Notes of

Fintech

From Prof. Marazzina & Prof. Fabrizio's lectures
for the MSc in Mathematical Engineering

by Teo Bucci

Politecnico di Milano
A.Y. 2021/2022

© The authors, some rights reserved.

This work is lincensed under CC BY-NC-SA 4.0.
http://creativecommons.org/licenses/by-nc-sa/4.0/

In summary: you can share the contents of the book, in whole or in part, and make your changes, provided you cite the source, share the changes under the same license, and do not use the material for commercial purposes (it is not allowed to print the book for resale).

The LaTeX source code is available at
https://github.com/teobucci/fintech

Document created on June 29, 2022
Revision 003d05a4e5228b9617251c38adb591202458e13a

Developed by:
Teo Bucci - teobucci8@gmail.com

To report any errors or suggestions you can contact the authors or make a Pull Request.

Preface

This book collects the notes of the *Fintech* course for students of Mathematical Engineering, held at the Politecnico di Milano in the Academic Year 2021-2022.

Contents

Chapter 1

Machine Learning (D. Marazzina)

- Machine learning is a branch of AI.
- The idea underlying machine learning is that we give a computer program access to lots of data and let it learn about relationships between variables and make predictions.
- Some of the techniques of machine learning date back to the 1950s but improvements in computer speeds and data storage costs have now made machine learning a practical tool.

Different kinds of *learning:*

- **Unsupervised** learning (find patterns)
  - Examples: interpolation (with noise data), calibration, lapse prediction
  - Methods: Neural Networks, Clustering
- **Supervised** learning (predict numerical value or classification)
  - Examples: credit scoring, client selection
  - Methods: Neural Networks, Decision Trees, Random Forests
- **Reinforcement learning** (multi-stage decision making)
  - Examples: trading
  - Methods: Q-Learning and Neural Network/Random Forest (to approximate matrix Q)

Software:

- There a several alternatives such as Python, R, MATLAB, Spark, and Julia.
- Need ability to handle very large data sets and availability of packages that implement the algorithms.
- Python seems to be winning at the moment.

Traditional statistics Means, SDs, Probability distributions, Significance tests, Confidence intervals, Linear regression.

Modern statistics

- Huge data sets
- Fantastic improvements in computer processing speeds and data storage costs
- Machine learning tools are now feasible.
- Can now develop non-linear prediction models, find patterns in data in ways that were not possible before, and develop multi-stage decision strategies
- New terminology: features, labels, activation functions, target, bias, supervised/unsupervised learning.

Features are the *old* independent variables for linear regression, but in ML it is no more important that they are independent (no problem of collinearity).

ML good practice:

- Divide data into three sets
  - Training set
  - Validation set
  - Test set
- Develop different models using the training set and compare them using the validation set
- Rule of thumb: increase model complexity until model no longer generalizes well to the validation set
- The test set is used to provide a final out-of-sample indication of how well the chosen model works

1.1 Data cleaning

Data cleaning refers to identifying and correcting errors in the dataset that may negatively impact a predictive model. Data cleaning is used to refer to all kinds of tasks and activities to detect and repair errors in the data. Although critically important, data cleaning is not exciting, not does it involve fancy techniques. Just a good knowledge of the dataset. Cleaning up your data is not the most glamorous of tasks, but it's an essential part of data wrangling. Knowing how to properly clean and assemble your data will set you miles apart from others in your field. *Garbage in, garbage out!*

There are many types of errors that exist in a dataset, although some of the simplest errors include columns that don't contain much information and duplicated rows.

- Identify (and delete) columns that contain a single value
- Consider columns that have very few values (near 0 variance)
- Identify (and delete) rows that contain duplicate data

It is not always true that more features you gave, better it is. If you think that a feature is not important for your model, try to remove it.

If some observations contain missing or inconsistent data, consider the possibility

- to remove the observation
- to remove the feature (if this affect only a feature and a large number of observations, e.g., only 60% of observations has the data of a given feature known)
- to "fill the gap" inserting, e.g., a mean/median value (if numerical) or interpolating (if they are time-dependent data, and you know the values at time t_{i-1} and t_{i+1}, but not at time t_i).

Isolation forest is an algorithm for anomaly detection that works on the principle of isolating anomalies. In statistics, an **anomaly** (or **outlier**) is an observation or event that deviates so much from other events to arouse suspicion it was generated erroneously.

Anomalies in a big dataset may follow very complicated patterns, which are difficult to detect "by eye" in the great majority of cases. Isolation Forest explicitly isolates anomalous points in the dataset. From a mathematical point of view, recursive partitioning can be represented by a tree structure named Isolation Tree, while the number of partitions required to isolate a point can be interpreted as the length of the path, within the tree, to reach a terminating node starting from the root. Intuitively, the anomalous points are those (easier to isolate, hence) with the smaller path length in the tree.

1.2 Unsupervised Learning

- In unsupervised learning we are not trying to predict anything.
- The objective is to **cluster** data to increase our **understanding** of the environment.
- How? With the **k-means algorithm**.

Before using many ML algorithms (including those for unsupervised learning), it is important to scale feature values so that they are comparable.

- **Z-score** scaling involves calculating the mean m and SD from the values of each feature from the

training set. Scaled feature values for all data sets are then created by subtracting the mean and dividing by the SD. The scaled feature values have a $m = 0$ and $SD = 1$

$$\frac{V - m}{SD}$$

- An alternative is the **Min-max** scaling: it involves calculating the maximum and minimum value of each feature from the training set. Scaled feature values for all data sets are then created by subtracting the minimum and dividing by the difference between the maximum and minimum. The scaled feature values lie between zero and one.

$$\frac{V - \min}{\max - \min}$$

For clustering we need a **distance** measure. The simplest distance measure is the Euclidean distance measure.

$$d(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

In general when there are m features the distance between P and Q is

$$d(P, Q) = \sqrt{\sum_{j=1}^{m} (v_{pj} - v_{qj})^2}$$

where v_{pj}, v_{qj} are the values of the j-th feature for P and Q.

Now that we have a distance measure, it is important to define the **center of a cluster**, also known as **cluster centroid**.

k-means algorithm to find k clusters:

1. Choose k random points as cluster centers
2. Assign each observation no the nearest center
3. Calculate new cluster centers
4. If cluster centers have changed, go to (2) otherwise end.

Inertia:

For any given k the objective is to minimize **inertia**, which is defined as the within cluster sum of squares:

$$\text{inertia} = \sum_{i=1}^{n} {d_i}^2$$

where d_i is the distance of observation i from its cluster center, and n is the number of observations. In practice we use the k-means algorithm with several different starting points and choose the result that has the smallest inertia.

Generally, **the inertia decreases as k increases**, approaching 0 when $k = n$, i.e., we have n observations and n clusters, i.e., each cluster coincide with a single observation.

Therefore we need to find a proper way to set k.

- **The elbow approach:** in this case, $k = 4$ seems the best choice, since the inertia decreases of a small quantity moving from 4 to 5 w.r.t from 3 to 4.
- **The elbow approach + distance between clusters analysis:** if moving from k to $k + 1$ clusters we create a cluster with center very close to the center of another cluster, we stop the algorithm to k.
- **The silhouette method:**[1] for each observation i calculate $a(i)$, the average distance from other observations in its cluster, and $b(i)$, the average distance from observations in the **closest other cluster**, i.e., for each cluster (excluding the one which contains the observation), we compute the average distance of the considered observation i from the observations of the cluster, and **then take the minimum**. The silhouette score for observation i, $s(i)$, is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

[1] `https://en.wikipedia.org/wiki/Silhouette_(clustering)`

Choose the number of clusters that maximizes the average silhouette score across all observations. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

- The **Gap statistic** compares the within cluster sum of squares with what would be expected with random data.
  - Let us create N sets of random points.
  - For each set, let us cluster this points in k clusters, and compute the Inertia, for different k.
  - Let us define with
    * w_k the inertia computed for the real data
    * $m_k(s_k)$ the mean value (standard deviation) of the N inertias computed exploiting the random sets
  - We define the Gap as
$$\texttt{gap}(k) = m_k - w_k$$
  - We set k such that
$$\texttt{gap}(k) \in (s_{k+1}, \texttt{gap}(k+1))$$

1.2.1 The Curse of Dimensionality

- The Euclidean distance measure increases as the number of features increase.
- This is referred to as the **curse of dimensionality**, and could prevent the k-means algorithm to work efficiently.
- Consider two observations that have values for feature j equal to x_j and y_j. An alternative distance measure that always lies between 0 and 2 is
$$1 - \frac{\sum_{j=1}^m x_j y_j}{\sqrt{\sum_{j=1}^m x_j^2 \sum_{j=1}^m y_j^2}}$$

Now we see some alternative clustering approaches.

1.2.2 Hierarchical Clustering

- Start with each observation in its own cluster
- Combine the two closest clusters
- Continue until all observations have been combined into a single cluster (or since we have only k clusters)
- Can be implemented in Python with `sklearn.cluster.AgglomerativeClustering`.
- Measures of closeness of clusters:
  - Average Euclidean distance between points in clusters
  - Maximum distance between points in clusters
  - Minimum distance between points in clusters
  - Increase in inertia (a version of Ward's method)

1.2.3 Density-based clustering

- Forms clusters based on the closeness of individual observations.
- Unlike k-means the algorithm, it is not based on cluster centers.

- We might initially choose 8 observations that are close. After that we add an observation to the cluster if it is close to at least 5 other observations in the cluster, and repeat.

1.2.4 Distribution-based Clustering

Assumes that observations come from a mixture of distributions and uses statistical procedures to separate the distributions.

1.3 Reduce dimensionality

1.3.1 Principal Components Analysis (PCA)

This is an approach to reduce the number of variables

- PCA replaces a set of n variables by n **new** factors so that:
  - Any observation on the original variables is a linear combination of the n factors
  - The n factors are uncorrelated
  - The quantity of a particular factor in a particular observation is the factor score
  - The importance of a particular factor is measured by the standard deviation of its factor score across observations
- The idea is to find a few variables that **account for a high percentage of the variance** in the observations.

The algorithm goes as follow:

1. Standardize data with the Z-score scaling.
2. Compute the variance-covariance matrix.
3. Compute its eigenvalues and eigenvectors.
4. Eigenvectors are the Principal Components. *Higher is the eigenvalues, more important is the corresponding principal components.*
5. The factor scores are the eigenvalues, and the SD of the factor scores are the square root of the eigenvalues itself.

Last step: **recast the data along the principal components axes**.

In the previous steps, apart from standardization, we did not make any changes on the data, we just selected the principal components and formed the feature vector, but the input data set remains always in terms of the original axes (i.e, in terms of the initial variables).

We now need to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$\texttt{FinalDataSet} = \texttt{FeatureVector}^{\mathrm{T}} \cdot \texttt{StandardizedOriginalDataSet}^{\mathrm{T}}$$

1.4 Supervised Learning: with focus on Classification

1.4.1 Linear Regression

Linear regression is a very popular tool because once you have made the assumption that the model is linear you do not need huge amount of data.

In ML we refer to the constant term as the *bias* and the coefficients as *weights*. Assume m features and n observations:

$$Y = a + b_1 X_1 + \cdots + b_m X_m + \varepsilon,$$

we choose a, b_i to minimize the mean squared error:

$$\text{MSE} = \frac{1}{n}\sum_{j=1}^{n}[Y_j - (a + b_1 X_{1,j} + \cdots + b_m X_{m,j})]^2$$

This can be done analytically by inverting a matrix. Alternatively a numerical (**gradient descent**) method can be used. The objective is to minimize a function by changing parameters. Steps are as follows:

1. Choose starting value for parameters.
2. Find the steepest slope: i.e. the direction in which parameter have to be changed to reduce the objective function by the greatest amount.
3. Take a step down the valley in the direction of the steepest slope.
4. Repeat steps (2) and (3).
5. Continue until you reach the bottom of the valley.

The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect). **A low p-value (less than 0.05) indicates that you can reject the null hypothesis**. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable.

R^2 varies between 0 and 1: when 0, the model used does not explain the data at all; when it is 1 the model perfectly explains (i.e *describes*, bad!) the data.

Categorical features are features where there are a number of *non-numerical* alternatives. We can define a *dummy variable* for each alternative. The variable equals 1 if the alternative is true and zero otherwise. This is known as one-hot encoding. But sometimes we do not have to do this because there is a natural ordering of variables, e.g.:

- small 1, medium 2, large 3.
- assist. prof 1, assoc. prof 2, full prof 3.

Dummy Variably Trap. Suppose we have a constant term and a number of dummy variables (equal to 0 or 1) where, for each observation, only one dummy can take value 1 (e.g., dummies $X_1, X_2, \ldots$: dummy *born in Europe, born in Africa*, ...).

There is then no unique solution because, for any C, we can add C to the constant term and subtract C from each of the dummy variables coefficient **without changing the prediction.**

$$Y = a + b_1 X_1 + b_2 X_2 = a + C + (b_1 - C)X_1 + (b_2 - C)X_2$$

Regularization solves this problem.

- Linear regression can over-fit, particularly when there are a large number of correlated features.
- Results for validation set may not then be as good as for training set
- Regularization is a way of avoiding over fitting and reducing the number of features. Possible Regularization technique:
  - Ridge
  - Lasso
  - Elastic net
- Must first scale feature values

1.4.2 Ridge regression

- Reduce magnitude of regression coefficients by choosing a parameter λ and minimizing:

$$\frac{1}{n}\sum_{j=1}^{n}[Y_j - (a + b_1 X_{1,j} + \cdots + b_m X_{m,j})]^2 + \lambda \sum_{i=1}^{m} b_i^2$$

- What happens as λ increases? Minimizing the above function, all b_is approach 0 as λ increases.
- Ridge regression, as linear regression, can be solved analytically.

1.4.3 Lasso Regression

- Similar to ridge regression except we minimize:

$$\frac{1}{n}\sum_{j=1}^{n}\left[Y_j-(a+b_1X_{1,j}+\cdots+b_mX_{m,j})\right]^2+\lambda\sum_{i=1}^{m}|b_i|$$

- This has the effect of completely eliminating the less important factors .
- Must use gradient descent.

1.4.4 Elastic Net Regression

- Middle ground between *Ridge* and *Lasso:*
- Minimize:

$$\frac{1}{n}\sum_{j=1}^{n}[Y_j-(a+b_1X_{1,j}+\cdots+b_mX_{m,j})]^2+\lambda_1\sum_{i=1}^{m}b_i^2+\lambda_2\sum_{i=1}^{m}|b_i|$$

- Must use gradient descent.

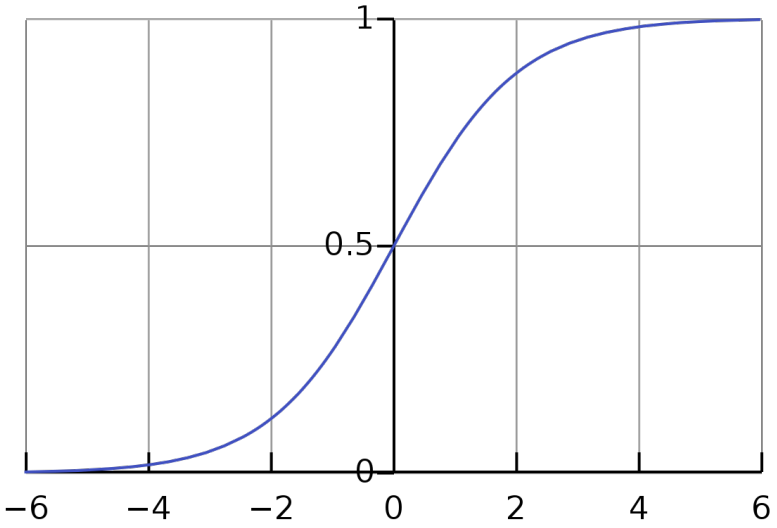
1.5 Logistic Regression

- The objective is to classify observations into a "positive outcome" and "negative outcome" using data on features
- Dependent variable: binary (1 – 0)
- Probability of a positive outcome is assumed to be a **sigmoid** function:

$$Q=\frac{1}{1+e^{-Y}}$$

where Y is related linearly to the values of the features:

$$Y=a+b_1X_1+\cdots+b_mX_m$$



$Y\in\{0,1\}$. Let $Q=p(X)$ be the probability that $Y=1$, conditioned to the value of X.

$$\log\left(\frac{p(X)}{1-p(X)}\right)=\beta_0+\beta_1X$$

Multivariate

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

Maximum Likelihood Estimation (MLE)

- We use the training set to maximize

$$\sum_{\text{pos. outcomes}} \log(Q) + \sum_{\text{neg. outcomes}} \log(1-Q)$$

- This cannot be maximized analytically but we can use a gradient ascent algorithm.

Logistic regression as classifier.

Once betas are estimated, we use the regression to estimate $p\,(X)$.

We have to choose a **threshold!**

$$\begin{aligned} p > \overline{Z} &\implies \widehat{Y} = 1 \\ p < \overline{Z} &\implies \widehat{Y} = 0 \end{aligned}$$

- **Sensitivity.** True Positive Rate, the proportion of class 1 (positive) members classified as such.
- **Specificity.** True Negative Rate, the proportion between the members of class 0 (negative) classified as such.

The **confusion matrix** and common ratios[2]

	Predict pos.	Predict neg.
Outcome pos.	TP	FN
Outcome neg.	FP	TN

$$\text{accuracy} = \frac{\texttt{TP} + \texttt{TN}}{\text{all}}$$

$$\text{True Positive Rate (TPR or sensitivity)} = \frac{\texttt{TP}}{\texttt{TP} + \texttt{FN}}$$

$$\text{True Negative Rate (TNR or specificity)} = \frac{\texttt{TN}}{\texttt{TN} + \texttt{FP}}$$

$$\text{False Positive Rate} = \frac{\texttt{FP}}{\texttt{TP} + \texttt{FP}} = 1 - \texttt{TNR}$$

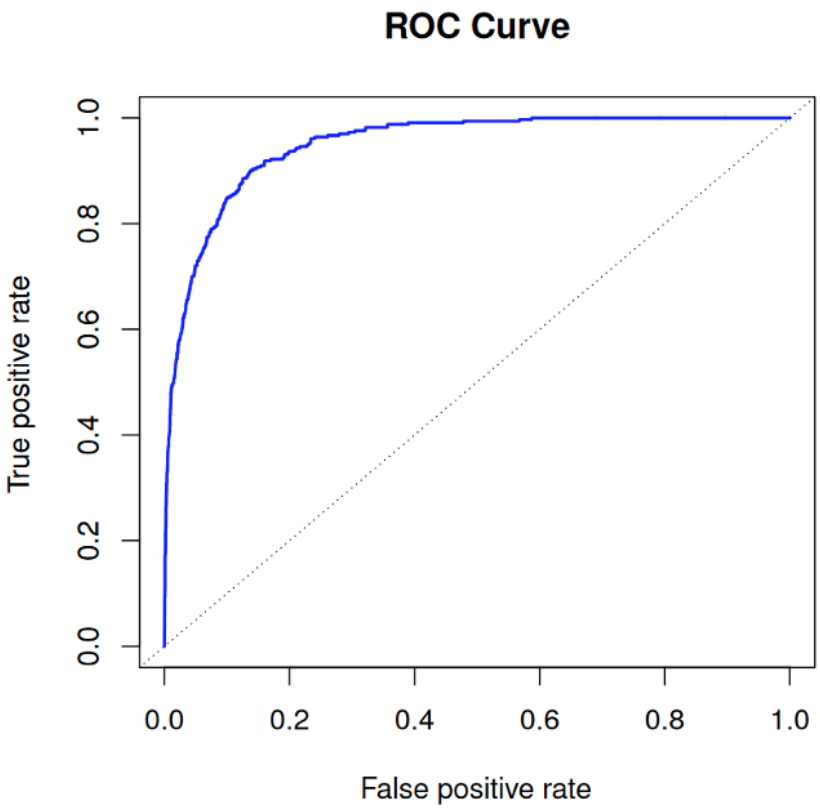
$$\text{Precision (P)} = \frac{\texttt{TP}}{\texttt{TP} + \texttt{FP}}$$

$$F\text{-score} = 2 \cdot \frac{\texttt{P} \cdot \texttt{TPR}}{\texttt{P} + \texttt{TPR}}$$

ROC Curve (receiver operating characteristic).

Sensitivity and specificity are tested by varying the classifier parameters - e.g. the logistic classifier barrier.

[2] https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers



The performance of a classifier can be measured using the so-called `AUC`**, the area under the ROC curve**. The *greater the area the greater the goodness* (sensitivity and specificity) that the classifier is able to obtain. The bisector in the graph is obtained through a trivial classifier (I randomly assign the observations to the two classes). For a classifier to be useful, the `AUC` **must beat the trivial classifier** (i.e. `AUC` greater than 0.5).

The area under the curve is a popular way of summarizing the predictive ability of a model to estimate a binary variable

- When `AUC` $= 1$ the model is perfect.
- When `AUC` $= 0.5$ the model has no predictive ability.
- When `AUC` < 0.5 the model is worse than random.

Imbalanced Dataset [3] typically refers to a problem with classification problems where the classes are not represented equally.

The *accuracy paradox* is the name for the exact situation in the introduction to this post.

It is the case where your accuracy measures tell the story that you have excellent accuracy (such as 90%), but the accuracy is only reflecting the underlying class distribution.

If we have 99% of observations belonging to class 1, the trivial predictor giving as output «class 1» for any input will have a 99% accuracy.

- **Can You Collect More Data?**
- **Try Changing Your Performance Metric**
  Accuracy is not the metric to use when working with an imbalanced dataset. We have seen that it is misleading.
- **Try Resampling Your Dataset**
  You can change the dataset that you use to build your predictive model to have more balanced data.

[3] `https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/`

This change is called sampling your dataset and there are two main methods that you can use to even-up the classes:
You can add copies of instances from the under-represented class called over-sampling (or more formally sampling with replacement), or
You can (randomly) delete instances from the over-represented class, called under-sampling.

- **Try Generate Synthetic Samples**
  A simple way to generate synthetic samples is to randomly sample the attributes from instances in the minority class.
- **Try Penalized Models**
  You can use the same algorithms but give them a different perspective on the problem.
  Penalized classification imposes an additional cost on the model for making classification mistakes on the minority class during training. These penalties can bias the model to pay more attention to the minority class.

1.5.1 Alternative to regression: clusterization

Use Unsupervised Learning to construct clusters.

- Each cluster corresponds to a set of probabilities, e.g., the probability of being a class 1 cluster is equal to the number of observations in this cluster belonging to class 1 divided by the total number of observations in this cluster.
- If a new observation fall in a cluster, it inherits all the cluster's class probabilities.

1.5.2 Alternative to regression: k-nearest neighbors classifier

- Normalize data.
- Measure the distance in n-dimensional space of the new data from the data for which there are labels (i.e. known outcomes).
- Distance of point with feature values x_i from point with feature values y_i is the euclidean distance.
- Choose the k closest data items and average their labels.
- For example if you are forecasting car sales in a certain area with $k = 3$ and the three nearest neighbors for GDP growth and interest rates give sales of 5.2, 5.4 and 5.6 million units, the forecast would be the average of these or 5.4 million units.
- If you are forecasting whether a loan will default with $k = 5$ and that of the five nearest neighbors four defaulted and one was good loan, you would estimate an 80% chance of default

1.5.3 Networks for clustering

Graph (directed/undirected/partially directed/weighted/bipartite).

Adjacency matrix

$$A_{ij} = \begin{cases} 1 & i,j \text{ connected} \\ 0 & \text{otherwise} \end{cases}$$

Adjacency list: stores only the non zero elements of the Adjacency matrix in a list of all vertices.

Centrality measures: *which vertices are more important?*

- **degree centrality,**
- **closeness centrality,**
- **betweenness centrality,**
- **eigenvector centrality.**

Degree Centrality

- The simplest measure of importance is the degree centrality
- The vertex with largest degree exerts the greatest effect on the network

- Degree centrality: number of nearest neighbours

$$\boxed{D(i) = \sum_{j=1}^{n} A_{ij} = \sum_{j=1}^{n} A_{ji}}$$

- Sensitive to the addition of one more node
- We cannot compare degree centrality of vertices belonging to *two different* networks
- Normalized degree centrality (relative metric to compare degree centrality)

$$\boxed{N_D(i) = \frac{1}{n-1} D(i)}$$

In directed graphs:

- **In-Degree:** the number of edges *incoming* to a vertex

$$d_i^{\leftarrow} = \sum_{j=1}^{n} A_{ji}$$

- **Out-Degree:** the number of edges *outgoing* from a vertex

$$d_i^{\rightarrow} = \sum_{j=1}^{n} A_{ij}$$

Shortest Path

A **path** in a network is a sequence of vertices $x, y, \ldots, z$ such that each consecutive pair of vertices i, j is connected by an edge (i, j) in the network.

The **shortest path** is the shortest of all possible paths between two vertices.

Average Path Length

The average graph-distance between all pair of nodes

$$L_G = \frac{1}{n(n-1)} \sum_{i \neq j} d(v_i, v_j)$$

If all nodes are connected, then the graph distance equals to 1

Closeness Centrality

- Measures how close a node is to all the other nodes in network (in terms of the shortest path)

$$\boxed{C(i) = \frac{1}{\sum_{j=1}^{n} d(i,j)}}$$

  where $d(i, j)$ is the *shortest* path between the nodes i and j.

- Normalized closeness centrality (relative metric to compare closeness centrality)

$$\boxed{N_C(i) = (n-1)C(i)}$$

Betweenness Centrality

- The number of shortest paths from all vertices to all others that pass through a node.

$$\boxed{B(i) = \sum_{s \neq i \neq t} \frac{\delta_{st}(i)}{\delta_{st}}}$$

  where $\delta_{st}(i)$ is the number of shortest paths between s and t that pass through i, and δ_{st} is the total number of shortest paths between s and t.

- Normalized betweenness centrality

$$N_B(i) = \frac{2}{(n-1)(n-2)} B(i)$$

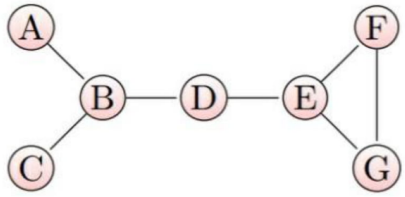
- Probability that a communication from s to t will go through i.

Eigenvector Centrality

- Eigenvector centrality measures assign an importance score to vertices in a way that is proportional to the importance scores of its neighbors hence the importance of a node depends on the importance of its neighbors.
- This involves an eigenvector problem of the form

$$Av = \lambda_1 v$$

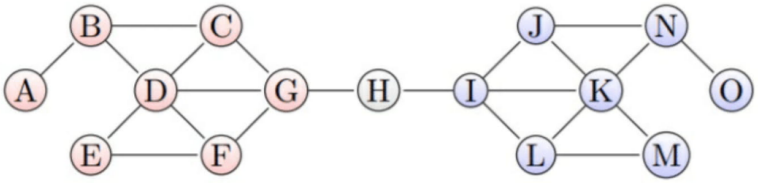
  where A is the adjacency matrix, v is a vector containing the eigenvector centralities, and λ_1 is the *largest eigenvalue* of A.
- Other forms of eigenvector centrality
  - Bonacichs centrality
  - PageRank centrality
  - Katz centrality and hub/authority scores



▶ Comparison of different centralities

Node	Degree	Closeness	Betweeness	Eigenvector
A	0.1667	0.3750	0	0.2134
B	**0.5000**	0.5454	**0.6000**	0.4814
C	0.1667	0.3750	0	0.2136
D	0.3333	**0.6000**	**0.6000**	0.6574
E	**0.5000**	0.5454	0.5333	**1.0000**
F	0.3333	0.4000	0	0.7979
G	0.3333	0.4000	0	0.7979

Consider



Comparison of different centralities:

- Degree most Central Nodes: D and K.
- Closeness most Central Nodes: H.
- Betweenness most Central Nodes: H.
- Eigenvector most Central Nodes: D and K.

1.6 Decision Trees

1.6.1 Measures of Uncertainty

- Suppose that there are n possible outcomes and p_i is the probability of outcome i with $\sum_i p_i = 1$.
- **Entropy** measure of a node:

$$\texttt{entropy} = \mathbb{E}\left[\texttt{information}\right] = \sum_{i=1}^{n} p_i \cdot \log\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \log p_i$$

- **Gini** measure of a node:

$$\texttt{gini} = 1 - \sum_{i=1}^{n} p_i^2$$

- In case of a binary variable ($n = 2$) we have

$$\texttt{gini} = 1 - (p^2 + (1-p)^2) = 2p(1-p)$$

1.6.2 Information Gain

- The information gain is the expected decrease in uncertainty (as measured by either entropy or Gini).
- Suppose that there is a 20% chance that a person will receive a job offer
- Suppose further that there is a 50% chance the person has a relevant degree. If the person does have a relevant degree the probability of a job offer rises to 30%, otherwise it falls to 10%
- Initial entropy:

$$-[0.2\log(0.2) + 0.8\log(0.8)] = 0.7219$$

- Expected entropy:

$$-0.5\left[0.1\log\left(0.1\right) + 0.9\log\left(0.9\right)\right] - 0.5\left[0.3\log\left(0.3\right) + 0.7\log\left(0.7\right)\right] = 0.6751$$

- Expected information gain from knowing whether there is a relevant degree:

$$0.7219 - 0.6751 = 0.0468$$

- In general, suppose that node x is split into x_L, x_R and our split is called $\mathcal{J}$

$$\texttt{informationGain}(\mathcal{J}) = \underbrace{\texttt{entropy}(x)}_{\substack{\text{entropy before} \\ \text{split}}} \underbrace{-p_L\texttt{entropy}(x_L) - p_R\texttt{entropy}(x_R)}_{\text{entropy after split}}$$

  where p_L and p_R are the proportions of observations that from node x fall into nodes x_L and x_R.

1.6.3 The Decision Tree Algorithm

- Algorithm chooses the feature at the root of the tree that has the greatest expected information gain.
- At subsequent nodes it chooses the feature (not already chosen) that has the greatest expected information gain.
- When there is a threshold (numerical feature), it determines the optimal threshold for each feature (i.e., the threshold that maximizes the expected information gain for that feature) and bases calculations on that threshold.

Classification Tree:[4] the output variable is discrete and finite (usually binary).

Regression Tree:[5] the output variable has continuous values.

[4] `https://youtu.be/_L39rN6gz7Y`
[5] `https://youtu.be/g9c66TUylZ4`

1.6.4 Classification Tree

- Classification trees are recursive algorithms that split the dataset into smaller sets (nodes) in a step by step fashion thanks to binary rules defined on the features of the observations.
- The complete dataset is the root node of the tree. To split the node, a feature is selected and a binary rule, e.g., if the value of the feature is larger or smaller than a given threshold, is defined on it in order to obtain two disjoint datasets. These two datasets become nodes of the tree. This procedure is repeated for each new node until a stopping criterion is met, e.g., the specified maximum depth of the tree is reached.
- So a classification tree is a growing tree with nodes refining the information about the exogenous variables to classify an item in the database (in the binary case, class 0 or 1).
- Each node is associated with a measure of *Impurity*. Such a measure is high when the dataset representing the node contains observations belonging to different classes of the endogenous variable, and reaches its lowest value (zero) when the node only contains observations belonging to a single class.
- In the binary case, i.e. only two classes, we can use the Gini impurity. Given a node x, the Gini impurity, denoted by $G(x)$, for a binary classier is defined as

  $$G(x) = 2p(1-p)$$

  where p denotes the proportion of observations belonging to class 1, contained in node x. p can be interpreted as the probability of being of class 1 in the node. We can also use Entropy, instead of Gini index.
- We aim to **minimize** the Gini index; its maximum (1/4) corresponds to the case in which a node is perfectly balanced (50% of the observations in the node are of class 0 and 50% of class 1, and therefore $p = 1/2$).
- The feature (exogenous variable) and the binary rule that are used for the split at a node are defined through a search process that maximizes the impurity decrease after the split. Using the terminology introduced before, the information gain here is:

  $$\texttt{informationGain}(\mathcal{J}) = G(x) - p_L \cdot G(x_L) - p_R \cdot G(x_R)$$

  The last nodes of the tree (the arrival point of the algorithm, i.e., the nodes that are not split anymore) are called *leaves*.
- The leaves contain the estimation of the tree (in our case class 0 or 1), which is given by the most recurrent class in that leaf: if more than 50% of the observations in the leaf are of class 1, then the leaf is a class 1 leaf.
- We also get a probability of being of class 1 for each leaf, given by the proportion of observations of class 1 in that leaf.

1.6.5 Regression Tree

- Regression trees have basically the same structure as classification trees, but the dependent variable is a *numerical* variable, and the measure of impurity is the MSE of the observations in the node.
- The quantity predicted by each leaf is the average of the values of the target variable of the observations in the leaf.
- The MSE measures the pureness of the node, computing the distance between the estimated value of the node (the average of the value of all the observations belonging to the node) and the value of the observations contained in the node, i.e., if there are N observations belonging to a node, and p_i is the value of observation i, $i = 1, \dots, N$, then we compute

  $$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - \frac{1}{N}\sum_{j=1}^{N} p_j\right)^2$$

  The information gain here is:

  $$\texttt{informationGain}(\mathcal{J}) = \text{MSE}(x) - p_L \cdot \text{MSE}(x_L) - p_R \cdot \text{MSE}(x_R)$$

- We construct a tree where instead of maximizing expected information gain we maximize the **expected decrease in mean squared error.**

1.6.6 Random Forest

- This involves constructing many trees by for example:
  - Using samples bootstrapped from the original data
  - Using a random subset of features at each node
  - Randomizing thresholds in some way
- The final decision can be a majority vote or a weighted majority vote. Weights can reflect probability estimates (when available) or evidence from a hold-out test data set.

Bagging

- Sample with replacement to create new data sets.
- Use voting or averaging methods for final estimate.

Boosting

- Predictions are made sequentially, each trying to correct the previous error.
- One approach (AdaBoost) increases the weight given to misclassified observations.
- Another approach (Gradient boosting) tries to fit a new predictor to the error made by the previous predictor.

From Bagging...

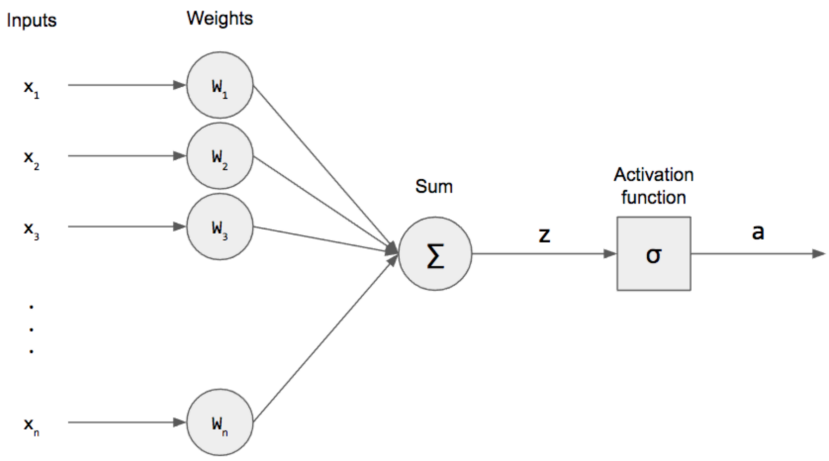
- Bagging (Bootstrap Aggregation) is used to reduce the variance of classification/decision trees.
- The methodology builds several datasets from the training set choosing randomly observations with re-entry from the original dataset (statistical bootstrap).
- Each dataset is used to train a tree. As a result, we obtain an ensemble of trees-models and the final classifier is obtained averaging the predictions from the different trees yielding a more robust outcome than the one obtained from a single tree.

...to Random Forests

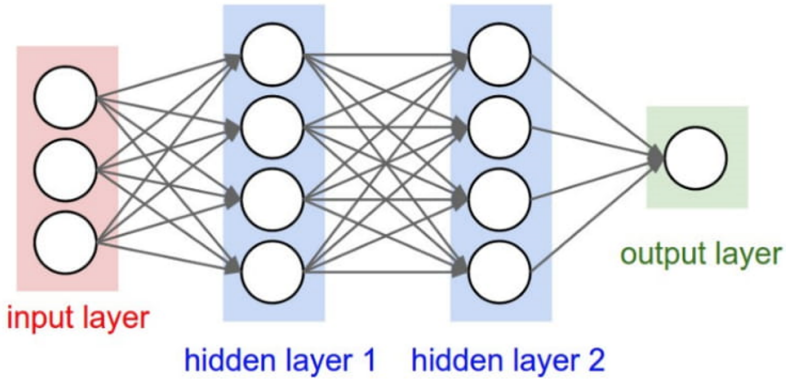
- A RF classifier is an extension of the bagging methodology. The approach considers random subsets of observations from the original dataset and it also randomly selects the features rather than using all the features to construct each tree.
- The output of the RF classifier is computed averaging the predictions of all the trees.
- To calibrate the parameters of the RF we have to set a priori some hyper-parameters. The parameters of the model are then calibrated on the training set.
- The hyper-parameters are chosen evaluating the performance of the calibrated models on the validation set.
  - **maximum tree depth**: the maximum depth for a tree;
  - **minimum leaf size**: the minimum number of observations contained in a leaf. A split will only be considered if the number of observations belonging to each child node will be higher than the threshold;
  - **minimum decrease of impurity after a split**: a node is split if it generates an impurity decrease greater than or equal to the threshold;
  - **minimum split size**: the minimum number of observations belonging to a node required to split it;
  - **number of trees**: the number of trees in the forest.

1.7 Supervised Learning: Neural Networks

Perceptron

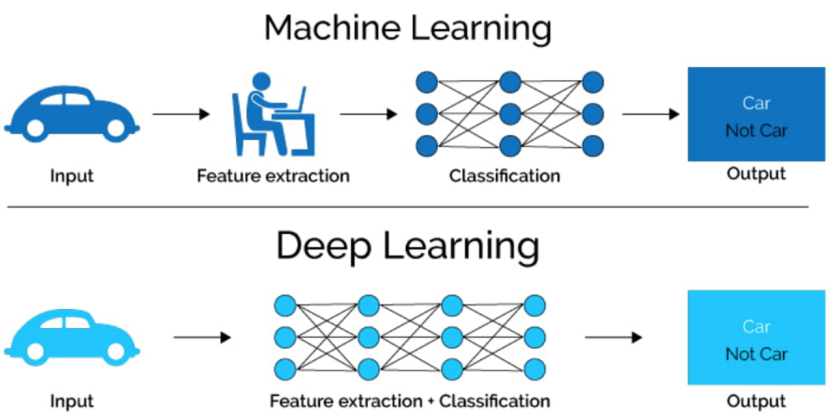


A 3-layer Neural Network (NN) with 2 layers of hidden units



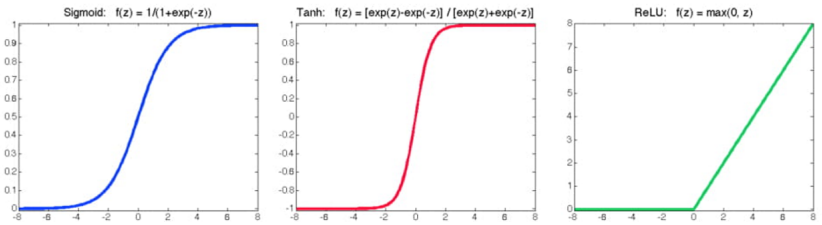
Naming convention: a N**-layer** neural network with $N-1$ layers of **hidden units** and one **output layer**. Usually we don't count the input layer.

Usually every layer has its activation function.



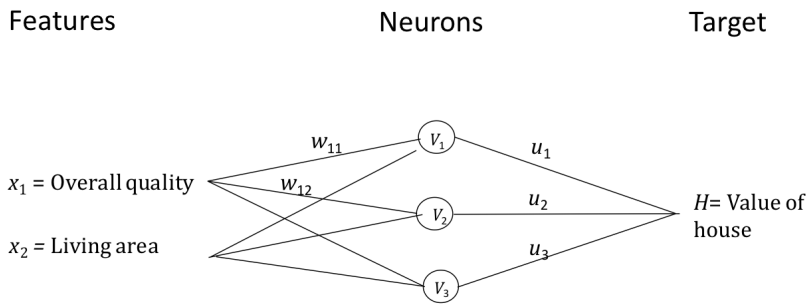
The most common activation functions are

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \qquad \mathrm{ReLU}(z) = \max(0,z)$$



Remember that ReLU (Rectified Linear Unit) is unbounded.

Example.



Let us use the Sigmoid to relate the v_k to the features.

$$v_1 = \frac{1}{1+e^{-a_1-w_{11}x_1-w_{21}x_2}} \quad v_2 = \frac{1}{1+e^{-a_2-w_{12}x_1-w_{22}x_2}} \quad v_3 = \frac{1}{1+e^{-a_3-w_{13}x_1-w_{23}x_2}}$$

The output is a bias (constant) plus linear combination of the preceding values

$$H = c + u_1v_1 + u_2v_2 + u_3v_3$$

We have 13 degrees of freedom.

Universal Approximation Theorem. Any continuous function can be approximated to arbitrary accuracy with one hidden layer[6], but this may require a very large number of neurons. Using several hidden layers can be computationally more efficient.

However, the theorem doesn't tell how many neurons or layers.

If there are F features, H hidden layers, M neurons in each hidden layer and T targets the number of parameters is:[7]

$$\#\text{parameters} = (F+1)M + M(M+1)(H-1) + (M+1)T$$

How to deal with such a large number of parameters?

To minimize an objective function such as MSE, a **gradient descent** algorithm calculates the direction of steepest descent, takes a step, calculate a new direction of steepest descent, takes another step, and so on. The partial derivatives with respect to the parameters are calculated by a procedure known as **backpropagation**.

The size of the step is the **learning rate:** if the step is too small the algorithm will be very slow. If it is too large there are liable to be oscillations.

Stopping rule

It is important to use a stopping rule to **avoid over-fitting**. We calculate results for both the validation set and the training set: **when the results for the validation set start to get worse we stop**.

[6] See: K. Hornik, Neural Networks, 1991, 4:251-257.
[7] Idea: (what comes from previous + 1) · what comes next.

1.7.1 Convolutional Neural Network (CNN)

In a vanilla ANN the layers are **fully connected** which can give rise to a very large number of parameters. In a CNN, the number of neurons in one layer that affect the next layer is **reduced**. Often used for image recognition where each pixel can be a feature.

Key advantages. Consider a image that is $100 \cdot 100$ or 10.000 pixels. A regular neural network would lead to $10.000 \cdot (10.000 + 1)$ or about 100 million parameters to define the first layer. If a CNN's receptive field is $10 \cdot 10$ and a layer has 6 feature maps only $6 \cdot 101 = 606$ are required. More importantly, once the CNN has learned to recognize a pattern in one location it can recognize it in another location.

1.7.2 Recurrent Neural Network (RNN)

In a vanilla ANN, the v's are functions of the inputs and the sequence of the inputs does not matter. In a recurrent neural network there is a **time dimension** in the data. The current v's are made functions of the corresponding previous v's (or possibly previous outputs) as well as current inputs. A **long short-term memory (LSTM)** network is a type of RNN where part of the algorithm learns what should be remembered and what should be forgotten from previous data.

Applications are when working with sequential data where the relationship between targets and features may be changing through time. Also used in NLP (autocompletion for example).

1.8 Interpretability

Why is model interpretability important?

- Users must understand a model to have confidence in it, know when it is appropriate, be aware of its biases, etc
- It is also important to be able to explain the predictions made by the model, e.g.,
  - Why was someone refused for a loan?
  - Why is house A worth more than house B
- The General Data Protection Regulation in the European Union requires model interpretability

White-box vs black-box models

White-box models

- k-nearest neighbors
- Decision trees
- Linear regression

Black-box models

- Neural networks
- Ensemble models (e.g. random forests)

Interpretability issue: explain black-box algorithm decisions!

Features dependency could be a problem: we might be able to group features that should be considered together. Sometimes a PCA is used to create uncorrelated features.

Black-box models

- Models must be re-run to determine *the impact of the change in a feature value on a prediction.*
- In general there is non-linearity so that when changes are made to the feature values the sum of the contributions of the features does not equal the change in the prediction

Local Interpretability: why the black-box model returns a prediction for a given observation?

Global Interpretability: what are the drivers of the black-box predictions?

1.8.1 Local Interpretable Model-Agnostic Explanations (LIME)

LIME modifies a single data sample by perturbing the feature values and observes the resulting impact on the output. LIME tries to understand a black-box model by fitting a simpler model to data that is close to the currently observed datum.

Procedure is:

1. Perturb feature values
2. Run black-box model to get predictions
3. Train an easy to interpret model such as linear regression or decision trees to fit the data set that is created from samples and predictions

1.8.2 Partial Dependence Plot

The partial dependence plot is the expected prediction as a function of the value of a particular feature.

The values of all features except the one under consideration are chosen randomly

1.8.3 Shapley Values

- The SHAP (SHapley Additive exPlanations) method allows us to capture the impact of the different features on the ML output.
- The method borrows from **cooperative game theory** and consists in the calculation of SHAP value, which represents a measure of the **importance** of a feature.
- More precisely, the SHAP value of a feature measures how much it contributes, either positively or negatively, to the classifier prediction.
- The goal of the SHAP method is to explain a prediction computing the contribution of each feature to the prediction itself.
- More precisely, the method shows the contribution of each feature to push the model output from the base value (the average model output over the training dataset) to the model output associated with the observation. Given a single observation, a set of SHAP values, one for each feature, is calculated.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!}(v(S \cup \{i\}) - v(S))$$

Properties

- If a feature never changes the prediction, its contribution is zero.
- If two features are symmetrical in that they affect the prediction in the same way, they have the same contribution.
- For an ensemble model where predictions are the average of predictions given by several underlying models, the Shapley value is the average of the Shapley values for the underlying models.
- Calculation time increases exponentially with the number of features

1.8.4 Global Interpretability in Random Forests

Mean Decrease Accuracy (%IncMSE) This shows how much our model accuracy decreases if we leave out ***that feature.***

Mean Decrease Gini (IncNodePurity) This is the total decrease in node impurities, measured by the Gini Index from splitting on ***the feature***, averaged over all trees.

The higher the value of mean decrease accuracy or mean decrease Gini score, the higher the importance of the feature to our model.

1.9 Reinforcement Learning

Reinforcement learning is concerned with finding a strategy for taking a series of decisions rather than just one. The environment is usually changing unpredictably.

Rewards and costs There are rewards and costs and the algorithm tries to maximize expected rewards (net of costs) in its interaction with the environment. In doing that, should you choose best decision based on evidence to date or try something new?

The time horizon may be

- finite: finite and fixed number of steps
- indefinite: until some stopping criteria is met (*absorbing* states)
- infinite: forever

There are different types of rewards.

- **Total reward**
$$V = \sum_{i=1}^{\infty} r_i$$
- **Average reward**
$$V = \lim_{n\to\infty} \frac{r_1 + \cdots + r_n}{n}$$
- **Discounted reward**
$$V = \sum_{i=1}^{\infty} \gamma^{i-1} r_i$$

We define the **return** v_t as the total discounted reward from time $t+1, \ldots$

$$v_t = r_{t+1} + \gamma r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

- The **discount** $\gamma \in [0, 1)$ is the present value of future rewards.
- The value of receiving reward r after $k+1$ steps is $\gamma^k r$
- **Immediate** reward vs **delayed** reward
  - γ close to 0 leads to myopic evaluation
  - γ close to 1 leads to far-sighted evaluation
- γ can also be interpreted as the probability that the process will go on.

Policies A policy, at any given point in time, **decides** which action the agent selects. A policy fully defines the **behavior** of an agent.

Value function Given a policy π it is possible to define the **utility** of each state: **Policy Evaluation**. The value function $V^\pi(s)$ of a Markov Decision Process (MDP) is the expected return starting from state s, and then following policy π

$$V^\pi(s) = \mathbb{E}_\pi[v_t \mid s_t = s]$$

Action-value function For control purposes, rather than the value of each state, it is easier to consider the value of each action in each state. The action-value function $Q^\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π

$$Q^\pi(s, a) = \mathbb{E}_\pi[v_t \mid s_t = s, a_t = a]$$

Optimal value function The **optimal value function** $V^\star(s)$ is the maximum value function over all policies

$$V^\star(s) = \max_\pi V^\pi(s)$$

The **optimal action-value function** $Q^\star(s,a)$ is the maximum action-value function over all policies

$$Q^\star(s,a) = \max_\pi Q^\pi(s,a)$$

Optimal Policy Value functions define a partial ordering over policies

$$\pi \geq \pi' \iff V^\pi(s) \geq V^{\pi'}(s), \forall\, s \in \mathcal{S}$$

Theorem: for any Markov Decision Process

- There exists an **optimal policy** $\pi^\star$ that is better than or equal to all policies $\pi^\star \geq \pi, \forall\, \pi$
- **All** optimal policies achieve the optimal value function, $V^{\pi^\star}(s) = V^\star(s)$
- **All** optimal policies achieve the optimal action-value function, $Q^{\pi^\star}(s,a) = Q^\star(s,a)$
- There is always a **deterministic optimal policy** for any MDP.

Monte-Carlo (MC) The simplest approach to model learning from purely trial and error is Monte Carlo learning. It's an episodic learning algorithm: **it needs to run an entire episode** (many states, e.g. until the end of the game) until it can use that information for learning. I pick some policy and run through the game, I compute the cumulative discounted reward and divide that reward among every state that I passed in the episode. Updating value function:

$$\boxed{V(s_t)^{\text{new}} \leftarrow V(s_t)^{\text{old}} + \frac{1}{n}\left(v_t - V(s_t)^{\text{old}}\right) \qquad \forall\, t = 1, \dots, n}$$

where n is number of iteration taken to complete the episode.

It's inefficient, takes many episodes and it's assuming that every single step taken to victory or loss is equally important.

I can do the same updating the action-value function:

$$\boxed{Q(s_t,a_t)^{\text{new}} \leftarrow Q(s_t,a_t)^{\text{old}} + \frac{1}{n}\left(v_t - Q(s_t,a_t)^{\text{old}}\right) \qquad \forall\, t = 1, \dots, n}$$

Temporal Difference (TD) A more advanced approach is **Temporal Difference TD(**0**)**, it highlights events that are more recent in giving the rewards: events that happened recently are more correlated to the reward I'm getting. The value function can be written as:

$$V(s_t) = \mathbb{E}[r_t + \gamma V(s_{t+1})]$$

i.e. expected reward now plus discounted value function at the next state I find myself in (Bellman Equation).

I now update the value function in this way:

$$\boxed{V(s_t)^{\text{new}} \leftarrow V(s_t)^{\text{old}} + \alpha\left(r_t + \gamma V(s_{t+1}) - V(s_t)^{\text{old}}\right)}$$

The correction term doesn't average across all the trajectories (unlike MC). The first part of the correction is exactly the expected value function (TD target estimate) and the I subtract the old estimate. The whole correction term is called TD Error and is multiplied by a rate α. TD(0) looks only one instant in the future, one Δt delay between action and rewards, but it can be generalized to TD(N) (n**-step bootstrapping**), as N goes to infinity this method converges to MC.

Q-Learning[8] is essentially TD learning on the Q function:

$$\boxed{Q(s_t,a_t)^{\text{new}} \leftarrow Q(s_t,a_t)^{\text{old}} + \alpha\left(r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)^{\text{old}}\right)}$$

Notice that there is a max over a: when I get to r_t it doesn't have to be the optimal on policy action, I can do some random action. However, when I compute the quality for the next state s_{t+1} I optimize over the action a.

[8] `https://www.youtube.com/watch?v=0iqz4tcKN58`

MC vs TD Control

- TD learning has several advantages over Monte-Carlo (MC)
  - Lower Variance
  - Online
  - Incomplete sequences
- Natural idea: use TD instead of MC in our control loop
  - Apply TD to $Q(s, a)$
  - Use ε-Greedy policy improvement
  - Update every time-step

Exploration vs Exploitation

Example of Greedy Action Selection[9]

- There are two doors in front of you
- You open the left door and get reward 0, $V(\text{left}) = 0$
- You open the right door and get reward $+1$, $V(\text{right}) = +1$
- You open the right door and get reward $+2$, $V(\text{right}) = +2$
- You open the right door and get reward $+2$, $V(\text{right}) = +2$

Are you sure you've chosen the best door?

ε-Greedy exploration is the simplest idea for ensuring **continual exploration**. **All** m actions are tried with **non-zero** probability

- With probability $1 - \varepsilon$ choose the **greedy** action
- With probability ε choose an action at **random**

It makes sense to reduce ε over time. For example we can let it decline exponentially. The initial Q-values makes a difference. It keeps both exploration (visiting random, unknowned states) and exploitation (searching best reward).

[9] `https://www.davidsilver.uk/wp-content/uploads/2020/03/control.pdf`

Chapter 2

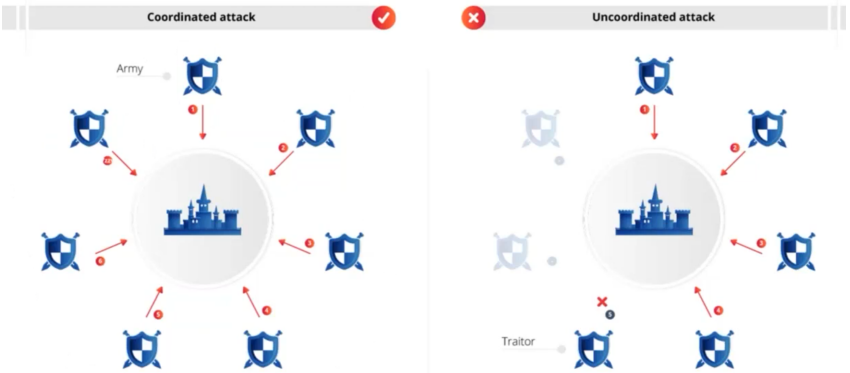
Blockchain (D. Marazzina)

A blockchain is a **DL** (Distributed Ledger[1]) where data are replicated, shared, and synchronized all over the world. **No one is the owner**. Any involved block cannot be altered retroactively, without the alteration of all the subsequent blocks. Blocks are connected in a chain.

- **Trust** is not required in the blockchain;
- It is required in Dropbox, Google Drive, etc.

What happens if a node (by mistake or not) propagates an **incorrect transaction**? The system must not validate it. How? Through a **consensus mechanism**.

The problem of the **Byzantine generals** is a computer problem on how to reach consensus in situations where errors are possible. The problem consists of finding an agreement, communicating only through messages, between different components of conflicting information.



2.1 Bitcoin

Bitcoin is an electronic currency created in 2009 following the famouse paper[2] published in November 2008 by the inventor known under the pseudonym of **Satoshi Nakamoto**.

Keywords:

- Consensus mechanism for lack of trust
- Decentralized database
- Cryptography
- Generation of new money

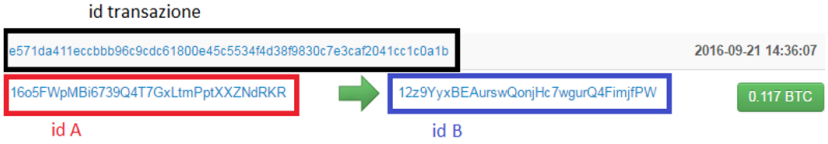
[1] in Italian: *libro mastro*.
[2] `https://bitcoin.org/bitcoin.pdf`

2.2 Decentralization

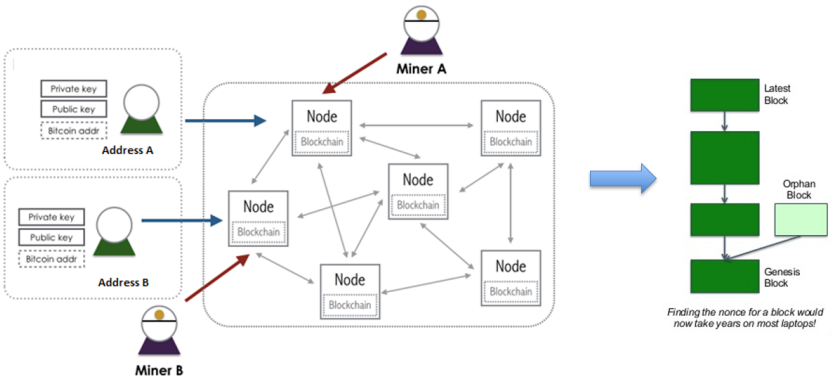
- Politically decentralized: Bitcoin is not controlled by a specific entity.
- Architecturally decentralized (or distributed): Bitcoin lives on machines spread across the planet.
- Logically centralized: all machines that Bitcoin lives on agree on a shared status (e.g. how many Bitcoins each has).

A blockchain is a distributed (or architecturally decentralized) and politically decentralized database managed by a peer-to-peer network through distributed consensus mechanisms (to agree on a shared state, as logically centralized).[3]

- A **transaction** is a Bitcoin transfer that is included in the Blockchain. A transaction is identified by the two addresses (counterparts) and by the amount exchanged. The cryptographic private key signature of the two counterparts provides **proof of their identity** and that the Bitcoins come from the real owner. The signature also **prevents that the transaction (once executed) is altered** by others. The two addresses are publicly identified by a code.



- A **block** is a part of the Blockchain that contains a set of (Bitcoin) transactions. On average, a new block, which includes transactions, is added to the Blockchain every 10 minutes (the system is tuned to make it happen).
- **Confirmation:** to be included in a block, and therefore in the Blockchain, a transaction must be confirmed by several nodes (**miners**), i.e. processed by the network.
- **Mining:** it is a mathematical process performed by the Blockchain nodes on the candidate block: once solved the block (and therefore all the transactions recorded in it) is confirmed and inserted into a block of the Blockchain. As a reward for their service, miners receive newly created Bitcoins (and the transaction fees). Mining 1 new block now makes you earn 6.25 BTC, mining can be done with ad hoc hardware and consumes a lot of energy. Miners accumulate transactions all the time (selecting them with a certain criterion, such as those that offer the highest fees). When they have enough transactions to fill a block (which has a fixed size) they try to solve the mining problem. The first miner that solves the mining problem propagates the block in the network, which is verified (they verify if the *Proof of Work* is valid) by the other nodes and added to their copy of the blockchain.



- **Cryptographic signature**: it is a mathematical mechanism that allows you to prove the identity and ownership of Bitcoins in this case. When your Bitcoin software marks a transaction with your private key, the entire network can recognize you (not as an individual but as an id) and verify that

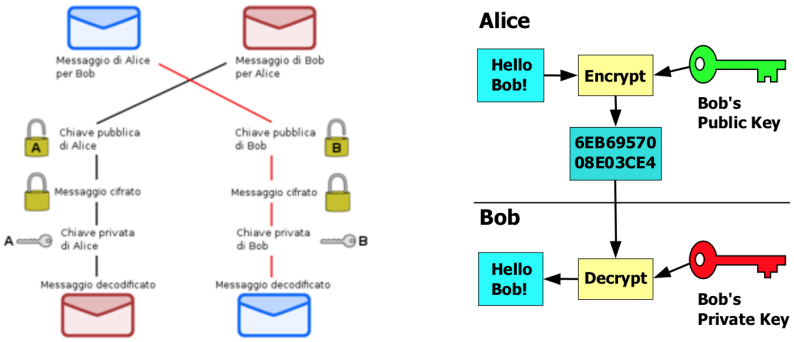
[3] Visit this website to explore it! `https://www.blockchain.com/explorer`

the Bitcoins spent are in your possession. The transaction becomes public with respect to your code. There is currently no way for other nodes to guess your private key, to steal your Bitcoins.

2.3 Asymmetric (or public key) encryption

- Step 1: a public-private key pair is generated for Alice
- Step 2: Bob uses Alice's public key to encrypt the message. Only those who have Alice's private key can decrypt it!

So, if Alice has kept the private key hidden, only she can read the message



Among the asymmetric cryptographies, Bitcoin uses **Elliptical Cryptography**.

Public keys are based on the creation of a mathematical problem that is very difficult to solve without information, but which – with the use of some information (the private key) – becomes easy and quick to solve. The user publicly distributes the problem (the public key) and keeps the additional information hidden (the private key).

An example: RSA encryption.

Let's take two prime numbers, `13` and `7` (and don't reveal them to anyone).

Let's multiply them together: we get `n=91`.

We choose a public key: `5`.

We use the *Extended Euclidean Algorithm* with inputs `5`, `7` and `13`, and we get the private key: `29`.

So `n=91`, public key `5`, private key `29`.

Let Alice and Bob know that `n=91`, Alice's public key is `5`, and her private key is `29`.

Bob wants to send a message to Alice: the message is `3`.

Then multiply `3` by itself `5` (public key) times, modulo `n=91`: `3*3*3*3*3=243`, `mod(243,91)=61`.

The public message is `61`.

By doing the opposite we can decode the message.

```
a=3; at=3;
for i=1:5-1
    at=mod(at*a,91)
```

```
end
Output=61.

a=61; at=61;
for i=1:29-1
    at=mod(at*a,91)
end
Output=3.
```

We have decoded the message!

- Alice and Bob know `n=91`.
- Everyone knows `5`.
- Only Alice knows `29`.
- Knowing `5` (public key) is not sufficient to decrypt the message.
- Can Bob trace Alice's private key? That is, is the information `91` sufficient to compromise security?
- Mathematical problem: factorization into prime numbers.

Prime factorization is not the most difficult problem on a bit-to-bit basis. Specialized algorithms such as Quadratic Sieve and General Number Field Sieve were created to address the factorization problem and have had some success. These algorithms are faster and less computationally intensive than the naive approach of guessing pairs of known primes.

These factoring algorithms become more efficient as the size of the numbers considered increases. The gap between the difficulty of factoring large numbers and multiplying large numbers narrows as the number (i.e. the key bit length) increases. As the resources available to decrypt the numbers increase, the size of the keys must grow even faster. This is not a sustainable situation for mobile and low-power devices that have limited computational power. The gap between factoring and multiplication is not sustainable in the long term.

All of this means that RSA is not the ideal system for the future of cryptography.

Techniques based on Elliptical Curves: `https://youtu.be/2RShPGAZqMs`.

Asymmetric algorithms guarantee confidentiality in communication. A message encrypted with the recipient's public key ensures that only the latter can decrypt this message, as it is the only one who has the corresponding private key.

Furthermore, by reversing the use of keys, that is, encrypting with the sender's private key and decrypting with the sender's public key, it is possible to guarantee authentication. It is on this principle that the digital signature is based.

The message is encrypted with the private key, so that anyone can, using the public key known by everyone, decrypt it and, in addition to being able to read it in clear text, be sure that the message was sent by the owner of the private key corresponding to the public one used to read it.

2.4 What about miners?

In addition to checking that the digital signatures of the transactions are authentic, that there is no double-spending, etc., and therefore verifying the transaction, their main job is to **build the Blockchain**, block by block. How? By concatenating the blocks together, this happens with the **solution of a computationally onerous mathematical problem**. If it were easily fixed, it would also be easy to **attack** the Blockchain.

2.5 Proof of Work

The Bitcoin Blockchain is therefore based on *Proof of Work*.

Proof of Work, or PoW, is an algorithm that is used by different cryptocurrencies – such as Bitcoin and Ethereum – to reach a decentralized agreement between different nodes in the process of adding a specific block to the blockchain.

Hashcash (SHA-256)[4] is the Proof of Work feature used by Bitcoin. Cryptocurrency forces miners to solve extremely complex and computationally difficult mathematical problems to add blocks to the blockchain. This function produces a very specific type of data that is used to verify that a significant amount of work has been done.

A hash function takes an input of *any* length and creates an output of *fixed* length.

```
a0680c04c4eb53884be77b4e10677f2b
```

This is referred to as the *message digest*.

It is also known as the digital fingerprint.

Every transaction has a hash associated with it. In a block, all of the transaction hashes in the block are themselves hashed (sometimes several times, the exact process is complex), and the result is the Merkle root. In other words, the Merkle root is the *hash of all the hashes of all the transactions in the block*.

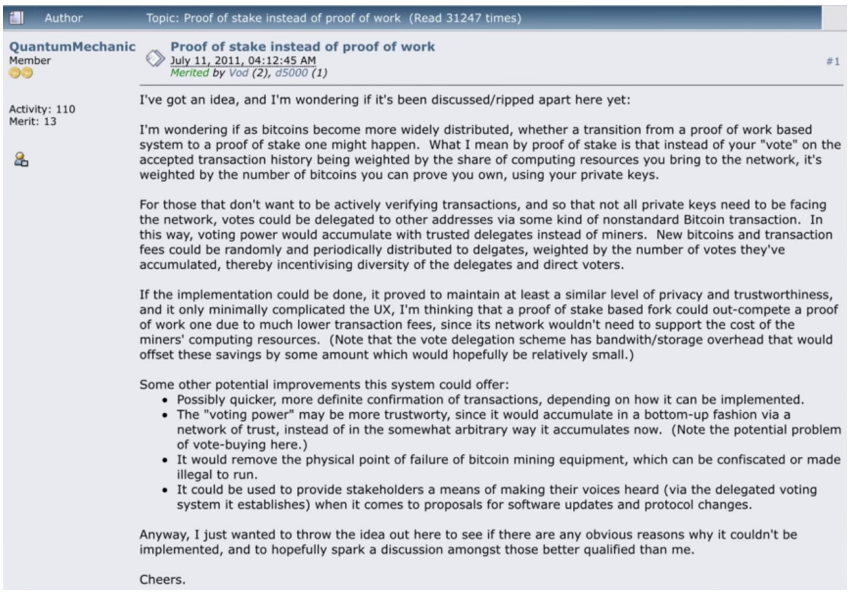
Our target is to **find a variation of it that SHA-256 hashes to a value beginning with '000'**. We vary the string by adding an integer value to the end called a `nonce` and incrementing it each time. Finding a match for "Hello, world!" takes us 4251 tries (but happens to have zeroes in the first four digits).

A PoW system **discourages attacks** and other abuse of service by enforcing certain jobs that require a computer's high processing time. A key feature of these works is their asymmetry: the work must be moderately, but easy to control.

Game theory: the nodes of the network compete with each other to register the new block thus obtaining compensation. Thanks to the expected remuneration, most of the nodes have an **interest in behaving in a *legitimate* manner**, thus contributing to the growth of the blockchain.

2.6 Proof of Stake

Proof of Stake is an alternative method, a way by which nodes reach a consensus.



Author | Topic: Proof of stake instead of proof of work (Read 31247 times)

QuantumMechanic
Member

Activity: 110
Merit: 13

Proof of stake instead of proof of work
July 11, 2011, 04:12:45 AM #1
Merited by Vod (2), d5000 (1)

I've got an idea, and I'm wondering if it's been discussed/ripped apart here yet:

I'm wondering if as bitcoins become more widely distributed, whether a transition from a proof of work based system to a proof of stake one might happen. What I mean by proof of stake is that instead of your "vote" on the accepted transaction history being weighted by the share of computing resources you bring to the network, it's weighted by the number of bitcoins you can prove you own, using your private keys.

For those that don't want to be actively verifying transactions, and so that not all private keys need to be facing the network, votes could be delegated to other addresses via some kind of nonstandard Bitcoin transaction. In this way, voting power would accumulate with trusted delegates instead of miners. New bitcoins and transaction fees could be randomly and periodically distributed to delgates, weighted by the number of votes they've accumulated, thereby incentivising diversity of the delegates and direct voters.

If the implementation could be done, it proved to maintain at least a similar level of privacy and trustworthiness, and it only minimally complicated the UX, I'm thinking that a proof of stake based fork could out-compete a proof of work one due to much lower transaction fees, since its network wouldn't need to support the cost of the miners' computing resources. (Note that the vote delegation scheme has bandwith/storage overhead that would offset these savings by some amount which would hopefully be relatively small.)

Some other potential improvements this system could offer:

- Possibly quicker, more definite confirmation of transactions, depending on how it can be implemented.
- The "voting power" may be more trustworty, since it would accumulate in a bottom-up fashion via a network of trust, instead of in the somewhat arbitrary way it accumulates now. (Note the potential problem of vote-buying here.)
- It would remove the physical point of failure of bitcoin mining equipment, which can be confiscated or made illegal to run.
- It could be used to provide stakeholders a means of making their voices heard (via the delegated voting system it establishes) when it comes to proposals for software updates and protocol changes.

Anyway, I just wanted to throw the idea out here to see if there are any obvious reasons why it couldn't be implemented, and to hopefully spark a discussion amongst those better qualified than me.

Cheers.

Figure 2.1: First proposal of the Proof of Stake.

In Proof of Stake you stake your money apart, and the greater the stake, the higher is the probability that you will be chosen to create a new block. Blocks are not mined, but minted. The idea is that you should not fraud the system, for example by adding fake transactions to the block, otherwise the money you put at stake will be taken from you. Provided you stake enough money, it should be more convenient to be fair. Moreover you are incentivized to do it because you get a reward.

[4]Visit this site to try some hashing: `https://emn178.github.io/online-tools/sha256.html`

The real problem is how to choose the validators, because rich people have more money to stake, thus more chances of being chosen as validators and becoming even richer.

There is no waste of energy, and there is no vulnerability to 50%+1 attacks.

2.7 Uniqueness

The chain is unique: if two different chains are created (for example, due to a delay in updating the blocks), the *incorrect* one is automatically deleted and overwritten by the correct one.

To determine the correct chain, there are mechanisms: the chain considered correct is always the **longest one** (longest chain rule). **The blockchain rewards computational effort.**

Generally, the longest chain is also the one stored by the majority of the network nodes (50%+1 rule), given that **miners are constantly incentivized to create valid blocks** through the fees and Bitcoins generated at each resolution of the mining problem.

From a game theory perspective, collaborating on the longest chain is a balance, as no one can achieve a better result for themselves (without collaborating with others) by creating invalid blocks (PoWs). **This makes it very difficult for a hacker to change even the last block**.

In **March 2013**[5][6][7] there was a problem, because the Bitcoin software was being updated from version 0.7 to 0.8. It happened that a new block was created by a miner who had version 0.8, and was correctly recognized to miners who had 0.8, but those with version 0.7 rejected it.

In the end miners decided to switch back to 0.7, since "*we cannot get every bitcoin user in the world to now instantly switch to 0.8 so no, we need to rollback to the 0.7 chain*" (Pieter Wuille). Eventually 24 blocks would be lost.

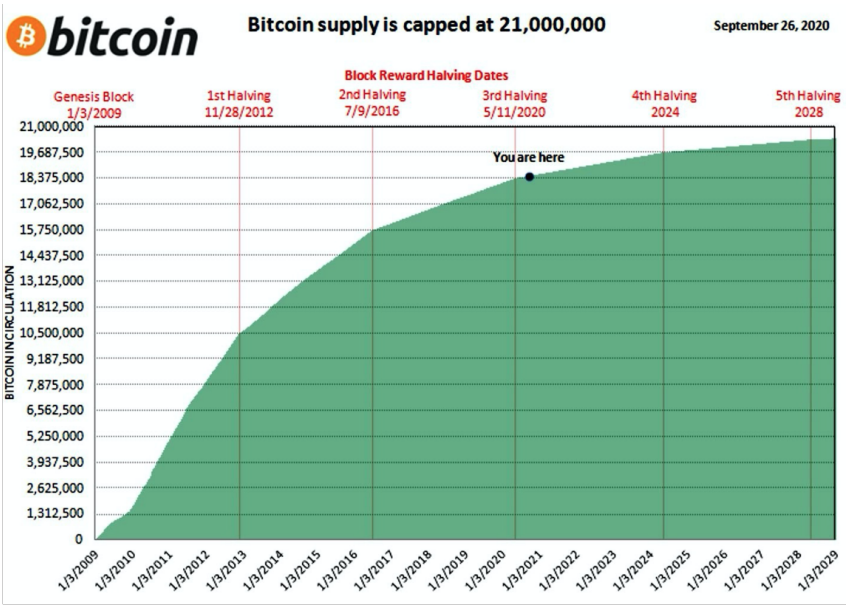
```
23:06  Luke Dashjr        so??? yay accidental hardfork? :x
23:06  Jouke Hofman       Holy crap


23:22  Gavin Andresen     the 0.8 fork is longer, yes? So majority hashpower is 0.8....
23:22  Luke Dashjr        Gavin Andresen: but 0.8 fork is not compatible earlier will be accepted by
all versions

23:23  Gavin Andresen     first rule of bitcoin: majority hashpower wins
23:23  Luke Dashjr        if we go with 0.8, we are hardforking


23:24  Luke Dashjr        so it's either 1) lose 6 blocks, or 2) hardfork for no benefit
23:25  BTC Guild      We'll lose more than 6

23:43  BTC Guild      I can single handedly put 0.7 back to the majority hash power I just need
confirmation

23:44  Pieter Wuille      BTC Guild: imho, that is was you should do, but we should have consensus
first
```

Figure 2.2: `BTC Guild` was a key player in solving the March 2013 fork, having at disposal 20-30% of the hashpower.

2.8 How the nodes get paid

Who pays the nodes? **The system itself**. When a new block is added, besides the fees that people paid with the transactions inside it to incentivize the miner to add them to the block, the miner also receives a quantity from the system, meaning that *everyone agrees* that they made enough computational effort. In other words, if everyone in the world suddenly decides to agree that you have a million dollars, even if you physically don't, you can go buy a Lamborghini, and then your fictitious money gets transferred to the company. It looks strange, but if you think about it, it's the process of printing virtual money for central governments to make inflation happen.

[5] `https://github.com/bitcoin/bips/blob/master/bip-0050.mediawiki`
[6] `https://freedom-to-tinker.com/2015/07/28/analyzing-the-2013-bitcoin-fork-centralized-decision-making-saved-the-day/`
[7] `https://twitter.com/gavinandresen/status/311290936527298561`



2.9 How much does it cost to produce a Bitcoin?

Until a few years ago a good computer was enough to produce (mine) Bitcoin, today thousands of very powerful processors and graphics cards are needed.

For the extraction of Bitcoins, computers with specific processors called ASICs (application-specific integrated circuits) have been developed; these processors, in addition to having a high cost, also involve a **high consumption of electricity**.

It's *not real-time* technology, it may take up to 1-2 hours to complete a transaction.

2.10 Notarization

Notarization is a very useful **application** of the Blockchain. Suppose you want to prove that a certain document (a thesis for example) was ready on a given day.

Suppose the hash of the file is recorded in the blockchain. All parties having access to the blockchain can individually verify the authenticity of these files.

It is not possible to attach documents to the blockchain. The procedure is the following:

1. The document is hashed. This means that the content of the document is *summarized*, or better represented, by a string of 64 characters (256bit).
2. The 64-character string is then put into a Bitcoin (or other) blockchain transaction using the `OP_RETURN` **field**, it's a *free* field presented in every Bitcoin transaction: free, but limited in size.
3. A small Bitcoin payment is made to process the transaction and *register it on the blockchain* (**you can also make a 0 BTC transaction**, so paying only the cost of the transaction – variable but around 0.00001 BTC).

Hashing is *irreversible* once you know the hash, it is *mathematically impossible* to deduce the original text.

Is it possible that two different files report the same Hash?

The hash projects a (theoretically infinite) sequence of data, in a finite hash and is usually much shorter than the original file, so duplicates can happen. Possible, but *very unlikely*! And, with the most advanced hashing techniques, it is difficult (if not impossible) to use this vulnerability (known as **Collision**) to modify documents for illegal purposes. And then there is the avalanche effect, a property where a small variation in input produces a considerable variation in the hash.

2.11 Scalability

It is not yet possible to think of a DLT that **completely replaces the system of financial transactions** (too large-scale). But it would be possible for two counterparties to create a DLT and use it for their transactions.

The cryptography used in Blockchain could certainly be considered to make current databases more secure.

Not just finance: the Blockchain system can be used as a register in which to enter any type of information and consequently also a contract, an act or a certificate, avoiding the intermediation of third parties, but maintaining the guarantee of advertising.

Comparison of transactions capacity[8]

- Bitcoin – 3 to 4 transactions per second (1 block every 10 minutes);
- Ethereum – 20 transactions per second (1 block every 15 seconds);
- PayPal – 193 transactions per second average;
- Visa – 1.667 transaction per second.

PoW is **expensive:** it is the main cause of scalability problems (as well as memory problems: all nodes should store all transactions).

Solutions under study: off-chain operations, Proof of Stake, Proof of Work only on some random blocks, permissioned ledgers.

Permissioned ledgers (or private blockchains) can be controlled, and therefore can have ownership. When a new data or record is added to the blockchain, the approval system is not bound to the majority of participants but to a limited number of actors who can be defined as trusted.

Three false myths?

1. Less expensive system
   (a) Bitcoin transactions on the Blockchain cost (nearly) nothing.
   (b) But this is possible because new Bitcoins are created for the remuneration of the network nodes that work for the proper functioning of the system.
2. Safer
   (a) Potentially yes: safer in terms of immutability (less than 50%+1 attacks), but not privacy.
   (b) The system of remuneration of the nodes pushes to use their own computing power to make the system work, not to try to force it.
   (c) The database is distributed all over the world.
3. Faster
   (a) The current financial transaction system is *slow* not for a technological issue, but for a consensus issue.
   (b) "Consensus by reconciliation": a process that the financial markets have chosen as their "checks and balances" system.
   (c) It is, therefore, a question of moving to a new form of "decentralized consensus": an automatic mechanism that allows transactions to be quickly validated. Like in Bitcoin!

Read here to see how to verify a block on the blockchain!

`https://blockchain-academy.hs-mittweida.de/courses/blockchain-introduction-technical-beginner-to-intermediate/lessons/lesson-13-Bitcoin-block-hash-verification/topic/how-to-calculate-and-verify-a-hash-of-a-block/`

2.12 Cryptoassets (D. Marazzina)

Every coin that is not Bitcoin is referred to as **altcoin**.

A **cryptoasset** (coin) is a financial asset that is created and exchanged on a platform; like a blockchain,

[8] `https://altcointoday.com/Bitcoin-ethereum-vs-visa-paypal-transactions-per-second/`

but not only a blockchain.

Assets are divided into

- **cryptocurrencies:** the native currency of the blockchain (Bitcoin)
- **tokens:** crypto asset not native of the technology

Is Bitcoin a token? Not exactly. While all cryptocurrencies are technically tokens, Bitcoin is usually not considered a token, but a coin. This is because it is generated by the blockchain that supports it. Tokens, on the other hand, operate on existing blockchains. In fact, a single blockchain offers space for **many different tokens**. Suffice it to say that the Ethereum network currently hosts almost half a million.

It's not easy to *diversify* cryptoassets because correlation is very high.

It may happen that there are so called **forks** on the Blockchain:

- **Soft fork:** when there are two versions of the Blockchain and it's solved automatically.
- **Hard fork:** when something (a software update that only some install) creates two versions of the Blockchain.

Polkadot Multi-chain etherogeneous and scalable technology that gives freedom on the structure and the chains of its network (Ethereum is stricter). Consensus is granted on all the chains, which all communicates.[9] **DOT** is the cryptocurrency of the Polkadot platform.

Ripple Built for enterprise use, Ripple aims to be a fast (4 seconds), cost-efficient cryptocurrency for cross-border payments. Within the Ripple system, money isn't actually transferred from one place to another: only the promise of payment is transferred. It's designed to find the most convenient way to exchange two cryptocurrencies. The transaction fee is paid and then destroyed by the network.[10] **XRP** is the cryptocurrency used by the Ripple payment network.

Bitcoin Cash. It's a cryptocurrency that is a fork of Bitcoin that was created in 2017:

- On 21 July 2017 Bitcoin miners locked in a software upgrade referred to as **Bitcoin Improvement Proposal (BIP) 91: the Segregated Witness (Consensus layer).**
- SegWit alleviates the scaling problem by enabling the Lightning Network, an overlay network of micropayment channels, hypothetically resolving the scaling problem by enabling virtually unlimited numbers of instant, low-fee transactions to occur *off chain*.
- By 8 August 100% of the Bitcoin mining pools signaled support for SegWit.
- A small group of mostly China-based Bitcoin miners, that were unhappy with Bitcoin's proposed SegWit improvement plans, pushed forward alternative plans for a split which **created Bitcoin Cash**. The proposed split included a plan to increase the number of transactions its ledger can process by increasing the block size limit.

In 2018 Bitcoin Cash subsequently split into two cryptocurrencies: Bitcoin Cash, and Bitcoin SV.

Bitcoin SV.

- On 15 November 2018, a hard fork chain split of Bitcoin Cash occurred between two rival factions called Bitcoin Cash and Bitcoin SV.
- The split originated from what was described as a *civil war* in two competing Bitcoin cash camps.
- The first camp promoted the software entitled Bitcoin ABC (short for Adjustable Blocksize Cap) which would maintain the block size at 32 MB.
- The second camp put forth a competing software version Bitcoin SV, short for *Bitcoin Satoshi Vision*, that would increase the block size limit to 128 MB.

Bitcoin Gold. Another hard fork of Bitcoin.

[9] `https://en.wikipedia.org/wiki/Polkadot_(cryptocurrency)`
[10] `https://en.wikipedia.org/wiki/Ripple_(payment_protocol)`

- Bitcoin Gold is a fork of Bitcoin that seeks to reduce the influence of miners who use specialized equipment known as ASICs[11]. The team's stated goal is to "make Bitcoin decentralized again."
- Bitcoin Gold hard forked from the Bitcoin blockchain on October 24, 2017.
- In May 2018, Bitcoin Gold was hit by a 50%+1 hashing attack by an unknown actor. This type of attack makes it possible to manipulate the blockchain ledger on which transactions are recorded and to spend the same digital coins more than once. During the attack, 388.000 BTG (worth approximately $18M) was stolen from several cryptocurrency exchanges.
- Bitcoin Gold suffered from 50%+1 attacks again in January 2020[12].

TheDAO incident [13]

The Decentralized Autonomous Organization (known as TheDAO) was meant to operate like a venture capital fund for the crypto and decentralized space. The lack of a centralized authority reduced costs and in theory provides more control and access to the investors.

On **June 17, 2016**, a hacker found a loophole in the coding that allowed him to drain funds from TheDAO. In the first few hours of the attack, 3.6M ETH were stolen, the equivalent of $70M at the time.

It's important to understand that this bug did not come from Ethereum itself, but from this one application that was built on Ethereum.

If the blockchain is immutable, how can it be canceled?

Two solutions:

- **Soft Fork**: doing a software update to lock the funds (but would remain on the attackers wallet).
- **Hard Fork**: go back to a previous version of the blockchain.

A hard fork was made, and the blockchain was rewritten by eliminating the transaction.

How? With distributed consent! More than 50% of Ethereum users have accepted the update. The new separate version became **Ethereum (ETH)** with the theft reversed, and the original continued as **Ethereum Classic (ETC)**.

Distributed consent overcomes the immutability of the Blockchain.

2.13 Derivatives

A **futures contract** is an agreement between counter-parties to buy or sell an asset at an explicit price and date in the future. The buyer is obligated to buy the underlying asset a specific price once the contract expires, and the seller is required to furnish the asset at the time of expiry.

In **inverse futures contract** BTC are the coin while USD are the commodity.

BTC Perpetual Swap Theese are perpetual inverse futures contract. *Perpetual* means without expiration. For example I give you 20 BTC and you give me 40 USD, I'm short on BTC; if BTC goes down you're losing – paying me. Every 8 hours we repeat. It's a way of **funding**, the longs will pay and the shorts will receive.

2.14 Decentralized Finance (DeFi)

DeFi is an ambitious attempt to decentralize core traditional financial use cases like trading, lending, investment, wealth management, payment and insurance on the blockchain. Good trust on Ethereum, even though usability is not awesome.

Lightning Network Layer 2 payment protocol on top of a blockchain cryptocurrecy where payments can happen fast solving the scalability problem. The idea is: you open a *payment channel* by queueing

[11] `https://academy.bit2me.com/it/que-son-mineros-asic/`
[12] `https://thenextweb.com/hardfork/2020/01/27/Bitcoin-gold-51-percent-attack-blockchain-reorg-cryptocurrency-binance-exchange/`
[13] `https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee`

it in the blockchain, then you make a lot of fast transactions, finally you close the channel and only the final version of the transactions in broadcasted.

2.15 Fungible and Non-fungible Token (NFT)

Parameters	Fungible	Non-Fungible
Exchangability	Fungible tokens can be exchanged with other tokens of the same type	Non-Fungible tokens cannot be exchanged with similar type NFT's. For eg:- A car cannot be exchanged with another car
Uniformity	All Fungible tokens are identical to each other	NFT's are unique and not similar to each other
Fractionalisation	Fungible tokens can be divided into smaller units. For eg: a $100 note can be exchanged with another $100 or two $50 tokens	NFT's cannot be divided but are one entire unit

CryptoKitties is a blockchain game on Ethereum developed by Axiom Zen that allows players to purchase, collect, breed and sell virtual cats. It is one of the earliest attempts to deploy blockchain technology for recreation and leisure. The game's popularity in December 2017 congested the Ethereum network, causing it to reach an all-time high in number of transactions and slowing it down significantly.

CryptoKitties operates on Ethereum's underlying blockchain network, as a **non-fungible token**, unique to each CryptoKitty.

Several traits can be passed down from the parents to the offspring. There are a total of 12 *cattributes* for any cat, including pattern, mouth shape, fur, eye shape. Other features like cool down times are not passed down but are instead a function of the *generation* of the offspring, which is one higher than the maximum generation between the two parents. In December 2017 a CryptoKitty sold for $100.000.

A CryptoKitty's ownership is tracked via a smart contract on the Ethereum blockchain.

Ethereum's Tokens Any **fungible** commodity can be represented by a token derived from **ERC-20 standard**. **NFTs** can be derived from **ERC-721 standard**. This recent standard has made possible to represent goods like pieces of art, personalized access rights, properties.

- More than 80% of DeFi solution is on Ethereum
- Algorand is a promising technology (but young)

Purpose Parameters	**Class**	Coin / Cryptocurrency		Utility Token		Tokenised Security	
	Function	Asset-Based Token		Usage Token		Work Token	
	Role	Right	Value Exchange	Toll	Reward	Currency	Earnings
Governance Parameters	**Representation**	Digital		Physical		Legal	
	Supply	Schedule-based	Pre-mined, scheduled distribution	Pre-mined, one-off distribution		Discretionary	
	Incentive System	Enter Platform	Use Platform	Stay Long-Term		Leave Platform	
Functional Parameters	**Spendability**	Spendable		Non-Spendable			
	Tradability	Tradable		Non-Tradable			
	Burnability	Burnable		Non-Burnable			
	Expirability	Expirable		Non-Expirable			
	Fungibility	Fungible		Non-Fungible			
Technical Parameters	**Layer**	Blockchain (Native)		Protocol (Non-Native)		Application (dApp)	
	Chain	New Chain new Code	New Chain, forked Code	Forked Chain, forked Code		Issued on top of a protocol	

Figure 2.3: Classification of tokens.

Purpose Parameters	**Class**	Coin / Cryptocurrency		Utility Token		Tokenised Security	
	Function	Asset-Based Token		Usage Token		Work Token	
	Role	Right	Value Exchange	Toll	Reward	Currency	Earnings
Governance Parameters	**Representation**	Digital		Physical		Legal	
	Supply	Schedule-based	Pre-mined, scheduled distribution	Pre-mined, one-off distribution		Discretionary	
	Incentive System	Enter Platform	Use Platform	Stay Long-Term		Leave Platform	
Functional Parameters	**Spendability**	Spendable		Non-Spendable			
	Tradability	Tradable		Non-Tradable			
	Burnability	Burnable		Non-Burnable			
	Expirability	Expirable		Non-Expirable			
	Fungibility	Fungible		Non-Fungible			

Figure 2.4: Example of a restaurant voucher.

2.16 Coin vs Token

Coin It's a unit of value that is native to a blockchain. It is a means of exchange within the blockchain to incentivize the network of participants to use the blockchain. The cryptocurrencies Bitcoin, Ether, Ripple, and Litecoin are all examples of native cryptocurrencies. The sole purpose of a cryptocurrency is for exchange of value, and it has limited functionality beyond that.

Token It's a piece of business logic (i.e., "smart contract") coded into an existing blockchain. A token can have a functionality beyond an exchange of value – it can represent any asset or functionality desired by the developer for use on a platform. Most tokens are created on the Ethereum blockchain by using ERC-20 smart contracts, such as Tron and VeChain.

Coins and tokens differ not only for the IT definition, but also for **market behaviour**.

2.17 How to create a crypto?

Tokens are often released through a crowdsale known as an initial coin offering (ICO) in exchange for existing coins, which in turn fund projects like gaming platforms or digital wallets. You can still get publicly available tokens after an ICO has ended-similar to buying coins-using the underlying currency to make the purchase.

1. **Build Your Own Blockchain** or **Fork an Existing One**. Both methods require quite a bit of technical knowledge. You can fork an existing blockchain by taking the open-source code found on Github making a few changes, and launching a new blockchain with a new name.
2. Launch a Coin or Token Using a Cryptocurrency Creation Platform – all you have to do is to choose a platform, pay for this service, and then enter the parameters, from the logo to the number of coins awarded for signing a block. This is faster, simpler, and cheaper than creating a coin because it doesn't require the time and effort to build and maintain a new or forked blockchain and instead relies on the technology already in use for Bitcoin or Ethereum.

Then you need to find someone which will buy the coin or token. This is not easy: you need a community interested in your coin/token: usually it is easier if the coin/token is connected to a real project.

Plenty of cryptocurrencies are unsuccessful, even questionable from a legal standpoint, because the ICO wasn't created in good faith or the coin failed to generate lasting interest.

2.18 Exchange

You can visit `https://blog.coingecko.com/trust-score/` to see the best exchanges according to some score.

Quadriga Gerald Cotten created Quadriga, a Canadian exchange, where he controlled fake accounts with fake money. People exchanged real money with these fake accounts and were scammed for $135M. See
`https://www.youtube.com/watch?v=NMDZcbkgpJA`
`https://www.youtube.com/watch?v=jT55oLyT0ac`

Bitgrail Nearly one year after Francesco Firano, the owner and operator of the Bitgrail cryptocurrency exchange, announced that the exchange had lost 17M NANO (approximately $170M), an Italian Bankruptcy Court and a court-appointed technical expert concluded that Mr. Firano ("The Bomber" as Mr. Firano called himself on social media) was at fault for the loss and is required to return as much of the assets to his customers as possible. The ruling is a landmark decision that sets an important precedent for the protection of cryptocurrency users worldwide. In its decision, the court concluded that both Bitgrail and Mr. Firano, personally, be declared bankrupt, authorizing seizures of many of Mr. Firano's personal assets. So far, authorities have seized over $1M in personal assets, including Mr. Firano's car. Millions of dollars in cryptocurrency assets have been seized from Bitgrail's exchange accounts and moved to accounts managed by trustees appointed by the Court. The Court found that the NANO reported lost by Mr. Firano on February 9, 2018, had actually been removed from the exchange months earlier, between July 2017 and December 2017. The Court criticized Mr. Firano for not immediately taking steps to account for the losses.

2.19 Smart Contracts

Ethereum

- It is another platform developed on blockchain technology.
- In order to be able to run on the peer-to-peer network, Ethereum users pay for the use through a cryptocurrency, called Ether, which therefore acts both as a cryptocurrency and as a fuel.
- Ethereum is different from the Bitcoin blockchain since **it is constructed to create Smart Contracts**. Even in Bitcoin, elementary smart contracts can be created. In Ethereum the peculiarity is that they are **listed on a Turing-complete language**.
- A smart contract is a program that runs on Ethereum's decentralized virtual machine.
- The smart contract cannot act actively (for example, it cannot monitor the status of another smart contract and act accordingly), but it acts reactively (a user, not necessarily directly, must invoke it to perform any operation).
- The smart contract cannot interrogate the world outside the blockchain. For example, in the case of a sports bet managed through a smart contract, this cannot obtain the result of the game from a website and act accordingly.
- The smart contract can instead obtain information using **oracles**.
- Oracles are smart contracts in which information from the outside world is loaded in exchange for a reward.

Example. AXA is testing insurance on late flights using smart contracts.

Collateral Put simply, collateral is an item of value that a lender can seize from a borrower if he or she fails to repay a loan according to the agreed terms. One common example is when you take out a mortgage. Normally, the bank will ask you to provide your home as collateral. Collateral acts as a **guarantee** that the lender will receive back the amount lent even if the borrower does not repay the loan as agreed.

Example. I want to create a tourism insurance contract against rain. Bob signed a tourism insurance contract for an holiday in New York for May 2, 2020. This contract will reimburse the payment for his trip ($200) if that day it will rain. In the insurance contract there is this clause which defines the meaning of "it will rain". Rainfall amount is described as the *depth of water reaching the ground*, typically in inches or millimeters (25 mm equals one inch). It rains more than 0.1 inch on that day, according to the Laguardia Airport Station result, reported by the website `https://www.wunderground.com/history/monthly/us/ny/new-york-city/KLGA`.

We need an **oracle**.

1. The insurer creates the code of the smart contract.
2. The insurer pays a third part to be an oracle, i.e. to report the website information to the smart contract.
3. The insurer is ready to sell the insurance contract.

Why Bob should prefer a blockchain-based insurance? Based on a **third part**, which has no reason to say that it did not rain, if it happened. **Automatic payment**: if the oracle tells to the smart contract that it happened, the smart contract pays.

What happens if Bob wants to be paid in fiat money?

The smart contract can only say "this contract is to be paid", for example creating a null transaction with this sentence in the `OP_RETURN` field. This is written on the blockchain, which is immutable. Then, the insurer, once he/she reads this on the blockchain can tell to his/her bank "please, pay". So, not a real automatic payment without cryptocurrencies.

Possible solution: Stablecoins.

2.20 Coin Offerings

2.20.1 Initial Coin Offering (ICO)

Origins of the ICO J. R. Willett needed to fund his idea (MasterCoin, a *supplementary protocol to Bitcoin* with built-in support for custom tokens), so he asked for some BTC donations. He also added several features inside the protocol that would only be available to those owning MasterCoins. Willet introduced the **first ICO**, and the first utility token. Total raised: $600K.

The MasterCoin protocol turned out to be incredibly successful. Rebranded to and known today as **Omni Layer (OMNI)**, it now serves as the underlying protocol for the **Tether token (USDT)**.

NextCoin (NXT) is a peculiar second ICO.

Ethereum In 2014 Vitalik Buterin tries raising funds for a new and noteworthy ICO for what he envisions the world's first zero-infrastructure platform, named **Ethereum**. The token sale raised 3700 BTC in the first 12 hours. Total raised: $18.3M.

Since Ethereum's inception, over 1.000 **ERC20** cryptocurrencies have been issued on the platform since its initial ICO. ERC20 itself is a protocol standard that defines certain rules and standards for issuing tokens on Ethereum's network and infrastructure. ERC stands for *Ethereum Request For Comments* and *20* stands for a unique ID number to distinguish this standard from others.

In 2016 a decentralized autonomous organization called TheDAO, a set of smart contracts developed on the platform, raised a record $150M in a crowd/token sale to fund the project. **TheDAO was exploited** in June when $50M in Ether were taken by an unknown hacker. Subsequently, Ethereum was split into two separate blockchains (see 2.12) – the new separate version became **Ethereum (ETH)** with the theft reversed, and the original continued as **Ethereum Classic (ETC)**. **Solidity**, initially proposed by Gavin Wood in August 2014, is the primary **programming language** (contract-oriented) used on the Ethereum platform.

Fall of ICO According to the ICORating, in the second half of 2018, the profitability of investments in blockchain startups decreased by 22%, and 58% of ICO projects announced in Q4 2018 were not able to raise more than $100.000. Moreover, in Q4 2018 40% of projects with previously announced ICOs have already deleted their social network accounts and websites

Cayman Island and British Virgin Islands rank among top ICO countries volume-wise.

Disadvantages and limitations The biggest disadvantage of an ICO investment is the **risk**. The market is volatile, and you never know the actual intentions of a newly minted company. The first risk can be called **ordinary fraud** when the team of the project pursues only one goal: to collect investors' money. In addition, since **there are no laws at the moment** that would regulate the conduct of cryptocurrency crowdsales from the position of an investor, it cannot be ruled out that the project may

not live to the stage of product appearance or disappoint the investor with its implementation. Based on the statistical research provided by Satis Group, premier ICO advisory company, approximately 81% of ICO's are **scams**, 6% Failed, 5% had **gone dead**, and 8% went on to trade on an exchange.

Actually, one of the main reasons for such statistics might be **lack of token holders' control over their investments**, the absence of bills and laws regulating the legal field in the sphere of ICO.

The other serious threat is **hackers' attacks**. Research, conducted by Ernst&Young (2017), showed that more than 10% of all funds raised by ICO was stolen by cybercriminals. Analysts examined 372 ICOs conducted between 2015 and 2017. The monthly amount of losses from hackers in an ICO is $1.5M. Moreover, **attackers often manage to gain access to personal data from investors**: from their addresses and phone numbers to billing information.

Before launching ICO, the development team specifies tasks, for which funds should be raised, and indicates in its whitepaper 2 figures: the minimum and maximum, which are called **Soft Cap** and **Hard Cap**.

- Hard Cap – defines the final goal, the upper limit of the amount of money invested, the most desired result. This is a very important indicator, precisely, because **many cryptocurrencies have a limit on the total number of units in circulation**. This, in turn, is one of the most important factors influencing the value of the coin, naturally, in addition to supply and demand.
- Soft Cap – the minimum required amount of the investments for the team to proceed the project implementation in accordance with the plans. If it is not reached within the specified period, the contract is closed and automatically returns all funds raised to depositors. If Hard Cap is achieved, the sale of tokens stops. However, after overcoming soft cap investors control only purchased tokens and can't monitor invested money or withdraw part of the investment.

Finally, the big concern is the **connection between the token holders and holding company** and several relevant questions arise, for example, **what happens if the company which issued the tokens is sold or will the token holders have any rights under the new management**?

2.20.2 Decentralized Autonomous Initial Coin Offering (DAICO)

DAICO is a new fundraising model. Founder of Ethereum blockchain Vitalik Buterin proposed this model, combining the advantages of Decentralized Autonomous Organizations (DAO) with the classic ICO. This synergistic model allows you to make the process of collecting and spending funds as **transparent** and **safe** as possible.

DAICO is **based on a smart contract that regulates all actions** to attract and work with funds.

The difference between DAICO and ICO begins after the first stage when a mechanism called a **tap** is launched. Tap allows tokens holders to **control how much money is available to the team**. The **crane** determines the **amount per second that the development team can withdraw** from the contract. Payments to developers are made not once, but **gradually**, for example, once a month. If they need a larger amount than the one that is written in the smart contract, then this question is put to the **vote**. And holders of tokens can either approve this proposal or not.

2.20.3 Initial Exchange Offerings (IEO)

The cryptocurrency exchange is directly involved in the selection of projects, the organization of the tokensale and the sale of tokens.

There are several advantages of IEO which are in overcoming the disadvantages of ICO:

- **The risk of Scams for investors is lower**. The project is launched at the exchange after the serious procedure of verification. The exchange rejects the doubtful project in order to save its' reputation.
- The process of **listing new tokens is faster**.
- **Redistribution of costs becomes available**. According to Autonomous Research, there is $1-3 M cost to list the token at the exchange. An IEO project has lower costs for listing.
- The **speed of funding is higher**. In ICO the primary distribution of tokens may been lasted for several days whereas in IEO it lasts **several minutes** or even seconds.

- Investors' gain is higher. The listing **token values are bigger** than in primary distribution.
- There is **no need to open another wallet**.
- The process of investing is simple, the investors need to replenish the balance on the exchange, wait for the token-sale and put an order for buying.
- **Tokens are traded at an equal price**. This reduces the probability of falling rates from the early investors who purchased first.

At the same all these advantages have some drawbacks. For example, due to the high distribution speed, **some investors have no time to make an order and buy tokens of big projects**. Moreover, nowadays there is limited number of IEO, and they are not mainstream way of funding the projects. Reason for lower popularity is the **unwillingness of exchanges to take additional work**. The mentioned verification procedure is very strict, e.g. there is an obligation for **verification of identity**.

Example. BitTorrent token (BTT) on Binance exchange[14] and LEO token on the Bitfinex exchange.

The evident advantage of IEO over an ICO is the presence of existing user base on the exchange platform that allows to raise tremendous investments even on the private sale stage.

2.20.4 Security Token Offerings (STO)

The popular utility tokens used in ICO have the main disadvantage, which is the **absence of any compensation for investors in case of ICO fail** as utility tokens are not security papers leading to the absence of any obligations for making beneficial conditions for investors. **Security tokens represent real capital** in the enterprise, at the same time such token is **not necessarily tied to a share in the company**, it can be used to **divide the rights of the ownership**. In fact, they may **give the holder a number of rights**: ownership of shares, periodic dividends, cashflows, payment of debts, the right to vote, etc. All these rights are secured by a smart contract. Due to the features of these tokens, their value is supported by securities, therefore, they are considered an investment. The issue of security tokens requires **serious supervision by regulatory authorities**. This supervision leads to the protection of investments and gives investors more rights, thus restoring the balance of power from the point of view of stakeholders. **Security token offering (STO)** is the initial offer of security tokens. STO is similar to ICO in one thing: both issues token for investors. The main reason for buying securities token is **dividends or voting rights**. STO project meets all the requirements of the SEC: investor's money is protected by law. In case of disputes, the investor can file a complaint with the appropriate authority since this type of token falls under the legislation on securities.

By 20th March 2019, 122 STOs have already been completed, raising $512M. 54 Security Token Offerings are currently listed and ongoing. Out of 328 STOs launched so far, only 12 of them have failed (3.65%). STO raised $1258M in overall.

Examples. Bolton Coin, UniCrypy, GG World Lottery.

Criteria	ICO	DAICO	IEO	STO
Definition	Crowdfunding by issuing utility token/coin	A synergy of DAO and ICO makes the ICO more secure as investors' funds are available in a more controlled manner	A modified version of ICO. Crowdfunding by issuing utility token/coin by cryptocurrency exchange without first ICO step	Type of digital "securities papers" comparable with IPO
First starting date	Early 2013	Mid 2018	Early 2017	End 2017, evolution to ETO[38] at end 2018
Fundraising is conducted at	The token issuer's website	Similar to ICO	The platform of the exchange	Token security platform
Crowdsale counterparty	The project's development team	The project's development team and mechanism for an investor to control	The cryprocurrency exchange	Security token issuer brokered through STO platform
Smart contract managed by	The startup conducting the token sale	Similar to ICO	The cryprocurrency exchange	Token security platform
AML/KYC need by the token issuer	Yes, it can vary between the different projects. Each investor has to go through a dedicated KYC/AML vetting process, accommodated for by the project	Each investor has to go through a dedicated KYC/AML vetting process, accommodated for by the project	Not necessarily -the exchange conducts AML/ KYC on its users. KYC/AML is done on the exchange, so existing exchange account holders do not have to go through it again.	KYC/AML is done on the platform, so existing platform account holders do not have to go through it again.
Marketing	Marketing budget needed for funding companies is significantly high, the project will have to invest many resources in order to get the attention of the public	Similar to ICO	A token issuer can tap on the exchange' reach and users. Joint marketing with exchange	The token issuer has to market to individuals' investors. Though the platform might be able to provide extra marketing.
Screening required before a startup can launch a crowdsale	No – anyone can launch an ICO (in a country where it is legal)	Similar to ICO	Yes – the exchange screens the company before it allows it to raise funds on its platform	Probable none
Automatic token listing after crowdsale	No – the startup has to reach out to exchange to list its tokens. Not immediate. Listing fees to be negotiated with each exchange.	Similar to ICO	Yes – the exchange where the IEO is conducted list. The exchange listing is immediate. Included in IEO fees.	Depends. If the selected STO platform also is an exchange, then it is likely immediate.
Difficulty to set up	Easy	Easy	Medium	High
Fundraising cost	Low	Low	Medium	Medium
Investor protection	Low (limited)	Medium	Limited-Medium	Strong
Investor accessibility	High	High	Low-Medium	Low
Regulation level	Low (but the is an improvement)	Low	Low-Medium	High
Governance level	Low	Medium-High	Medium	High
Liquidity	Medium	Low-Medium	High	Low
Centralization level	Relatively centralized	Relatively decentralized: decisions on funds are decided by a voting system, thus are democratized. In principle, there is not a centralized team that takes all the decisions.	Relatively centralized	Relatively centralized

[14] https://icoholder.com/en/bittorrent-28385

2.21 Stablecoins

A stablecoin is a new class of cryptocurrencies that attempts to offer **price stability** and are **backed by a reserve asset**.

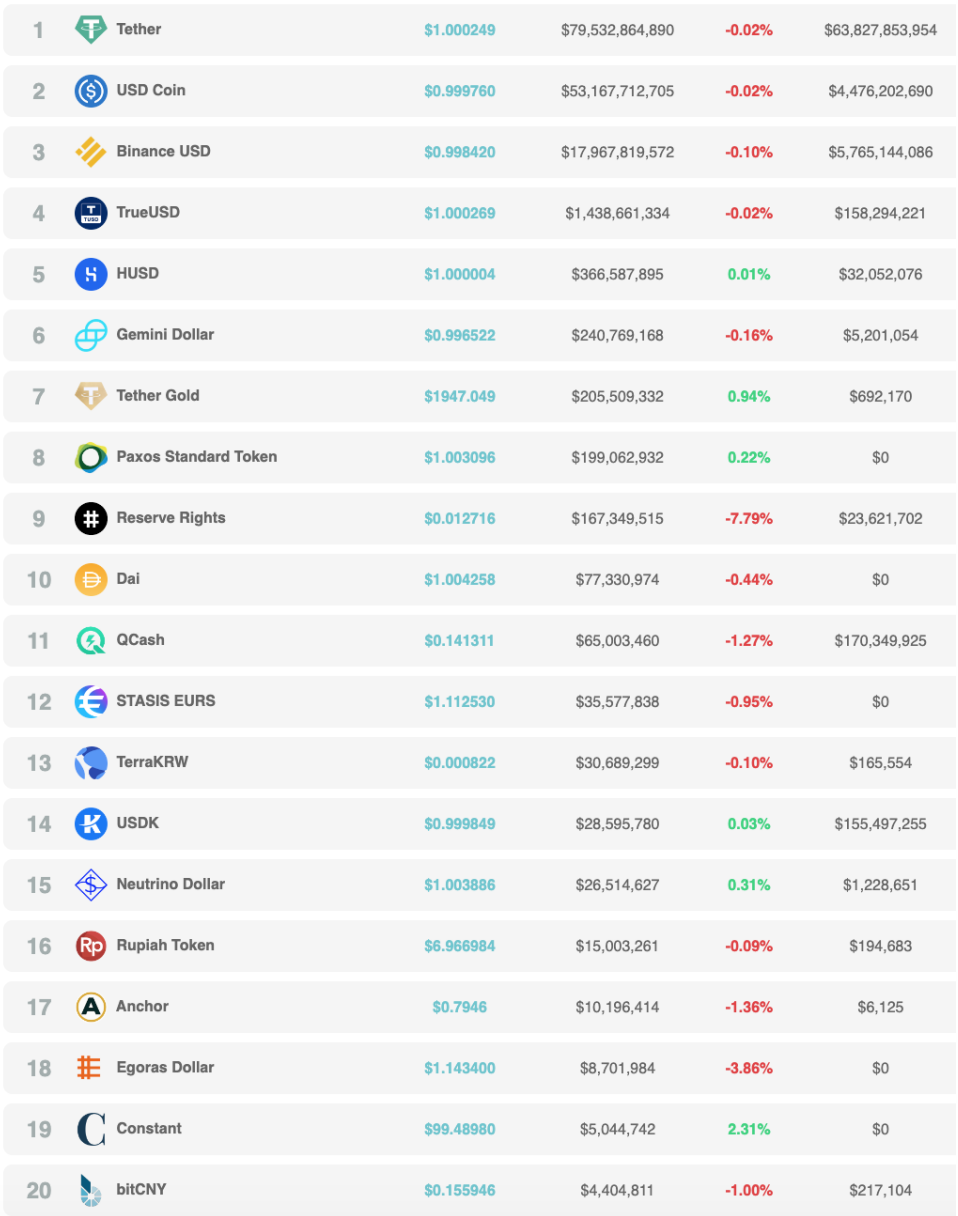
Stablecoins may be pegged to a currency like the USD or to a commodity's price such as gold.

Stablecoins achieve their price stability via collateralization (backing) or through algorithmic mechanisms of buying and selling the reference asset or its derivatives.

Stablecoins have gained traction as they attempt to offer the best of both world's-the instant processing and security or privacy of payments of cryptocurrencies, and the volatility-free stable valuations of fiat currencies.

Example. The Reserve protocol comprises two tokens: the Reserve token (RSV – a decentralized stablecoin) and the Reserve Rights token (RSR – a cryptocurrency used to facilitate the stability of the Reserve token and confers the cryptographic right to purchase excess Reserve tokens as the network grows).

Example. bitCNY Pegged to the Chinese Yuan



1	Tether	$1.000249	$79,532,864,890	-0.02%	$63,827,853,954
2	USD Coin	$0.999760	$53,167,712,705	-0.02%	$4,476,202,690
3	Binance USD	$0.998420	$17,967,819,572	-0.10%	$5,765,144,086
4	TrueUSD	$1.000269	$1,438,661,334	-0.02%	$158,294,221
5	HUSD	$1.000004	$366,587,895	0.01%	$32,052,076
6	Gemini Dollar	$0.996522	$240,769,168	-0.16%	$5,201,054
7	Tether Gold	$1947.049	$205,509,332	0.94%	$692,170
8	Paxos Standard Token	$1.003096	$199,062,932	0.22%	$0
9	Reserve Rights	$0.012716	$167,349,515	-7.79%	$23,621,702
10	Dai	$1.004258	$77,330,974	-0.44%	$0
11	QCash	$0.141311	$65,003,460	-1.27%	$170,349,925
12	STASIS EURS	$1.112530	$35,577,838	-0.95%	$0
13	TerraKRW	$0.000822	$30,689,299	-0.10%	$165,554
14	USDK	$0.999849	$28,595,780	0.03%	$155,497,255
15	Neutrino Dollar	$1.003886	$26,514,627	0.31%	$1,228,651
16	Rupiah Token	$6.966984	$15,003,261	-0.09%	$194,683
17	Anchor	$0.7946	$10,196,414	-1.36%	$6,125
18	Egoras Dollar	$1.143400	$8,701,984	-3.86%	$0
19	Constant	$99.48980	$5,044,742	2.31%	$0
20	bitCNY	$0.155946	$4,404,811	-1.00%	$217,104

Gold-Backed Stablecoins Usually one token of stablecoin equals one gram of the gold. Since they're tied to the gold, this stablecoin's price can't fall below the current price of the gold. A third party holds the gold in reserve.

LIBRA Morgan Beller planted the first seeds of Libra,[15] she is known as the key figure behind Facebook's push into blockchain technology; she also considers herself a co-creator of Libra.[16]

- May 2017: Morgan Beller left her job at Medium.com and started working at Facebook. Nine months later, she joined the blockchain strategy team and started working on Facebook blockchain initiative.
- 8 May 2018: David Marcus (Facebook vice president) joined the new Libra team and officially began working on the new project. They recruited other top Facebook talent onto the new blockchain division.
- 13 June 2019: The Wall Street Journal reported that Facebook is working on its own cryptocurrency that will be announced next week, and set to be launched next year alongside the blockchain-based network that will support it. The Wall Street Journal also reported that Facebook has secured backing from more than a dozen companies like PayPal, Visa, Mastercard, Stripe, Booking.com, and each company will invest around $10M to fund the new currency development.
- 18 June 2019: Facebook announced Libra digital currency, Calibra digital wallet, and the nonprofit Libra Association, an organization based in Geneva, Switzerland, that will govern the new currency.
- 16 July 2019: David Marcus told the Senate Banking Committee in U.S. Senate Hearing, "that Facebook will only build its own Calibra cryptocurrency wallet into Messenger and WhatsApp, and will refuse to embed competing wallets" and also expressed that "Libra will not launch until the U.S. lawmakers' concerns have been answered".
- 4 October 2019: PayPal withdraws from the Libra Association.
- 9 October 2019: U.S. Senator Sherrod Brown and Senator Brian Schatz sent letters to the CEOs of Stripe, Visa, and MasterCard to express deep concerns over Facebook's Libra Association
- 11 October 2019: eBay, Mastercard, Strip, Visa, and Mercado Pago withdraw from the Libra Association.
- 15 October 2019: Facebook launches Libra Association. Representatives from the remaining 21 organizations (out of 28 original members) met and signed Libra Association charter in Geneva. 21 initial members include: Coinbase, Lyft, Uber Technologies, Spotify AB, PayU, Vodafone, Women's World Banking, Mercy Corps, Creative Destruction Lab any other organizations

Tether [17]

Abstract. A digital token backed by fiat currency provides individuals and organizations with a robust and decentralized method of exchanging value while using a familiar accounting unit. The innovation of blockchains is an auditable and cryptographically secured global ledger. Asset-backed token issuers and other market participants can take advantage of blockchain technology, along with embedded consensus systems, to transact in familiar, less volatile currencies and assets. In order to maintain accountability and to ensure stability in exchange price, we propose a method to **maintain a one-to-one reserve ratio between a cryptocurrency token, called tethers, and its associated real-world asset, fiat currency**. This method uses the Bitcoin blockchain, Proof of Reserves, and other audit methods to prove that issued tokens are fully backed and reserved at all times.

Beginning with a whitepaper published online in January 2012, J.R. Willett described the possibility of building new currencies on top of the Bitcoin Protocol. The precursor to Tether, originally named "Realcoin", was announced in July 2014. The first tokens were issued on 6 October 2014, on the Bitcoin blockchain. On 20 November 2014, Tether CEO Reeve Collins announced the project was being renamed to "Tether". In January 2015, the cryptocurrency exchange Bitfinex enabled trading of Tether on their platform. In early 2018 Tether accounted for about 10% of the trading volume of Bitcoin, but during the summer of 2018 it accounted for up to 80% of Bitcoin volume. It formerly claimed that each token was backed by one United States dollar, but on 14 March 2019 changed the backing to include loans to

[15] `https://sls.gmu.edu/pfrt/wp-content/uploads/sites/54/2020/02/LibraWhitePaper_en_US-Rev0723.pdf`
[16] `https://www.publish0x.com/crypto-timelines/a-brief-timeline-of-libra-xqqdww`
[17] `https://tether.to/en/`

affiliate companies On 30 April 2019 Tether Limited's lawyer claimed that each Tether was backed by only $0.74 in cash and cash equivalent.

MakerDAO [18]

It's the protocol behind the stable coin Dai – a cryptocurrency that maintains a 1:1 peg to the USD, being backed by collateral (Ether to be specific).

Let's say you're an Ether holder and you would like to create Dai. Your first move would be to send your Ether to a **Collateralized Debt Position (CDP)**. A CDP is a type of software that runs on the blockchain, in this case the Ethereum blockchain, and lives within the Maker ecosystem.

Once your Ether is in the CDP smart contract, you are able to create Dai. The amount of Dai you can create is relative to how much Ether you have put into the CDP. This ratio is fixed, but can be changed over time. The amount of Dai you can create relative to the Ether you put in is called the **collateralization ratio**.

Assume 1 Ether = $100 and collateralization ratio is 1.5. If you send 1 Ether into the CDP smart contract, then you are now able to create 66 Dai, i.e. 100/1.5.

- **what happens when Ether goes up?**
  The system becomes over collateralized and Dai becomes stronger. Maker has mechanisms that incentivize users to create more Dai if the price of Dai should trade above one dollar.
- **what happens when Ether goes down?**
  If Ether goes down, now that can cause problems.
  If the value of Ether held as collateral is worth less than the amount of Dai it's supposed to be backing, then Dai would not be worth one dollar and the system could collapse.
  Maker combats this by liquidating CDPs and auctioning off the Ether inside before the value of the Ether is less than the amount of Dai it is backing.
  Basically, if the price feed into the CDP indicates that the value of Ether has gone below a certain threshold, the liquidation ratio (let's use 125% of created Dai), then the CDP is "liquidated" and the Ether inside the CDP is auctioned off for Dai until there is enough Dai to pay back what was extracted from the CDP.

Example. Two simple examples assuming the price of Ether is $150 and you deposit 1 ETH at this price. Liquidation ratio is 1.5.

- You decide to take out 50 Dai which means your CDP is **collateralised** 300% (collateralization ratio = 3). As long as the price of Ether doesn't drop below ($50 · 1.5) = $75 your position will be safe. After one year, you decide to pay back the 50 Dai and retrieve your Ether locked up. Upon closing the position or other interactions with your CDP, you'll pay the annual stability fee (set at 3.5% as of March 2019).
- You decide to take out 95 Dai which means your CDP is collateralised at just 150.26% (collateralization ratio = 1.5026, very close to liquidation ratio). The price of Ether drops to $100 which means your CDP is **under-collateralized**: $95 · 1.5 = $142.50 > $100.
  A 3rd party will realise that you don't have enough collateral and liquidate your CDP on your behalf. This results in your position being liquidated by 3 rd parties with a penalty. These 3rd parties have various ways to profit from your position being liquidated.

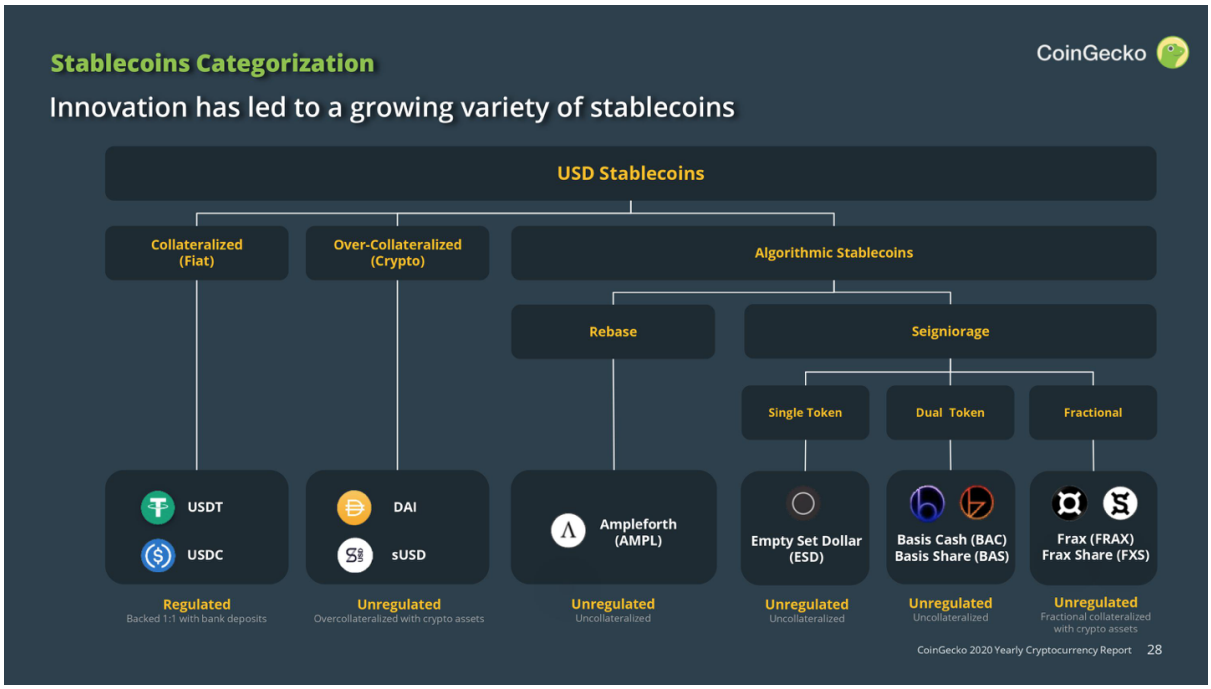
Volatility cannot be destroyed, it can only be transferred. If we have a stable token like Dai that has been stripped of its volatility, where did it go? In the Maker system, volatility is transferred entirely to the holder of the CDP. Using our prior example, should I withdraw 66 Dai from a CDP containing one Ether, I will only own that one Ether if its price is above the liquidation ratio. Dai is effectively a loan on my Ether.

This leads us to the interesting consequence: I can take the Dai that I borrowed and use it to buy more Ether. By doing this, I am basically buying Ether on margin. That's right, completely decentralized leverage!

MakerDAO is like a credit facility that issues loans with a certain interest rate. If the interest rate (stability fee) is low, people are encouraged to borrow more (lock up more ETH). If the interest rate is high, the cost of capital is high making it less attractive to borrow (close out CDPs).

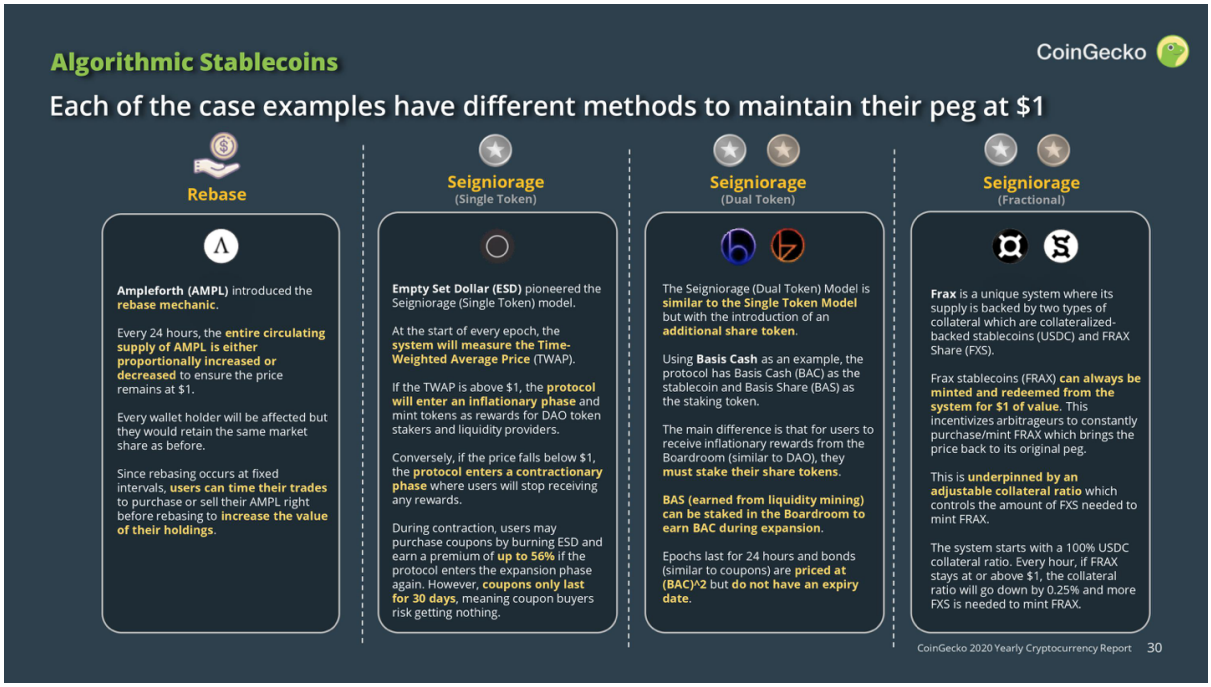
[18] `https://makerdao.com/`

Tether is pegged to an asset, MakerDAO is tied to the amount you put in a CDP and can cause problems because Ether is much more volatile.



Algorithmic Stablecoins `https://www.youtube.com/watch?v=S7-rfvpEpJs`

Frax: `https://docs.frax.finance/price-stability`



2.22 Cryptocurrencies and Asset Allocation

Some meaningful questions:

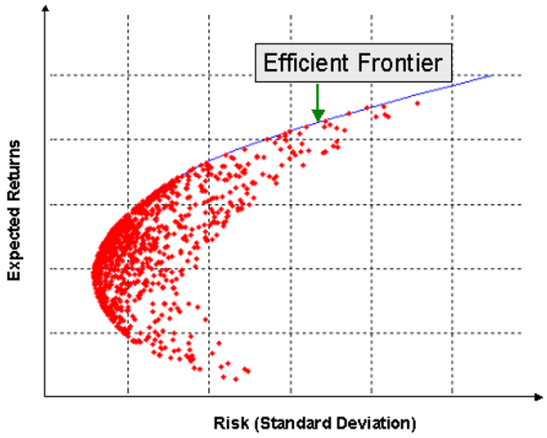
- Are cryptocurrencies **expected returns higher** than (classical) financial assets?
- Are cryptocurrencies **riskier** than (classical) financial assets?
- What about diversification?

Let $\pi = (w_1, \dots, w_N)$ be a portfolio, where w_i is the amount of asset i, and is positive if bought, negative if short.

Mean-Variance Analysis It's the process of weigthing risk, expressed as variance, against expected return, **applied to a portfolio, not a single asset**.

Investors use mean-variance analysis to make decisions about which financial instruments to invest in, based on how much risk they are willing to take on in exchange for different levels of reward.

The **efficient frontier** is the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return.



N assets: $R = (\bar{r}_1, \dots, \bar{r}_N)$ is the vector of expected return. V is the covariance matrix.

$$\bar{R} = \pi R \qquad \bar{\sigma}^2 = \pi V \pi^{\mathrm{T}}$$

are the expected return and variance of the portfolio.

Our market for the first (toy) problem

- Bitcoin
- Ether
- Amazon
- Apple
- Facebook
- General Electric
- General Motor

Daily datas (only working day) from September 2017 to February 2020

$$\begin{bmatrix} \text{return} \\ \text{variance} \end{bmatrix} = \begin{bmatrix} 1.0026 & 1.0017 & 1.0014 & 1.0012 & 1.0005 & 0.9993 & 1.0000 \\ 0.24 & 0.36 & 0.03 & 0.03 & 0.04 & 0.06 & 0.03 \end{bmatrix}$$

$$\text{correlation} = \begin{bmatrix} 1.0000 & 0.6975 & 0.0149 & 0.0188 & 0.0350 & 0.0160 & -0.0093 \\ 0.6975 & 1.0000 & 0.0546 & 0.0614 & 0.0441 & 0.0403 & 0.0561 \\ 0.0149 & 0.0546 & 1.0000 & 0.5807 & 0.5623 & 0.2261 & 0.3016 \\ 0.0188 & 0.0614 & 0.5807 & 1.0000 & 0.4706 & 0.2844 & 0.3454 \\ 0.0350 & 0.0441 & 0.5623 & 0.4706 & 1.0000 & 0.2132 & 0.3292 \\ 0.0160 & 0.0403 & 0.2261 & 0.2844 & 0.2132 & 1.0000 & 0.2809 \\ -0.0093 & 0.0561 & 0.3016 & 0.3454 & 0.3292 & 0.2809 & 1.0000 \end{bmatrix}$$

Portfolio 1

$$\pi = \begin{bmatrix} 0.9573 & -0.2611 & 2.8507 & 2.7995 & -1.4778 & -2.1661 & -1.7026 \end{bmatrix} \qquad \bar{R} = 1.01$$

Portfolio 2

$$\pi = \begin{bmatrix} 0.1037 & -0.0223 & 0.2308 & 0.3644 & 0.0439 & 0.0252 & 0.2543 \end{bmatrix} \qquad \bar{R} = 1.001$$

Portfolio 3

$$\pi = \begin{bmatrix} 0.0184 & 0.0015 & -0.0312 & 0.1209 & 0.1961 & 0.2444 & 0.4500 \end{bmatrix} \qquad \bar{R} = 1.0001$$

Suppose to invest \$100. After one year:

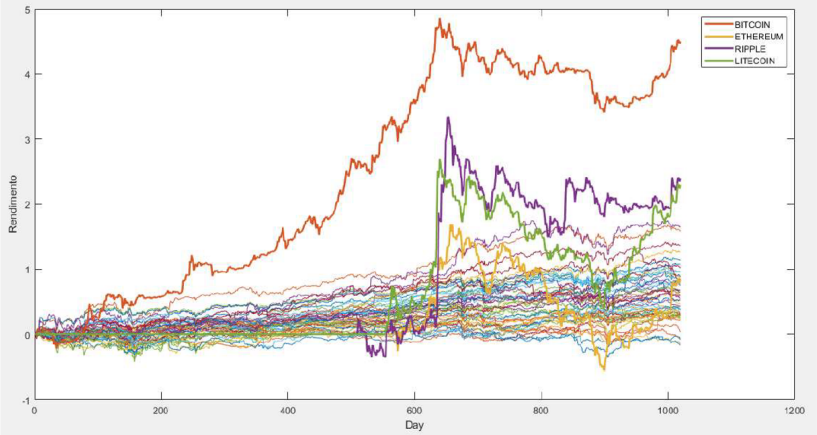
- Portfolio 1: $100 \cdot 1.01^{365} = 3778$
- Portfolio 2: $100 \cdot 1.001^{365} = 144$
- Portfolio 3: $100 \cdot 1.0001^{365} = 103$

The best reason to invest in cryptocurrency seems **diversification**.

Our market for the second (toy) problem

- First 50 stocks for capitalization (S&P50)
- 4 cryptocurrencies: Bitcoin, Ethereum, Ripple, Litecoin

Period: July 2015-May 2019



Cryptocurrencies are very uncorrelated with standard financial assets, as can be proved using statistics. However, they are quite correlated among them.

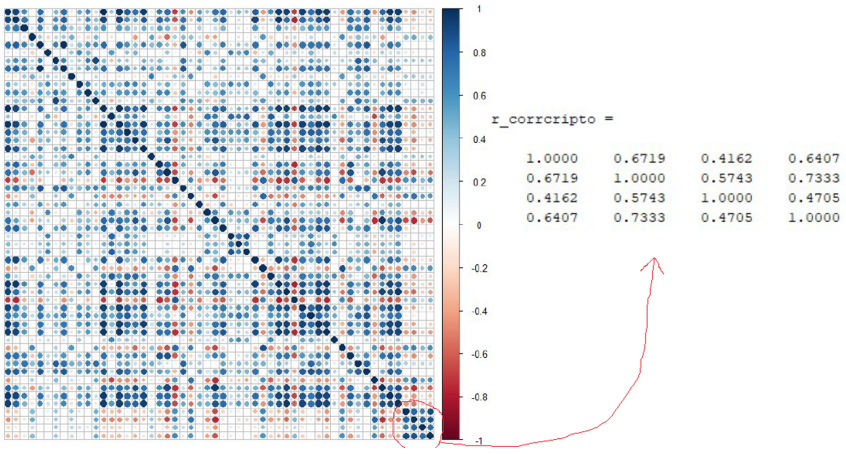


Figure 2.5: Correlation.

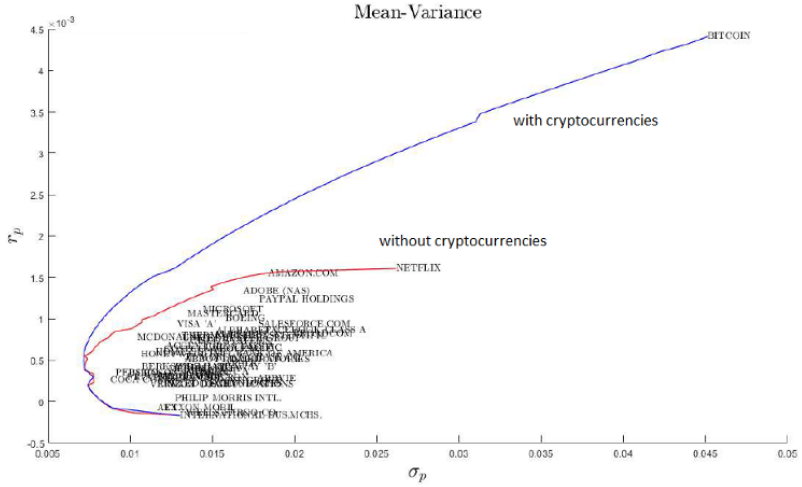


Figure 2.6: Mean-Variance Analysis

2.23 Central Bank Digital Currency (CBDC)

CBDC are digital tokens, similar to cryptocurrency, issued by a central bank. They are pegged to the value of that country's fiat currency.

Central bank money handled through electronic means and accessible to the broad public. A CBDC is issued and regulated by a nation's monetary authority or central bank and as such it's a central bank liability.

As a centralized form of currency, they may not anonymize transactions as some cryptocurrencies do.

CBDC is a third form of base money, next to deposits and banknotes.

The four key properties of Money:

- Issuer: central bank or not
- Form: digital or physical
- Accessibility: widely (retail) or restricted (wholesale)
- Technology: token – or account – based. Token-based retail CBDCs are accessible with private/public keys. This method of validation allows users to execute transactions anonymously. Account-based retail CBDCs require digital identification to access an account.

Cash features

- Inclusive: easy to use and available to everybody without electronic devices.
- Crisis proof: device independent (power blackout). Peer to peer transactions: there is no intermediation of a thrid party to complete a transaction.
- Anonymous.
- Off-line transactions.

Digital means of payments do not satisfy these features. Digital means require third parties (record keeping, identity validation). It is difficult to deploy off-line transactions.

Today: cash is the only form of central bank issued money available to individuals and part of money (bank deposits) is managed by profit oriented players. CBDC: universal/inclusive, digital, central bank issued (legal tender[19]).

State of the art Many central banks are still considering multiple options. 5 projects focus on a direct CBDC, 2 on an indirect CBDC, and 10 investigate several designs or do not specify the architecture.

Only one project focuses on conventional (database) technology whereas 5 focus on DLT. The problem of DLT is the inadequate performance and scalability, offering no advantages in a centralised system or cash-like resilience in case of prolonged outages.

Three projects focus on token-based access and three on account based access.

No CBDC project has an explicit focus on payments beyond the central bank's jurisdiction. Several central banks are working on cross-border payment trials in parallel to their CBDC efforts.

Motivation The Consumer perspective A digital currency like CBDC is an additional form of public money and means of payment. The currency should be cheap to use, secure, risk-free (its holders should not be subject to any market risk or issuer default risk), easy to use and efficient (permitting fast payments).

The possibility of cash to be used without any technical infrastructure is not (fully) matched by electronic payment solutions. As such, the CBDC should allow citizens to continue to make their payments as much as they do today with cash, allowing off-line payments, free of charge for basic users and privacy friendly.

The pandemic and resulting economic crisis have led to marked changes in the use of cash and in retail payment behaviour:

- temporary decline in cash withdrawals
- higher cash holdings

[19] Legal tender is anything recognized by law as a means to settle a public or private debt or meet a financial obligation, including tax payments, contracts, and legal fines or damages. Fiat currency of a country is a legal tender.

- increased use of contactless payments and surge in e-commerce

In a crisis period households without access to digital payment means could face barriers to making and receiving payments.

This has led to an acceleration in the reduction of ATM and physical bank branches, possibly impacting groups of people who do not have access to digital payment mechanism (seniors, migrants, ...).

Moreover, 1 billion people worldwide do not have basic identification (ID) credentials, and many more have IDs that cannot be trusted because are of poor quality or cannot be verified. 3.4 billion people have some form of ID but with limited ability to use it in the digital world. Lack of ID makes it difficult for citizens to access financial services, for financial service providers to on-board customers and for governments to efficiently transfer funds to the rightful beneficiary.

General purpose CBDCs could enhance inclusion in the medium term if inclusion features prominently in CBDC designs. In emerging markets general purpose CBDCs could improve financial inclusion by providing a low-cost means of payment and acting as an on-ramp to broader financial services and advanced economies central banks are considering these CBDCs in light of declining cash use and potential reduced access to cash.

Trade-offs between token and account based CBDCs from a consumer perspective:

- Account-based CBDCs require individuals to present valid ID, linking personal ID to a CBDC could generate data privacy concerns. Identification is not strictly needed for a token-based CBDC.
- Ownership of a token-based CBDC is linked to "knowledge" such as a digital signature. This design could improve access to financial services, substitute for cash in countries where access to cash is cumbersome, enable a degree of anonymity and reduce the need for financial accounts. Yet a token-based CBDC without formal identification presents integrity risks and would have limited usefulness for G2P payments due to opportunities for fraud.

The Central Bank perspective

The introduction of a CBDC might reinforce the transmission of monetary policy by allowing the central bank to set the remuneration rate on the digital currency in order to directly influence the consumption and investment choices of the non-financial sector, leading to monetary stability.

Cash and CBDC, if widely available and transacted via resilient channels that are separate from those of other payment services, could constitute a possible contingency mechanism for electronic retail payments that could remain in use even when private solutions are not available (because of outages of private card payment schemes, online banking and ATMs).

The real economy perspective

CBDC can encourage digital innovation by:

- filling the gap in the provision of digital payments
- enhancing financial inclusion (user solutions accessible to customers)
- preserve independence in retal payments

Foreign central banks are assessing the possibility of issuing their own CBDC and private actors, including large technology firms, are developing payment solutions that could threaten financial, economic and political sovereignty. Wide acceptance of a means of payment not denominated in sovereign currency could weaken the transmission of monetary policy, affect financial stability, safety and efficiency of the payment system.

CBDC may be the solution to these threats if it offers functionalities that are at least as attractive as those of the payment solutions available in foreign currencies or through unregulated entities.

A way to go (probably) The development of CBDC should be focused on creating the basis for these functionalities:

- Cash-like payment system
- Convenience to use
- Resilience
- Universal access-privacy

Consumers are unlikely to adopt a CBDC if it is less convenient to use than today's electronic payments/cash: key features are cash-like peer-to-peer usability, convenient real-time payments, privacy and wide accessibility.

Banks and payment service providers run sophisticated infrastructures that can handle peak demand and help to smooth the flow of payments by taking on risk, for example during connectivity breaks or offline payments: a CBDC must be secure not only from the insolvency or technical glitches of intermediaries, but also from outages at the central bank.

From a privacy perspective, there is an underlying trade-off between privacy and ease of access on the one hand and ease of law enforcement on the other. CBDC is tied to an identity system (account-based technology) or instead via cryptographic schemes (digital tokens).

Technology and architecture There are three main possible architectures, defining the role of the central bank in day-to-day payment and the structure of legal claims:

- Indirect CBDC (two-tier CBDC) model: the consumer has a claim on an intermediary, with the central bank keeping track only of wholesale accounts. The indirect CBDC implies loads similar to those of today's system.
- Direct CBDC model: the CBDC represents a direct claim on the central bank, which keeps a record of all balances and updates it with every transaction. The direct CBDC would require massive technological capabilities, as the central bank processes all transactions by itself, handling a volume of payments traffic comparable with that of today's credit or debit card operators.
- Hybrid CBDC model: an intermediate solution providing for direct claims on the central bank while allowing intermediaries to handle payments. The hybrid CBDC architecture is more complex to operate than the indirect model, as the central bank does maintain retail balances. It could be implemented at scale using today's technology.

  It might offer better resilience than the indirect CBDC, but at the cost of a more complex to operate infrastructure for the central bank. The hybrid CBDC is simpler to operate than a direct CBDC. As the central bank does not directly interact with retail users, it can concentrate on a limited number of core processes, while intermediaries handle other services including instant payment confirmation.

The infrastructure could be based on a conventional centrally controlled database, or on a distributed ledger. In conventional databases, resilience is typically achieved by storing data over multiple physical nodes, which are controlled by one authoritative entity. In DLT-based systems, the ledger is jointly managed by different entities in a decentralized manner and without such a top node. The overhead needed to operate a consensus mechanism is the main reason why DLTs have lower transaction throughput than conventional architectures.

Current DLT could not be used for the direct CBDC except in very small jurisdictions. DLT could be used for the indirect CBDC architecture, as the number of transactions in many wholesale payment systems is comparable with that handled by existing blockchain platforms.

Resilience: neither a DLT-based system nor a conventional one has a clear-cut advantage. The key vulnerability of a conventional architecture is the failure of the top node, for example via a targeted hacking attack. The key vulnerability of DLT is the consensus mechanism, which may be put under pressure, for example, by a denial-of-service type of attack.

Two future scenarios

1. CBDC replaces the use of banknotes by households.
   The balance sheet of banks and central bank would change but the effect on the economy is minor.
2. CBDC replaces bank deposits.
   Higher funding costs, larger balance sheet of the Central Bank, collateral scarcity, disintermediation.
   In the extreme (full substitution): banks do not create money, monetary policy rate would be given by refinancing conditions of the banking system but by the interest rate paid on CBDC.

Central Bank

The central bank would expand its role in the economy and its risk exposure buying assets (loans and securities) against CBDC. Illiquid assets, ultimately taking on more credit and market risk. A CBDC

would affect profitability and risk exposure of the Central Bank. The Central Bank is exposed to financial liabilities as an operator of a retail payment system. For example, malfunctioning of the IT infrastructure underlying the CBDC could cause loss and damage to individual users, raising questions about the responsibility of the central bank.

Financial stability

Bank runs in crisis periods: when savers have less confidence in the whole banking sector, liquid assets might be shifted very rapidly from commercial bank deposits to the digital euro if the operational obstacles to withdrawing money in the form of digital euro are lower than for withdrawing cash.

The role of banks as intermeidaries is affected. Financial inclusion is affected in case CBDC will substitute effectively cash. Possibility to introduce limits and "intermediation" for retail users.

Monetary policy

Investors may substitute safe assets (for example, sovereign bonds) with the digital euro, which would directly affect risk-free interest rates.

A CBDC may be remunerated for monetary policy reasons, but also for financial stability and structural reasons, such as to lower demand for CBDC for investment purposes and to prevent large inflows. When considering the features that would make the CBDC competitive relative to alternative digital payment instruments, its competitive advantages should be considered. it is not the aim of the central bank to compete with commercial banks for financial stability reasons and given their important role in monetary policy transmission.

Chapter 3

Blockchain (N. C. Fabrizio)

3.1 Intro to DTL

Bitcoin is the mother of all the blockchains. It's constructed around the idea of having a **ledger** where we track information. Typically there is a **central** authority which holds the only copy of it and is the only one allowed to modify it.

A distributed ledger is a **tamperproof** sequence of data that can be **read** and **augmented** by **everyone**. Nodes don't need to **trust** each other, the mechanism of **consensus** (a combination of game theory and probability) removes the need for it, and everyone has an interest in behaving as the blockchain requires.

DLT: which desirable properties?

- Immutability
- Accessibility
- Decentralisation

3.2 Bitcoin Protocol

Blockchain: a data structure to record information incrementally (a history). An append-only data structure, a growing chain of blocks, with immutable history: what is written cannot be altered.

It's maintained by a **peer-to-peer** open network of nodes: anyone can join! Identical replicated copies of the ledger/blockchain, **openly available**: anyone can read, **true by majority**.

PoW slows and controls the creation of new blocks, in this way you can't change the blockchain previously and have the time to re-calculate all hashes, because the chain will grow and the longest chain always wins (except with hard-forks).

No one is in charge, but the whole community.

Problem: how does the community select someone in charge to write the next block?

Who writes the next sentence?

In our case, whoever writes the next sentence gets 100 000 EUR!

Who writes the next sentence?

Proof of Work! This is the solution adopted by the Bitcoin blockchain: by controlling the difficulty of the challenge (against the available computational power in the network), it is possible to select, with a reasonable probability, a leader in charge of the next sentence/block. Note that the challenge is difficult to solve, i.e. find 431 and 433, but easy to validate, i.e. multiply them and check the result: difficult to be elected, straightforward to be recognized as a leader.

Bitcoin mining protocol [Nakamoto08] Each node of the Bitcoin network will

- choose and verify pending transactions
- solve the crypto-puzzle (depending on the previous block)

If solved/leader then

- create the next block and
- broadcast the new block

Everyone

- validate new block (Proof of Work + transactions) and embed it in the local copy of the blockchain
- start mining the next block

The blockchain is actually a tree, including *the* chain.

Currently, a block is about 1 MB and is added every 10 minutes (for an average of 3-7 trans/sec).

If a node wants to change an old block, or in general wants to make the chain take a direction he wants, he would need half as much the computational power of the network to keep up with the others. This is considered to be impossible, and everyone has an advantage in sticking with the same version.

3.3 Security aspects

3.3.1 Hash functions

Hashes. Are functions: $H(a) = h$ such that:

- from a to h is easy, but from h to a computationally infeasible (you should try every possible input);
- a can be of any length (thus could even be a document) but h is fixed by the specific hash function chosen.

Properties:

- **Collision resistance**, it's infeasible to find a and b such that $a \neq b$ and $H(a) = H(b)$
- **Hiding**, when a secret value r is chosen from a probability distribution that has high entropy, then given $H(r \mid x)$ it is infeasible to find x. "|" means concatenation of two strings.
- **Puzzle friendliness**, given h and a random k, it's unfeasible to find x such that $h = H(k \mid x)$.

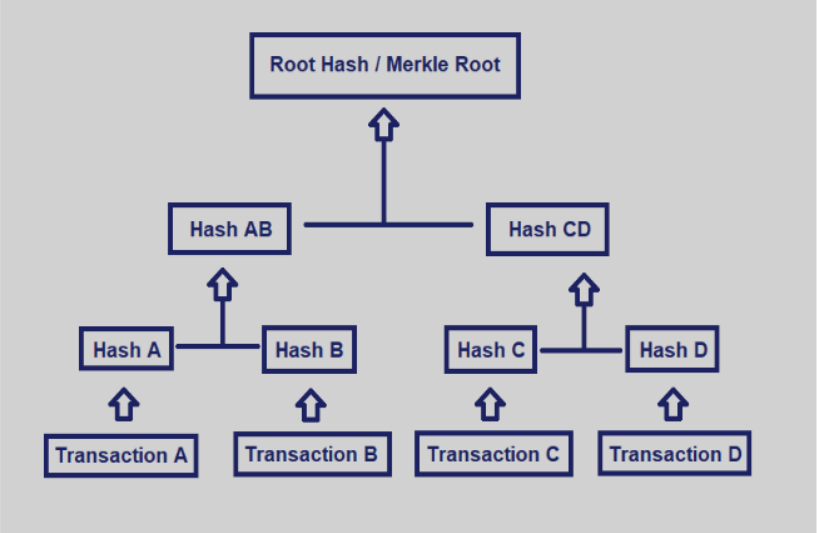
3.3.2 Merkle tree

Some desirable properties of a (new) block

- follows linearly (and exclusively) the previous one – links to the previous block
- the set of transactions is fixed before leader election – Merkle tree. against corruption
- the leader is identified – as the solver of the puzzle

Merkle tree. Is a digest of digests, i.e. a tree, a compact representation of a set of hashes of Bitcoin transactions, amongst the proposed and selected ones. The root represents all the hashes in the tree and any change anywhere would affect the root.

- Transactions are hashed two by two until the root, which is like the DNA of the block.
- In other words, a Merkle tree summarizes all the transactions in a block by producing a digital fingerprint of the entire set of transactions, thereby enabling a user to verify whether or not a transaction is included in a block.
- It is possible to **verify that a transaction belongs to a block without verifying all the blocks.**
- Thanks to this structure, it is quicker and more efficient to verify the consistency and content of the data



3.3.3 Proof of Work

Proof of Work (at a certain degree of approximation) is finding `nonce` (number only used once) such that

$$H(H(\text{previous block}), M(\text{transactions} + \text{revenue transaction}), \texttt{nonce})$$

is an hash starting with C zeros. M is the Merkle Root of the input transaction and the reward transaction. C controls the complexity and adapts regularly. There are some technical details involved, like the fact the timestamps are also hashed, everything has to be converted to hexadecimal and the hash is made two times.

PoW is very expensive in terms of energy consumption.

Miners have interest in solving this hard problem because they are rewarded:

- the fees from each transaction;
- an amount of BTC generated (mined), in other words, everyone suddenly agrees that such miner has that quantity in addition to their wallet. The quantity was initially 50 BTC and is halved every 210 000 blocks; currently, it's 6.25 BTC. This mechanisms guarantees that in circulation there will never be more than

$$210\,000 \cdot (50 + 25 + 12.5 + 6.25 + \cdots) = 21\,000\,000 \text{ BTC}$$

3.3.4 Public and private keys

Asymmetric[1] cryptography is based on two keys associated with an identity:

- a private (secret) key sk, known only by the owner. It is generally used to encrypt a message by the owner, $sk\{m\}$
- a public key pk associated to the identity and public. It can decrypt a message encrypted with sk, i.e. $pk\{sk\{m\}\} = m$ and also $sk\{pk\{m]\} = m$.

Properties

- one way function: unfeasible from $sk\{m\}$ to get m without pk, easy with pk.
- with n nodes, only n pairs (pk_i,sk_i) are needed instead of n^2
- anyone can secretly send to j by using $pk_j\{m\}$
- j can prove that they own a piece of information by using $sk_j\{m\}$

Signature of a message (of a transaction) – on top of asymmetric cryptography:

- $sig := sign(sk, \text{message})$

[1] *asymmetric* means that the two keys are different.

- $isValid := verify(pk, \text{message}, sig)$

Using the public key pk one can validate the author of a message (transaction!)

How is asymmetric cryptography is used in Bitcoin?

You generate your own private key from a very large set, from there you derive your public key, and from the public key, an address can be derived by hashing.

Bitcoin transactions move value from multiple input addresses to multiple output addresses.

3.3.5 Encryption with Elliptic Curves

In algebraic geometric, an elliptic curves (EC) over $\mathbb{R}$ is defined by the (Weirstrass's equation): $y^2 = x^3 + ax + b$. The curve is non singular if its determinant is non zero, which means $4a^3 + 27b^2 \neq 0$.

Bitcoin uses EC for public key, it uses a famous EC **secp256k1**:

$$y^2 = x^3 + 7$$

`https://youtu.be/muIv8I6v1aE`

`https://youtu.be/qCafMW4OG7s`

Bitcoin uses elliptic curves for public keys.

3.4 Structure of a Block

A block has two main components:

- **list of transactions**;
- **block header**, which contains:
  - `Version`: Block version number
  - `Previous Block hash`: This is used to compute a new block hash. Hence, making the blockchain temper-proof.[2]
  - `Merkle Tree Root`: The root of the Merkle tree is a verification of all the transactions.
  - `Timestamp`: The time at which block is mined.
  - `Bits/Difficulty`: Difficulty in Bitcoin is expressed by the hash of a Bitcoin block header being required to be numerically lower than a certain target.
  - `Nonce`: A 32-bit random number used in blockchain, while calculating the cryptographic hash for a Block.

3.5 Public, Hybrid and Private DLT

The ledger, replicated on each node, may be **public**, **private** or **hybrid**.

The network maybe **permissionless** or **permissioned**.

Public

- No one is in charge
- Anyone can participate
- Open and transparent
- Decentralized consensus mechanisms (PoW, PoS, ...)

[2] The first block doesn't have any previous one, it's called the **genesis block**.

- Mining
- Slow

Hybrid

- Multiple selected organizations
- Permissioned known identities
- Pre-approved participants
- Voting/multi- party consensus
- Lighter and faster

Private

- Private property of an
- individual or an organization
- Permissioned known identities
- Pre-approved participants
- Voting/multi- party consensus
- Lighter and faster

			READ	WRITE	COMMIT	EXAMPLE
BLOCKCHAIN TYPES	Open	Public permission less	Open to anyone	Anyone	Anyone	Bitcoin. Ethereum
		Public permissioned	Open to anyone	Authorized participants	All or subset of authorized participants	Supply chain ledger for retail brand viewable by public
	Closed	Consortium	Restricted to an authorized set of participants	Authorized participants	All or subset of authorized participants	Multiple banks operating a shared ledger
		Private permissioned "enterprise"	Fully private or restricted to a limited set of authorized nodes	Network operator only	Network operator only	External bank ledger shared between parent company and subsidiaries

Source: OECD (2018c)

In general, every blockchain has its token.

3.6 Ethereum

Ethereum is an **open-source** and public blockchain-based **distributed computing platform for building decentralized applications**. Vitalik Buterin envisioned Ethereum as a platform for developers to write programs on the blockchain. In short, Ethereum wants to be a **World Computer** that would decentralize – and some would argue, democratize – the existing client-server model. With Ethereum, servers and clouds are replaced by thousands of so-called "nodes" run by volunteers from across the globe (thus forming a "world computer").

Smart contracts are simple programs stored on the blockchain that automatically exchange money based on certain conditions.

A contract that self-executes, and handles the enforcement, the management, performance, & payment. You would require tokens for executing a smart contract as well as for trading. So basically, Ethereum is incomplete without cryptocurrency. The language is *Turing-complete*, meaning it supports a broader set of computational instructions.

Smart contracts can:

- Function as *multi-signature* accounts, so that funds are spent only when a required percentage of

people agree

- Manage agreements between users, say if one buys insurance from the other
- Provide utility to other contracts (similar to how a software library works)
- Store information about an application, such as domain registration information or membership records.

Ethereum runs on its native token (**Ether**) which serves two main purposes:

- Ether payment is **required for applications to perform any operation** so that broken and malicious programs are kept under control.
- Ether is **rewarded as an incentive to the miners who contribute to the Ethereum network** with their resources – much like Bitcoin's structure.

Every time a contract is executed, Ethereum consumes a token which is termed as **gas** to run the computations. Gas is required to be paid for every operation performed on the Ethereum blockchain. Its price is expressed in Ether and it's decided by the miners, which can refuse to process the transaction with less than a certain gas price.

Examples[3]

- A smart contract could be programmed to release funds for someone's birthday each year. It could also be programmed to release payment once someone confirms receipt of delivered goods. It could be used to enforce particular rights for holders of digital assets.
- A simple example could be in the case of **life insurance**. The policy terms would be encoded into the smart contract. In the event of a passing, the notarized death certificate would be provided as the input trigger for the smart contract to release the payment to the named beneficiaries.

Oracle A blockchain oracle is a third-party service that connects smart contracts with the outside world, primarily to feed information in from the world, but also the reverse. Information from the world encapsulates multiple sources so that decentralised knowledge is obtained. Information to the world includes making payments and notifying parties. The oracle is the layer that queries, verifies, and authenticates external data sources, usually via trusted APIs, proprietary corporate data feeds and internet of things feeds and then relays that information.

3.7 Ethereum vs. Bitcoin

While both the Bitcoin and Ethereum networks are powered by the principle of distributed ledgers and cryptography, the two differ technically in many ways. For example, **transactions on the Ethereum network may contain executable code, while data affixed to Bitcoin network transactions are generally only for keeping notes.**

Both Bitcoin and Ethereum currently use a consensus protocol called Proof of Work (PoW), which allows the nodes of the respective networks to agree on the state of all information recorded on their blockchains and prevent certain types of economic attacks on the networks. In 2022, **Ethereum will be moving to a different system called Proof of Stake (PoS)** as part of its Eth2 upgrade, a set of interconnected upgrades that will make Ethereum more scalable, secure, and **sustainable**.

Proof of Stake substitutes computational power with staking-making it less energy-intensive-and replaces miners with validators, who stake their cryptocurrency holdings to activate the ability to create new blocks.

While Bitcoin was created as an alternative to national currencies and thus aspires to be a medium of exchange and a store of value, **Ethereum was intended as a platform to facilitate immutable, programmatic contracts and applications via its own currency**.

3.8 Ethereum moving to Proof of Stake

Ethereum has announced the launch of 2.0 and the process of moving to Proof

[3] `https://medium.com/coreledger/what-are-smart-contracts-a-breakdown-for-beginners-92ac68ebdbeb`

of Stake (may be completed and happen soon)

- This approach does not require large expenditures on computing and energy, it will change the need for CPUs for mining
- Miners are now "validators" and post a deposit in an escrow account
- The more escrow you post, the higher the probability you will be chosen to nominate the next block
- If you nominate a block with invalid transactions, you lose your escrow
- One issue with this approach is that those that have the most Ethereum will be able to get even more
- This leads to centralization eventually
- On the other hand, it reduces the chance of a 50%+1 attack and allows for near-instant transaction approvals
- The protocol is called Casper and this will be a hard fork

3.9 Dapps

- **Decentralized Applications (Dapps)** are computer applications that operate over a blockchain enabling direct interaction between end-users and providers.
- **The interface of the decentralized applications does not look any different than any website or mobile app today.**
- The **smart contract represents the core logic** of a decentralized application. Smart contracts are integral building blocks of blockchains, that process information from external sensors or events and help the blockchain manage the state of all network actors.

3.10 ERC20

- First token standard
- Introduced in November 2015 as an Ethereum Request for Comments (ERC)
- Automatically assigned GitHub issue number 20, giving rise to the name "ERC20"
- A standard for fungible tokens, meaning that different units of an ERC20 token are interchangeable and have no unique properties
- The ERC20 protocol standard contains basic functions that any useful token should implement to enable trading. These include transferring tokens, inquiring about the balance of tokens at a certain address, and the total supply of tokens.
- The ERC-20 standard defines the interfaces for a few common methods: i.e. totalSupply, balanceOf, transfer, transferFrom, and approve. These methods allow Ethereum smart contracts to issue fungible tokens and token holders to transfer tokens to one another.[4]
- Today, there are thousands and thousands of ERC20 tokens (derivatives of ETH), almost in every sector and use case, not only in finance (for a list see Etherscan and select ERC20)

[4] `https://eips.ethereum.org/erc`