Python for Data Science
Introduction to (practical) Variable Analysis and Feature Engineering

Heru Praptono

heru.pra@cs.ui.ac.id

Salemba, December 2018

Outline

Insight

- What we want to obtain?

Variable Analysis

- Correlation
- F test

Feature Engineering

- Feature Selection
- Feature Transformation

Insight: What we want to obtain?

- Getting **insight** on **variables analysis and feature engineering**
- Getting in touch with Python more, on math computing.
  - A very central idea: **“primitive” vs matrix based** computation

A very important property of this universe...

- Universe contains randomness. From randomness, there is uncertainty. We need to know for at least a phenomena on what's going on between variables and their interacton.
- Fortunately, at least there is more and more data emission as the consequences of their existence.
  - Many of us have limited information on it.
- We then first "see" the data, and think what sort of operations/tasks to be implemented to the data, so that we gather information from data.
  - In general, the task is about to get the knowledge representations, that are constructed by learning from data.
- The central concepts/tasks can be:
  - UNSupervised: e.g. clustering
  - supervised: e.g. classification, numerical predicton (or generally said as "general" regression)

In order to make any inference, we need to be able make statistical assumption.
So we start from some variable representation and their analysis.

Introd to Correlation

The concept of correlation gives central idea on how we model relationship between two variables. More specificaly, it is an approximate measure of the size and direction of a relationship between two variables.

Introd to Correlation

Suppose that we have two random variables, x and y, where $x, y \in \mathbb{R}$. First, we consider their **sample covariance** $\mathrm{cov}(x, y)$, that is

$$\mathrm{cov}(x, y) \triangleq \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(y - \bar{y}) \tag{1}$$

which is the expected of product of their deviations, and it is an estimator of their covariance $\mathrm{Cov}(x, y)$, defined as

$$\mathrm{Cov}(x, y) \triangleq \mathbb{E}[(x - \mathbb{E}(x))(y - \mathbb{E}(y))] \tag{2}$$

that $\mathbb{E}(x) \approx \frac{1}{N} \sum_{i=1}^{N} (x_i)$ is an unbiased estimator, where x_i sampled as $x_i \overset{iid}{\sim} P_x(x_i)$ (this is also for $\mathbb{E}(y)$, respectively)

Introd to Correlation

For any multivariate x having D attributes, the corresponding sample covariance matrix is thus

$$\operatorname{cov}(x_1, \ldots, x_D) = \begin{pmatrix} \operatorname{cov}(x_1, x_1) & \operatorname{cov}(x_1, x_2) & \ldots & \operatorname{cov}(x_1, x_D) \\ \operatorname{cov}(x_2, x_1) & \operatorname{cov}(x_2, x_2) & \ldots & \operatorname{cov}(x_2, x_D) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}(x_D, x_1) & \operatorname{cov}(x_D, x_2) & \ldots & \operatorname{cov}(x_D, x_D) \end{pmatrix} \tag{3}$$

Introd to Correlation

The magnitude of $\mathrm{cov}(x, y)$ in this case is relatively not easy to understand, as that is unnormalised. Thus, we then need to find correlation value, that is:

$$\rho(x, y) = \frac{\mathrm{cov}(x, y)}{\sigma(x)\sigma(y)} \tag{4}$$

where $\sigma(x)$ is standard deviation for x and $\sigma(y)$ is standard deviation for y respectively. Thus the $\rho(x, y)$ value falls in the range $[-1, 1]$.

Introd to Correlation: A very important thing to remember!

Correlation score DOES NOT guarantee to imply any causation relationship!
This means that when two variables have high correlation (e.g. x_1 is highly correlated with x_2), it does not mean that the change value in x_1 causes change value in x_2.

Data and Variables

Any datapoint $\mathbf{x}^{(i)}$ in our dataset can be represented as vector, D dimensional vector, $\mathbf{x} \in \mathbb{R}^D$. Having two possible different tasks, we can represents our data as:

- in supervised setting: $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$ where $y \in \mathbb{R}$ for regression, and $y \in \mathbb{Z}$ for classification
- in unsupervised setting: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$

Data and Variables

- $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)} \ldots, x_D^{(i)})^T$ (our common datapoint representation)
- Any $\mathbf{x}^{(i)}$ is i.i.d (independent and identically distributed) from any probability distribution $P_{\mathbf{x}}$.

$$\mathbf{x}^{(i)} \overset{iid}{\sim} P_{\mathbf{x}} \tag{5}$$

(in general)

or

$$\mathbf{x}^{(i)}, y^i \overset{iid}{\sim} P_{\mathbf{x},y} \tag{6}$$

(in supervised learning setup)

PROBLEM with this world: We do **NOT** really know what exactly the true P is

Data and Variables

- so let us start from the most simple univariate predictor x and the relationship with response y.
- we need to analyse the relationship between x and y.
- an example and as the central idea for it, is correlation analysis, as we discussed before, where x and y are generally real number, $x, y \in \mathbb{R}$.
- when x is real number, $x \in \mathbb{R}$, and y is discrete/categorical $y \in \mathbb{Z}$, we may use F test anova.
- ...of course, those are some examples, and indeed there are many other ways of measurement like information gain, etc.

F test

Given dataset as the following:

x_1	x_2	y
0.5	0.11	A
1.5	0.13	A
2.4	0.15	B
3.6	0.17	B
4.3	0.19	C
5.6	0.21	C
...	...	...

- we can clearly see that $x_1, x_2 \in \mathbb{R}$ and $y \in \mathbb{Z}$
- thus, we can use F test anova.

F test

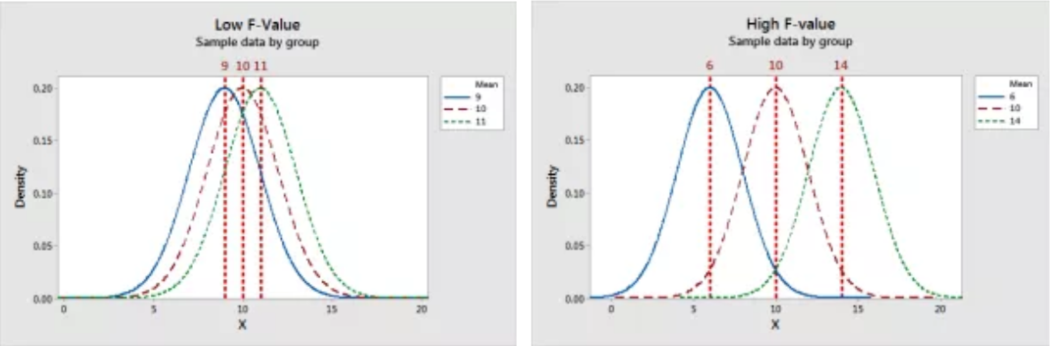
In order to calculate f value for F test:

$$\mathrm{F} = \frac{\mathrm{betweenGroupVariability}}{\mathrm{withinGroupVariability}} \tag{7}$$

so if we want to calculate f value on x_1, then

$$\mathrm{F}_{\mathrm{x}_1} = \frac{\mathrm{var}_{A,B,C}(x_1)}{\mathrm{var}_A(x_1) + \mathrm{var}_B(x_1) + \mathrm{var}_C(x_1)} \tag{8}$$

F test



from left to right: low f value, high f value.

Feature Engineering: Motivation

- Computational issue
- Model's performance issue

The curse of dimensionality

- As the dimensionality increases, the volume of the space increases so fast that the available data become sparse.
- The amount of data needed grows exponentially with the dimensionality

$$N = e^D \tag{9}$$

Feature Selection

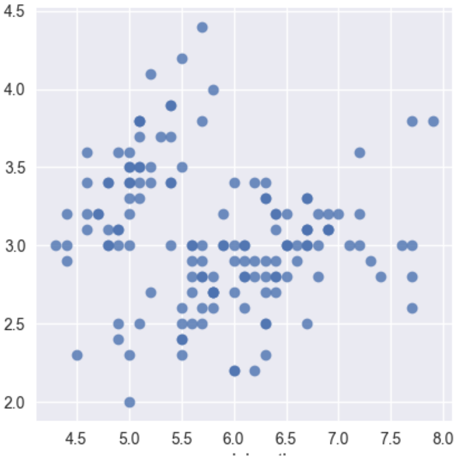
- The idea of feature selection, is creating the subset of features, that is, K features from the original D features, so that $K \leq D$.
- Conditioned on the relationship measure, the priority of chosing belongs to the strongest measure value between x (predictor/independent variable) and y (response/dependent variable)
- for example, the higher correlation between x and y should be the prioritised earlier.
- if any of x is indeed constant, then it gives no information, and in correlation case for example, its correlation value to any other variable gives 0.

but wait...

- What if there is multicolinearity?
- ...dependency between features?

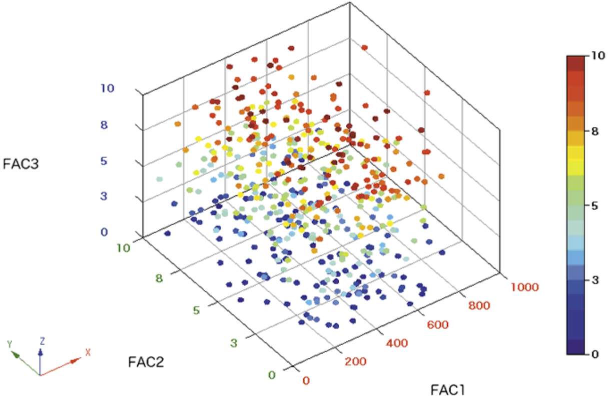
solution: orthogonal feature transformation.

and also, consider about this..



2D: that's good!

and also, consider about this..



3D: that's still OK!

but if..

Now, let us say, we have D=1000 features, how do we visualise the dataset?

Solution: reduce the dimensionality, utilise transformation

Feature Transformation

- key: create new feature space, that still represents the datapoint on its original position.
- example: Principal Component Analysis (PCA), autoencoder (neural network based).

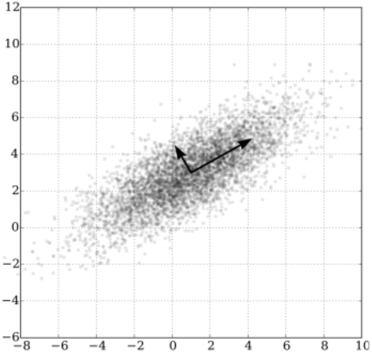
Feature Transformation: PCA

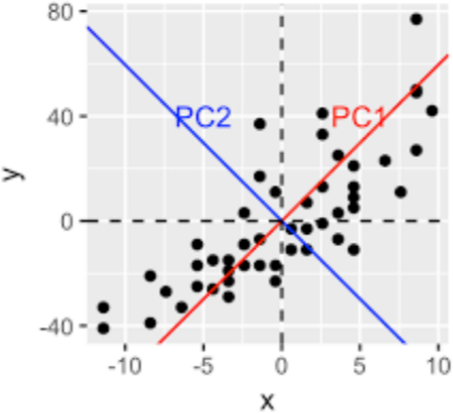
Intuition:

- Reduces dimensionality, by utilising linear orthogonal transformation, instead of feature selection.
- Transforms original feature into new feature representation, BUT still represents one data to others.
- Ensures independent attributes
- Find the axes (proportional to "principles") that maximise variance
- First principle component correspondences to greatest eigenvalue first, followed by further principle component

Feature Transformation: PCA

Intuition:





PCA: how it works

given covariance matrix A (size D x D), the goal is to find its corresponding eigen value λ and eigenvector $\mathbf{v}$, that fullfill:

$$A\mathbf{v} = \lambda\mathbf{v} \tag{10}$$

Solution:

$$(A - \lambda\mathbf{I})\mathbf{v} = 0 \tag{11}$$

for non zero $\mathbf{v}$, this requires determinant of $A - \lambda\mathbf{I}$, that is $|A - \lambda\mathbf{I}| = 0$ so that the solution for $\mathbf{v}$ exists.

$$|A - \lambda\mathbf{I}| = (\tau_1 - \lambda)(\tau_2 - \lambda)\ldots(\tau_D - \lambda) \tag{12}$$

here $\tau \in \mathbb{R}$ or $\tau \in \mathbb{C}$

PCA: how it works, new feature

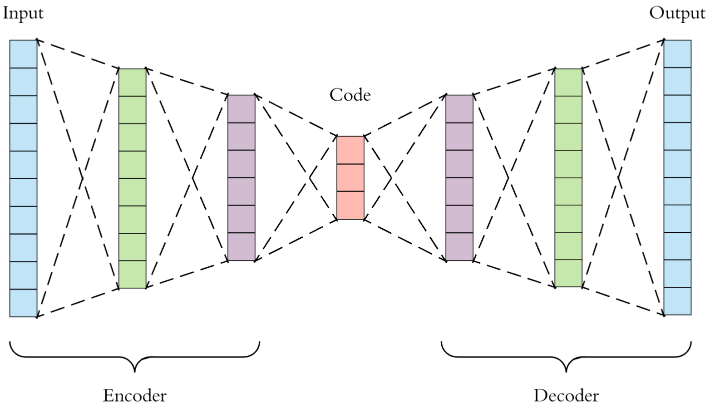
- Sort **v** based on its corresponding λ. The highest λ value represents first principal. **Question:** How many possible λ's, if solution exists?
- new first feature f/first principal component:

$$f_1 = A\mathbf{v}_1 \tag{13}$$

when λ_1 is the highest.

Other: Autoencoder

Idea: Use neural network architecture, deep learning, to find its latent representation, given observed data.



Thank You