbb{R}^n$$

Non-symmetric similarity

If $s(x,y) \neq s(y,x)$, then similarity measure can be made symmetric by setting :

$$s'(x,y) = s'(y,x) = \frac{s(x,y) + s(y,x)}{2}$$

Non-symmetric similarity Example

Example : A Non-symmetric similarity

Consider an experiment to classify characters as they flash on a screen. The confusion matrix for this experiment records: $s('0','o') = 40$ and $s('o','0') = 30$.

		Predicted	
		'0'	'o'
Actual	'0'	160	40
	'o'	30	170

Similarity Measures for Binary Data

- Similarity measures between objects that only contains binary attributes are called similarity coefficients and typically have values between 0 and 1.

Similarity Measures for Binary Data

- Similarity measures between objects that only contains binary attributes are called similarity coefficients and typically have values between 0 and 1.
- 1$\rightarrow$ completely similar

Similarity Measures for Binary Data

- Similarity measures between objects that only contains binary attributes are called similarity coefficients and typically have values between 0 and 1.
- $1 \rightarrow$ completely similar
- $0 \rightarrow$ not at all similar.

Similarity Measures for Binary Data

- Similarity measures between objects that only contains binary attributes are called similarity coefficients and typically have values between 0 and 1.
- 1$\rightarrow$ completely similar
- 0 $\rightarrow$ not at all similar.
- Let x & y be two objects that consists of p binary attributes. The comparison of two objects leads to the following quantities that defines **Simple Matching Coefficient(SMC)**
  1. f_{00} = the # of attributes where x is 0 and y is 0.
  2. f_{01} = the # of attributes where x is 0 and y is 1.
  3. f_{10} = the # of attributes where x is 1 and y is 0.
  4. f_{11} = the # of attributes where x is 1 and y is 1.

$$\boxed{SMC = \frac{\#of\ \ matching\ \ attributues}{total\#of\ \ attributues} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}}$$

Similarity Measures for Binary Data

- **SMC Example** : SMC could be used to find students who answered questions similarly on a test that consists of T/F questions.

Jaccard Coefficient

Similarity Measures for Binary Data

- **SMC Example** : SMC could be used to find students who answered questions similarly on a test that consists of T/F questions.

Jaccard Coefficient

- A Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes.

$$\boxed{J = \frac{\#of\ \ matching\ \ presences}{\#of\ \ attributues\ \ no\ \ involved\ \ 00\ \ matches} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}}$$

SMC versus Jaccard coefficent-Example

- Let x & y are data objects that represent two rows of a transaction matrix. Let 1 $\rightarrow$ purchased items and 0 $\rightarrow$ Not purchased. The number of not purchased products outnumbers the number of purchased products.

SMC versus Jaccard coefficent-Example

- Let x & y are data objects that represent two rows of a transaction matrix. Let $1 \rightarrow$ purchased items and $0 \rightarrow$ Not purchased. The number of not purchased products outnumbers the number of purchased products.
- Per the SMC, all transactions are very similar.

SMC versus Jaccard coefficent-Example

- Let x & y are data objects that represent two rows of a transaction matrix. Let 1 $\rightarrow$ purchased items and 0 $\rightarrow$ Not purchased. The number of not purchased products outnumbers the number of purchased products.
- Per the SMC, all transactions are very similar.
- Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes.

SMC versus Jaccard coefficent-Example

- Let x & y are data objects that represent two rows of a transaction matrix. Let 1 $\rightarrow$ purchased items and 0 $\rightarrow$ Not purchased. The number of not purchased products outnumbers the number of purchased products.
- Per the SMC, all transactions are very similar.
- Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes.
- Calculate the SMC and Jaccard Similarity coefficient for the following two binary vectors:

$$x = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$y = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$$

Cosine Similarity

- Documents $\Longrightarrow$ vectors where each component (attributes) represents the frequency with which a particular term (word) occurs in the document.

Cosine Similarity

- Documents $\Longrightarrow$ vectors where each component (attributes) represents the frequency with which a particular term (word) occurs in the document.
- Each documents have thousands or tens of thousands of attributes (terms).

Cosine Similarity

- Documents $\Longrightarrow$ vectors where each component (attributes) represents the frequency with which a particular term (word) occurs in the document.
- Each documents have thousands or tens of thousands of attributes (terms).
- Each document is sparse since it has relatively few non-zero attributes.

Cosine Similarity

- Documents $\Longrightarrow$ vectors where each component (attributes) represents the frequency with which a particular term (word) occurs in the document.
- Each documents have thousands or tens of thousands of attributes (terms).
- Each document is sparse since it has relatively few non-zero attributes.
- Thus similarity should not depend on the 0 values, because any two documents are likely to not contain many of same words.

Cosine Similarity

- Documents $\Longrightarrow$ vectors where each component (attributes) represents the frequency with which a particular term (word) occurs in the document.
- Each documents have thousands or tens of thousands of attributes (terms).
- Each document is sparse since it has relatively few non-zero attributes.
- Thus similarity should not depend on the 0 values, because any two documents are likely to not contain many of same words.
- If 0-0 matches are counted, most documents will be highly similar to most others.

Cosine Similarity

- Documents $\Longrightarrow$ vectors where each component (attributes) represents the frequency with which a particular term (word) occurs in the document.
- Each documents have thousands or tens of thousands of attributes (terms).
- Each document is sparse since it has relatively few non-zero attributes.
- Thus similarity should not depend on the 0 values, because any two documents are likely to not contain many of same words.
- If 0-0 matches are counted, most documents will be highly similar to most others.
- Similarity measure for documents needs to ignores 0-0 matches (like Jaccard) but also handle non-binary vectors. $\Longrightarrow$ $\boxed{\text{cosine similarity}}$

Cosine Similarity

Cosine Similarity

- If x and y are two documents vectors,

$$\cos(x, y) = \frac{\langle x, y\rangle}{\|x\|\|y\|} = \frac{x^T y}{\|x\|\|y\|}$$

where $\langle x, y\rangle$ is the inner product of vectors

$$\langle x, y\rangle = \sum_{k=1}^{n} x_k y_k = x^T y$$

And $\|x\|$ is the length of vector x

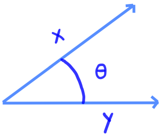
$$\|x\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{\langle x, x\rangle} = \sqrt{x^T x}$$

Cosine Similarity

- Cosine similarity also can be written as:

$$\cos(x, y) = \langle x', y' \rangle$$

where $x' = \frac{x}{\|x\|}$ and $y' = \frac{y}{\|y\|}$.

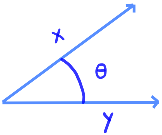


Cosine Similarity

- Cosine similarity also can be written as:

$$\cos(x, y) = \langle x', y' \rangle$$

  where $x' = \frac{x}{\|x\|}$ and $y' = \frac{y}{\|y\|}$.
- Dividing x and y by length normalizes them to have a length of 1.



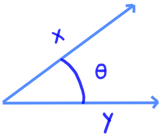
Cosine Similarity

- Cosine similarity also can be written as:

$$\cos(x, y) = \langle x', y' \rangle$$

where $x' = \frac{x}{\|x\|}$ and $y' = \frac{y}{\|y\|}$.

- Dividing x and y by length normalizes them to have a length of 1.
- Cosine similarity does not take the **length** into account for the similarity computation.



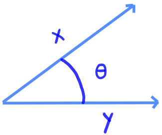
Cosine Similarity

- Cosine similarity also can be written as:

$$\cos(x, y) = \langle x', y' \rangle$$

where $x' = \frac{x}{\|x\|}$ and $y' = \frac{y}{\|y\|}$.

- Dividing x and y by length normalizes them to have a length of 1.
- Cosine similarity does not take the **length** into account for the similarity computation.
- The inner product of two vectors works well for asymmetric attributes since it depends only on components that are non-zero in both vectors.



Extended Jaccard Coefficient

Tanimoto Coefficient

The extended Jaccard coefficient (Tanimoto Coefficient) can be used for document data and that reduces in the case of binary attributes.

$$EJ(x, y) = \frac{\langle x, y \rangle}{\|x\|^2 + \|y\|^2 - \langle x, y \rangle} = \frac{x^T y}{\|x\|^2 + \|y\|^2 - x^T y}$$

- Example: Calculates the cosine similarity for the two data objects:

$$x = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$
$$y = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

Correlation

- Correlation is frequently used to measure the linear relationship between two sets of values. And is used to measure similarity between attributes.

Correlation

- Correlation is frequently used to measure the linear relationship between two sets of values. And is used to measure similarity between attributes.
- **Pearson's correlation** between two sets of numerical values x and y:

$$\boxed{corr(x,y) = \frac{covariance(x,y)}{std(x).std(y)} = \frac{s_{xy}}{s_x s_y}}$$

$$covariance(x,y) = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \overline{x})(y_k - \overline{y})$$

$$std(x) = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k - \overline{x})^2}, std(y) = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k - \overline{y})^2}$$

Correlation

$$\overline{x} = \frac{1}{n}\sum_{k=1}^{n} x_k, \quad \overline{y} = \frac{1}{n}\sum_{k=1}^{n} y_k$$

- Correlation is always in the range of $[-1, 1]$.

Correlation

$$\overline{x} = \frac{1}{n}\sum_{k=1}^{n} x_k, \quad \overline{y} = \frac{1}{n}\sum_{k=1}^{n} y_k$$

- Correlation is always in the range of $[-1, 1]$.
- A correlation of 1 (-1) means x and y have a perfect positive(negative) relationship.

Correlation

$$\overline{x} = \frac{1}{n}\sum_{k=1}^{n} x_k, \quad \overline{y} = \frac{1}{n}\sum_{k=1}^{n} y_k$$

- Correlation is always in the range of $[-1, 1]$.
- A correlation of 1 (-1) means x and y have a perfect positive(negative) relationship.
- Nonlinear relationship: If the correlation is 0, then there is no linear relationship between the two sets. However, nonlinear relationship can still exist.

Correlation

$$\bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k, \quad \bar{y} = \frac{1}{n}\sum_{k=1}^{n} y_k$$

- Correlation is always in the range of $[-1, 1]$.
- A correlation of 1 (-1) means x and y have a perfect positive(negative) relationship.
- Nonlinear relationship: If the correlation is 0, then there is no linear relationship between the two sets. However, nonlinear relationship can still exist.
- Example:

$$x = (-3, -2, -1, 0, 1, 2, 3)$$
$$y = (9, 4, 1, 0, 1, 4, 9)$$

$y_k = x_k^2$ the correlation coefficient is zero but nonlinear relationship exists.

Procedural Programming

- There ar primarily two methods of programming in use today:
  1. Procedural
  2. **Object Oriented Programming (OOP)**

Procedural

- Procedural program is made of one or more procedures.
- Think of procedural programming as a function that performs specific task such as gathering input from user, performing calculations, reading and writing files, display output and so on.
- In a procedural program, the data item are commonly passed from one procedure to another.
- Making changes (updating functions to meet new criteria) in procedural programming is difficult as the program becomes larger and more complex.

Object Orient Programming (OOP)

- Whereas procedural programming is centred on creating procedures (functions), Object-Orient Programming (OOP) is centred on creating objects.
- An Object is a software entity that contains both data and procedures.
- Everything in Python is an object such as integers, lists, dictionaries, functions and so on. Every object has a type and the object types are created using classes.
- Instance is an object that belongs to a class. For instance, list is a class in Python. When we create a list, we have an instance of the list class.

Object Orient Programming (OOP)

Attribute

- A *variable* stored in an instance or class is called attribute.
- A value associated with an object which is referenced by name using dotted expressions. For example, if an object VT has an attribute c it would be referenced as VT.c

Method

- A *function* stored in an instance or class is called Method.
- A function which is defined inside a class body. If called as an attribute of an instance of that class, the method will get the instance object as its first argument (which is usually called self).

Procedural versus Object Oriented Programming

```
#======================
# Procedural Programming
#======================
a = 10                          # variable
def f(b):                       # function
    return b**2
print(f(12))
#============================
# Object Oriented Programming
#============================
class C:

    c = 20                     # class attribute

    def __init__(self, number): # 'initializer' method
        self.num = number      # instance attribute

    def show(self):            # method
        print(self.num**2)

e = C(12)
e.show()
e.g = 40                       # another instance variable
```