Robin.

By Dr. Alexey Osipov,

Senior quantitative mathematician,

Alexey.Osipov@sidenis.com, osipovav28@googlemail.com

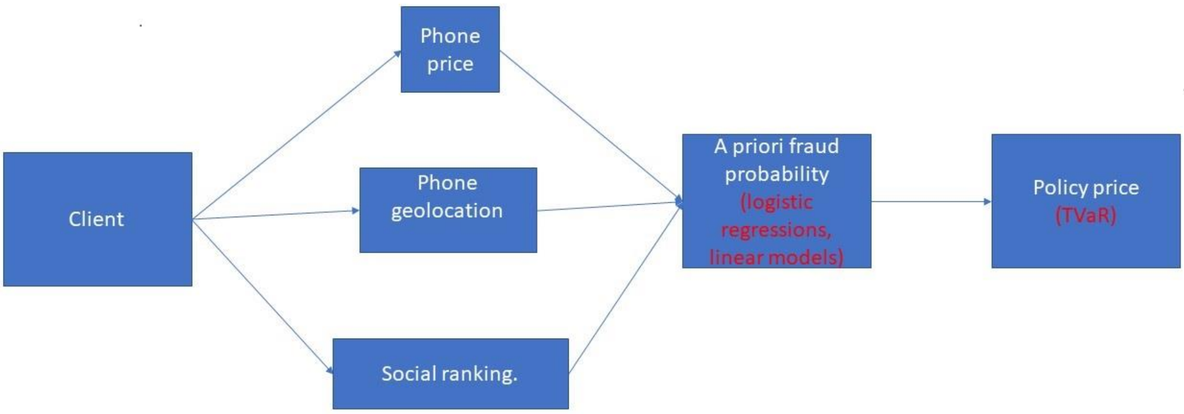
Sidenis.

The text consists of 2 parts: First, we describe everything from a business point of view, and then we give all mathematical details, which may be of interest only to specialists.

Business point of view.

Determine the insurance policy price.

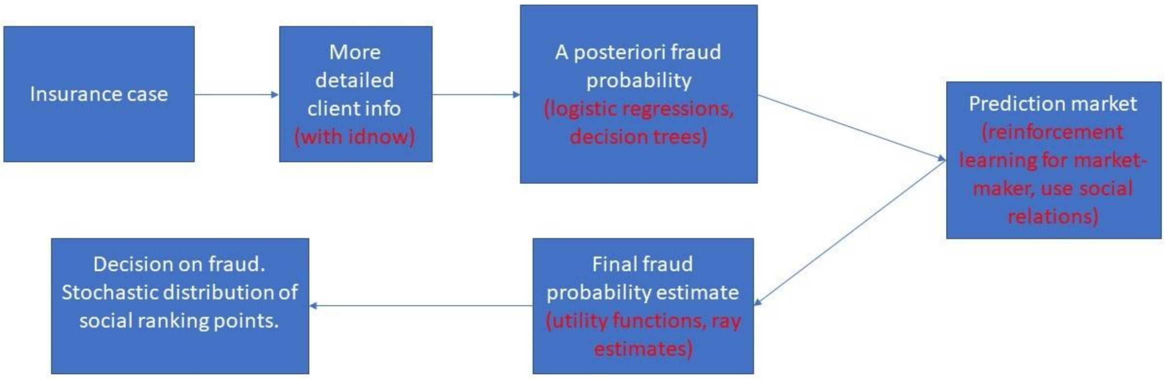
We follow the logic described in the scheme:



Assume that a client comes to us and wants to insure a phone. Our first step is to get data on phone prices in the region from websites of some large shops (like **Ulmart, Amazon, eBay**), and to determine the normalized price of the client's phone relative to this distribution of region phone prices. Afterwards we use **Model A** that determines fraud probability by the normalized phone price, adjust this fraud probability with **Model F** taking into account the geographic location (a country in which the phone was at the time of signing an insurance contract identified by geographic coordinates), and use **Model B** that determines the insurance policy price by fraud probability, client's insurance history, the client's **social ranking** in our system, and contract length.

Estimate fraud probability from data.

Now assume that the client comes to us, reports that the insurance case has occurred, and demands money. Our general logic follows the scheme:



We ask him/her to provide us some personal information using https://www.idnow.io/**:**

1) Name

We use it to check the validity of the client's name in our system.

2) Surname

We use it to check the validity of the client's surname in our system.

3) Age

4) Phone number registered under the client's name

It is an optional field. We do not check the registration ourselves, but we assume that the client believes that we do check.
*It can be also asked as an extra question in **idnow.***

5) A relaxation question like:

Do you have pets?
(as an extra question in **idnow**).
We do not need this information, we just want to ask an innocent question here in order to help the client to relax and answer other questions.

6) Employment information (as an extra question in **idnow**)

How long have you been working for your current company?

A) unemployed
B) ... < 1 year
C) 1 year <= … < 4 years
D) 4 years <= … < 7 years
E) …>= 7 years

We also determine the following information from our system:

1) Client's guarantor status:
A) standard client
B) in top 10% of our social ranking
C) currently is a guarantor (*it means that the client has decided to give us some part of his/her **social ranking** if the insurance case with the phone of another client occurs; it is possible to be a guarantor only for another person*).

2) Number of contracts with us.

3) Number of guarantors for the client.

We prefer not to use this information at the very beginning, when the client comes to us for the insurance, partially since we do not want to lose the client and partially since this information can change fast, and only the latest information is important for us.

Then we apply **Model C** to the collected data (together with the normalized phone price and contract length) and refine the fraud probability estimated by **Model A.**

Refine fraud probability using prediction markets.

After we have obtained fraud probability estimate p, we tell the client that we need 24 hours and refine the estimate using the prediction markets.

The idea of prediction markets is that people vote with their money: they bet virtual proxy of money (social ranking in our case) and help to determine the event probability by trying to make profit. This concept is used in many areas varying from medicine to politics.

We make a prediction market for the client's case. Any other client in our system can gamble on this market. Hereinafter we call such client a **gambler**.

The prediction market for fraud detection can be best described by this table:

Number of voices	Ray	Price	Ray	Number of voices
50	<= 0.8	1.25|5		
31	<= 0.7	1.43|3.33		
85	<= 0.65	1.54|2.85		
		1.76|2.32	>= 0.57	78
		2|2	>= 0.5	96
		2.49|1.67	>= 0.4	32
		5|1.25	>= 0.2	14

The table demonstrates an **order book.** Each row corresponds to certain probability q. The first and the fifth columns show number of voices (or number of social ranking points) for fraud probability contained in the ray <= q or >= q respectively. The second and the fourth columns show the direction of the order either: <= q or >= q respectively.
For example, in the table there are 50 voices for fraud probability <= 0.8, and there are 31 voices for fraud probability <= 0.7, and, finally in the table there are 50 voices for fraud probability >= 0.2. The third column shows 2 prices: for fraud probability contained in <= q or >= q respectively. The prices are inverse to the probabilities (either $1-q$ or q), and the social ranking points are redistributed according to them.

More detailed description of the prediction market:

1) A gambler sees the description of the fraud case: client's name, client's surname, his story describing what has happened with the phone.

2) A gambler sees not the whole order book, but only 2 lines in the middle of the table corresponding to **the best prices** for fraud and against fraud respectively. This is done, in order to reduce the possibility of speculation. The **best prices** are the two highest prices for <= and >= respectively. In the table they are: 1.54 and 2.85 for <=, and 1.76 and 2.32 for >=.

3) A gambler has an option of making an **order** of size b either claiming that the fraud probability is >= q, or claiming that it is <= q. This order is either registered in the order book or becomes a deal.

4) An order of size b claiming that the fraud probability is >= q by definition means that the corresponding price is $c_{for} = \frac{1}{q}$, and the gambler receives $b(c_{for} - 1)$ if the fraud is declared, but loses b if the fraud is not declared.
Contrary, an order of size b claiming that the fraud probability is <= q by definition means that the corresponding price is $c_{against} = \frac{1}{1-q}$, and the gambler receives $b(c_{against} - 1)$ if the fraud is not declared, but loses b if the fraud is declared.
Note that by construction:

$$\frac{1}{c_{for}} + \frac{1}{c_{against}} = 1.$$

In the table demonstrated above in the notation 1.76|2.32 the first price corresponds to betting on <= q and the second corresponds to betting on >= q.

Note that the higher the price c is (no matter whether it is for fraud or not), the better it is for the gambler who has placed the order. But the higher the price c is, the lower is its dual price $\frac{1}{1-\frac{1}{c}}$, so an order with very high price c is unlikely to be matched (since, the dual price is low). Thus, there should be a balance between 2 prices, which will allow to estimate the fraud probability.

5) Note that not every order can be registered in the order book. Some orders become deals instantly. Let c_{for}^{best} and $c_{against}^{best}$ be 2 best prices corresponding to the rays >= q_{for}^{best} and <= $q_{against}^{best}$. Then an order for the ray >= q can be registered in the order book only if

$$q \geq q_{against}^{best},$$

If not, the order is shifted to the order >= $q_{against}^{best}$, and matched with the already registered order <= $q_{against}^{best}$ (the order price is recalculated accordingly).

Similarly, an order for the ray <= q can be registered in the order book only if

$$q \leq q_{for}^{best},$$

If not, the order is shifted to the order <= q_{for}^{best}, and matched with the already registered order >= q_{for}^{best} (the order price is recalculated accordingly).

The matched orders become deals and are deleted from the order book.
This reasoning can be best demonstrated by a simple example given in the picture below:

Suppose that we have two best prices corresponding to orders with probabilities >= 0.57 and <= 0.65 in our order log. This two orders cannot make a deal taken together, since there is no contradiction between them. But if a gambler tries to register an order corresponding to >= 0.7, then this order will be in contradiction with the order <= 0.65. Then we just assume that the gambler tries to register an order corresponding to >= 0.65 and match it with the order corresponding to <= 0.65. It is a standard way for orders to become a deal.

6) Most deals are made with us. We are a **market-maker**, a liquidity provider, at each time moment we have two orders in the order book of the **market-maker volume**. If this condition does not hold, it is a reason to decrease spread size of the market-maker (parameter ρ defined below).

7) The **market-maker Model D** defines two **current prices** or **quotes**: a current price for the fraud $c_{for} = c$ and a current price against the fraud $c_{against} = \frac{1-\rho}{1-\frac{1}{c}}$, where ρ is the parameter responsible for spread size. This parameter is used to try to prevent gamblers from earning by arbitrage. Its value is determined by **Model D.**
If ρ is large, gamblers will refuse to play; if it is small, they may arbitrage on current prices movements (making a bet for, observing the price change, and then making a bet against). This arbitrage deals only spoil the fraud probability estimation.

8) The **current prices or quotes** (and, hence, the corresponding orders of the market maker) start from ones corresponding to a priori fraud probability and change basing on behavior of

all gamblers as proposed in **Model D**. This may lead to execution of orders.
Note that **Model D** is based on reinforcement learning and is inspired by
N.T. Chan & C. Shelton, An Electronic Market-Maker, MIT 2001.

9) The market-maker current prices and the best prices are common for all gamblers.

10) A gambler receives the reward only at the end of the voting (decision on payment is stochastic based on decisions of all gamblers).

11) A gambler sees all his/her deals and all his/her orders but does not have any access to deals or orders of other gamblers.

12) There is a chat for gamblers to discuss some event details.

13) Neither an order, nor a deal can be cancelled. The only exception are the orders of the market-maker (since it is a liquidity provider). This condition is required for stability of the fraud probability estimate.

14) After the voting the fraud probability is estimated by **Model E.**
In order to take into account social relations between the client and other gamblers, we suggest estimating the fraud probability both using all existing orders and using only the orders of the client's friends (if the number of such orders is enough for a good estimate); the resulting fraud probability estimate is the maximum of 2 fraud probabilities. The logic behind this is the following: if all your friends think that it is a fraud, but neutral people do not think so, then it is most probably a fraud.

15) In **Model E** market maker deals are also taken into account. The market maker is treated as a standard user, but with the maximal possible social ranking.

16) In order to determine whether it is a fraud, we consider the corresponding Bernoulli random variable with the fraud probability (or just toss a biased coin) and decide whether it is a fraud depending on the random variable outcome.
That is, if the fraud probability is estimated as 70%, we toss a coin with a probability of head 70%. In case of head, we state that it is a fraud and make payments with social ranking points in accordance with this decision, in case of tail, we state that it is not a fraud and make payments accordingly.
The reason behind this requirement is to prevent a collusion between gamblers. Also this condition is reasonable since we will never know the right answer anyway.

17) Note that **Model D (the market-maker model)** and **Model E (fraud probability estimation in prediction markets)** are independent of each other and based on completely different principles. **Model D** is completely agnostic of **Model E** actions, and **Model E** considers the market-maker orders produces by **Model D** as produced by one of the users.

18) If less than 3 people participated in the voting, we do not apply Model E, we just use the fraud probability estimate from the previous step. In this case all gamblers receive their social ranking points back.

19) If less than 3 client's friends participated in the voting, we do not estimate the fraud probability using their votes, we just apply **Model E.**

20) It is possible to implement several prediction markets for one event (thus, splitting users into groups or strats), in order to prevent the collusion, and then to average the resulting probabilities, but we prefer not to do it (at first) for simplicity.

21) **Model E** returns not only the fraud probability estimate, but also its variance. So theoretically it is possible to make confidence intervals for the fraud probability basing on it. But we choose not to do it, mostly for simplicity, and for the ease of distribution of social ranking points.

Note that **Model E** is based on the work of Dr. Nikolai Osipov from Saint-Petersburg Department of Russian Academy of Sciences. It is a simplified version, and currently he is working on much more sophisticated models of prediction markets (focusing on applications in medicine). In the beginning of his research on prediction markets, we were working together. Any extra funding of his research is highly appreciated.

Decision and consequences.

If we failed to estimate the fraud probability during the voting, we just use the pre-voting estimate of the fraud probability. We state that the case is a fraud, if the pre-voting probability estimate is larger than 0.5.
If we managed to estimate the fraud probability during the voting, we toss a coin as described above, and conclude if it is a fraud or not. Social ranking points are distributed according to this decision.

If we have finally decided that the case is a fraud, we ask for a police certificate, and make payments depending on whether it is provided. No matter whether the police certificate is provided, we assume that the case is a fraud, decrease the client's social ranking, and next time the client gets worse insurance conditions (according to **Model B**).

Social ranking.

A client receives 1000 points for obtaining insurance.

For being a guarantor, if an insurance case does not occur, a client gets 100 points, if an insurance case occurs a client loses 300 points.

If the final model based on prediction markets detects a fraud, a client loses 3000 points.

Note that the social ranking cannot be less than 0.

Exchange of social ranking is not allowed.

It is possible to increase or decrease social ranking by gambling on prediction markets, or by being a **guarantor**.

Note that there can be an inflation or deflation of social ranking points. It is difficult to predict it in a situation of a cold start. That is why it is reasonable to adjust the social ranking distribution policy in future.

Mathematical point of view.

The data.

It seems impossible to find open datasets that fit ideally with the problem under the consideration. We used the following datasets:

1) Dataset 1. Synthetic Financial Dataset for Fraud prediction:
https://www.kaggle.com/ntnu-testimon/paysim1
It contains synthetic data on fraud in mobile financial transactions.
We mostly use it to predict fraud probability using the (normalized) transaction amount.

2) Dataset 2. Credit Card Fraud Detection:
https://www.kaggle.com/mlg-ulb/creditcardfraud
It contains fraud data on fraud in credit card transactions.
We use it solely to predict fraud probability using the (normalized) transaction amount.

3) Dataset 3. Statlog (German Credit Data) Data Set:
https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29
It contains data on credit rating (either good or bad) of various clients. The dataset contains many features that are of interest to us.
We use it to predict fraud probability using various features.
We also use this dataset to bridge fraud probability in Germany with Germany crime ranking from Dataset 4.

4) Dataset 4. Movehub City Rankings:
https://www.kaggle.com/blitzr/movehub-city-rankings/home
It contains city rankings by various criteria. One of the criteria is a crime ranking. We use this crime ranking in **Model F**, in order to take into account the geographic location.

5) Dataset 5. Crimes in India:
https://www.kaggle.com/rajanand/crime-in-india
It contains statistical information about different crimes in India. We use it in order to estimate the fraud probability in India and to link it with India crime ranking from Dataset 4.

We understand that none of this data sets fits ideally with fraud prediction in phone insurance. But we believe that main psychological moments should remain the same. That is why we try to make our models as simple as possible.
We believe that **simplicity means robustness**.

The first 3 datasets have a column named **amount** (similar with the phone price in our case). But the distributions of amount are different, since the datasets origin from similar problems but in different areas. That is why we take logarithm of amount and normalize it. This way we get numbers on similar scale.

Note that for Dataset 1 we leave only transactions of type **‘CASH_OUT’**, since they fit well with our problem.

Model A. Or predicting fraud probability basing on normalized amount.

The idea is to fit a logistic regression on each of the three datasets, and then to blend all 3 logistic regressions together.

In order to check the model, we use a train-test split with ratio 75%-25%.

Here is the information on the model quality:

	Average precision score.	Accuracy.	Brier score.
Dataset 1	0.065	99.83%	0.012
Dataset 2	0.0017	99.82%	0.012
Dataset 3	0.288	71.2%	0.238

Note that Datasets 1 and 2 are highly imbalanced, that is why we get such values of statistics. The statistics table given above does not allow to state good model quality. But we believe that, since our model is blending of 3 simple models fitted on 3 different datasets, it will be sufficiently robust and will show a reasonable quality on real data.

Remark on neural networks.

It is possible to try to apply a sophisticated non-interpretable model in our situation. The first candidate that comes into our mind is a neural network model. We could split the normalized phone price into a moderate number of bins, and consider the problem of classification using neural networks (we could derive many features from the phone price by making dummy variables correspondent to normalized phone price splitting into many cells). However, we chose not to do it partially due to lack of time, partially since it is possible to implement this concept into many ways, and partially because we wanted to have an interpretable model that is as simple as possible. But we could make some experiments in future if there is an opportunity, and check possibility of their application.

Model B. Or determining the insurance price.

Denote by p the probability obtained by **Model A.**

Now we use weights in the following way:
1) If a client had our insurance K times in a row, there were no insurance events, then p is multiplied by $\mathbf{0.84^{K}}$ (where $\mathbf{0.84}$ is a global constant identified during the analysis of Dataset 1: the probability of fraud decreases in average **by 16%** if a client comes for the second time).
2) After an insurance event K is set to 0.
3) If we detect fraud for a client K times in a row, then p is multiplied by $\mathbf{1.16^{K}}$ (the same **16%** from analysis of Dataset 1)**.**

Let q be the new fraud probability that takes into account the weights (of course, it cannot be larger than 1).

Consider a random variable: it takes value 0 with probability $1-q$**,** and it takes value V with probability q (V is the phone price)**.** We approximate this Bernoulli random variable with a mixed random variable that either takes value 0 or is uniform in the range from $\frac{V}{k}$ to V with mixing coefficient λ**.** We have 2 parameters: k and λ, we can estimate them by the method of moments for the first 2 moments of the original Bernoulli variable.

Now we calculate TVaR (**tail value at risk** for level 99%), multiply it by $\frac{T}{T_0}$ (where T is the contract term that we are going to offer, and T_0 is our standard contract term), and take into account the loadings, in order to get the final price.

We offer to decrease the final price by 10% for being in top 10% of the social ranking (if the previous insurance case was not a fraud).

Model C. Or predicting fraud probability basing on more detailed personal data.

In order to construct this model, we use the data from Dataset 3. We use only certain columns from the dataset. In the table below we explain how the columns from the dataset fit with our problem:

Column from Dataset 3	In our case
Standardized duration	Standardized duration of the contract
Credit history	Insurance history
Employment	Employment
Guarantors	Guarantors
Age	Age
Number of credits in bank	Number of insurances with us
Number of people being liable	Number of guarantors for the client
Normalized log amount	Normalized log price
Telephone	Is a registered on the client's name telephone indicated?

Note that **Credit history/Insurance history** is an important column for our models. But clearly it will not be available in the situation of cold start, when, essentially, we have no insurance history. That is why we have constructed 2 models: the first for the standard situation and the second for the cold start.

We standardize **duration** column in the dataset by dividing by median. We use a train-test split with ratio 75%-25%. For the train set we use the standard 3-fold cross-validation. As for the metrics, we optimize mostly Brier score, but also look at accuracy and average precision score.

Since our problem differs significantly from the problem of Dataset 3, we restrict ourselves to the simplest machine learning models. We have constructed 2 models: one for the standard situation and one for the cold start. Each of this two models is a blending of a decision tree and a logistic regression.

We indicate the importance weights for the decision tree models in the table below:

Column name	Standard situation	Cold start
Standardized duration of the contract	29%	43%
Insurance history	28%	0%
Employment	0%	9%
Guarantors	0%	0%
Age	14%	13%
Number of insurances with us	0%	0%
Number of guarantors for the client	10%	9%
Normalized log price	19%	26%
Is a registered on the client's name telephone indicated?	0%	0%

Note that even columns with importance weights zero are used in the logistic regression model.

In the table below we give statistics for our models:

	Average precision score.	Accuracy.	Brier score.
Standard situation	0.309	71.6%	0.19
Cold start	0.327	72.4%	0.19

The statistics from the table above does not allow to boast good model quality. However, once again, since the data on which we are going to apply the models differ from the data of Dataset 3, we tried to make our models as simple as possible believing that **simplicity implies robustness**.

Model D. The market maker behavior.

Assume that we are interested in estimating probabilities with absolute accuracy of 0.1%. It means that we have a discrete set of probabilities that induces a discrete set of prices (by the formula $c = \frac{1-p}{p}$). A market maker is a logic of changing the current prices (or quotes) by taking into account the deals. Usually a market maker does not take into account orders. One of the characterizations of a market maker is speed, or the number of social ranking points required for moving it by 1 point in the discrete set of prices. It makes sense to choose speed such that no user alone could substantially move the market maker. Since a

user receives 1000 points for insurance, we can assume that a market maker should move by probability of 5% for every 1000 points, or by probability of 0.1% for every 20 points. Parameter ρ is chosen so that a gambler cannot shift probability by more than ρ using a regular bet size. It also can be viewed as precision with which we try to estimate the fraud probability. We suggest to set it to 0.5%, assuming that a regular bet size is 50 points. We set the market make volume to 10000 points. This number should be adjusted in future, and it is difficult to choose it in a situation of a cold start.
It is also possible to connect the market maker with the maximal social ranking (or some quantile of the social ranking), but we prefer not to do it (at least in the situation of **cold start**) mostly for simplicity.

We suggest to improve this naive market maker described above basing on reinforcement learning principles. For simplicity the market maker has a constant spread (defined above). Our marker maker has only one **state variable**, called **order imbalance**. We define it as the **accumulated market maker position** since the last change of current prices (or quotes), i.e.

$$OIMB = \left\lfloor \frac{Volume_{for} - Volume_{against}}{20} \right\rfloor,$$

where $Volume_{for}$ and $Volume_{against}$ are the volumes of deals with the market maker for fraud and against fraud respectively. Depending on the **state** the market maker takes **an action** with certain probability**,** i.e. either moves current prices by 1 point above or below on our discrete scale of prices, or does not move at all. A **policy** is a triple of probabilities for a random variable $p(a|s)$. For example, we have described above a naive market maker that moves completely deterministically basing on an order imbalance. This is an initial policy.

We represent a policy in a Boltzmann sense:

$$p_i = \frac{\exp(w_i)}{\sum \exp(w_j)}.$$

Thus, each policy is uniquely characterized by weights w_i (condition on the state).

An **event** is a voting round that takes 24 hours. After each round the market maker policy p_0 is updated using all information from last 100 rounds. We define a **reward** in a posteriori way as a delta of profit mathematical expectation using the fraud probability q estimated by **Model E** at the end of the round.

The update of the policy is done as follows. Using the data collected from the last 100 rounds and the Monte Carlo method, we estimate $Er(a|s)$, then we choose an optimal action a for each state s, turn it into a policy $p_*(a|s)$. Finally, we adjust the initial policy p_0 towards the optimal policy p_* increasing or decreasing weights by 1 in the Boltzmann formula.

Model E. Estimating fraud probability using prediction markets.

For fraud probability estimation, both orders and deals are taken into account. Any order (or deal) of size s (measured in units of **social ranking**) is interpreted as s voices of so-called **unit experts**.

We assume that each **unit expert** has an estimate of the event probability that is a random variable from $N_{[0,1]}(\mu, \sigma)$, where μ is the true fraud probability (that we want to estimate), σ is standard deviation, and $N_{[0,1]}$ means that the support of distribution is on interval $[0, 1]$**.** Thus, we have a mixture of the Gaussian distribution with 2 point measures concentrated at 0 and 1. For simplicity we assume that both μ and σ are common for all experts.

Denote by W the maximal possible social ranking, the minimal possible social ranking is 0.

Also, we assume that each expert takes decisions following the utility function:

$$U(v, \lambda) = \frac{1 - e^{-\frac{\lambda x}{W}}}{1 - e^{-\frac{\lambda M}{W}}},$$

where M is the total bank of the expert that can be interpreted as his/her social ranking. Thus, utility of doing nothing is

$$U(M, \lambda) = 1.$$

Note that λ is a risk attitude parameter that we assume to be common for all experts and to be dependent on the prediction market (i.e. on the test case under the consideration).

Suppose that a unit expert has made a bet for fraud at price $c = c_{for}$**.** It means that, from the expert's point of view, the true fraud probability lies in the interval $[p(c), 1]$**,** where $p(c)$ is found from the equality of mathematical expectation of the lottery utility function and the utility of doing nothing:

$$P(c)U(M + c - 1, \lambda) + (1 - P(c))U(M - 1, \lambda) = 1.$$

The solution should look like this:

$$P(c) = a\left(\frac{1}{c}, \lambda\right),$$

where

$$a(q, \lambda) = \frac{1 - e^{\frac{\lambda q}{W}}}{e^{\frac{-\lambda(1-q)}{W}} - e^{\frac{\lambda q}{W}}}.$$

Now suppose that a unit expert has made a bet against fraud at price $c_{against}$, and $\frac{\mathbf{1}}{\boldsymbol{c}} + \frac{\mathbf{1}}{\boldsymbol{c_{against}}} = \mathbf{1}$**.** It means that, from the expert's point of view, the true fraud probability lies in the interval $[0, p(c_{against})]$**,** where $p(c_{against})$ is found from the equality of the mathematical expectation of lottery utility function and utility of doing nothing:

$$(1 - P(c_{against}))U(M + c_{against} - 1, \lambda) + P(c_{against})U(M - 1, \lambda) = 1.$$

The solution should look like this:

$$P(c_{against}) = b\left(\frac{1}{c_{against}}, \lambda\right),$$

where

$$b(q, \lambda) = 1 - a(q, \lambda).$$

As a first step we translate all received orders into this ray bets as described above.

Next, consider the following process that results to what we call an **outer equation**:

1) Fix some λ**.**

2) Find μ and σ by maximizing the log-likelihood function:

$$\sum N(c_{for}) \ln(1 - cdf(p(c_{for}), \mu, \sigma)) + \sum N(c_{against}) \ln cdf(p(c_{against}), \mu, \sigma),$$

where $N(c_{for})$ is the number of experts that have voted for fraud at price c_{for}, and $N(c_{against})$ is the number of experts that have voted against fraud at price $c_{against}$. The sums are over all possible prices c_{for} and $c_{against}$, respectively.
Note that μ and σ do depend on λ, since ray probabilities depend on λ**.** But this dependence is only through ray probabilities.

3) Note that if **unit experts** know the fraud probability μ, and the current prices for an against fraud are $\frac{1}{\mu}$ and $1 - \frac{1}{\mu}$ respectively, approximately μ% of experts will vote for the fraud, and approximately $(1 - \mu)$% of experts will vote against the fraud. Using this simple observation, we construct what we call an **equilibrium equation:**

$$\ln(1 - \mu) + \ln\left(1 - cdf\left(b\left(\frac{1}{\mu}, \lambda\right)\right)\right) - \ln \mu - \ln\left(cdf\left(a\left(1 - \frac{1}{\mu}, \lambda\right)\right)\right) = 0.$$

We find λ_{next} as the solution of the **equilibrium equation.**

4) Following steps 1)-3) we have obtained:

$$\lambda_{next} = F(\lambda).$$

We call this equation an **outer equation**, we solve it for $\lambda_{next} = \lambda$, and find the corresponding μ and σ using step 2).

Parameter μ is our new estimate of fraud probability. We could use σ to construct confidence intervals for μ, but we do not do it for simplicity of vote distribution procedure.

This iterative procedure involves a lot of computations, but it is not supposed to occur **in life time.** We will use it for the final calculation of the fraud probability after the voting.
Also, it makes sense to consider using it every hour or so, in order to help our market maker defined by **Model D.**

Model F. Or estimating fraud probability by geographic location.

We use Dataset 4 for calculation of crime rankings by countries from crime rankings by cities (by calculation of the mean city ranking in each country). The numbers vary from Japan with 13.91 (the best ranking) to Senegal with 78.13 (the worst ranking); Germany, India, and Russia are ranked with 20.965, with 39.81, and with 55.075 respectively.

It is very tempting to use city rankings instead of country rankings in our models, but we decided not to do it, since currently we have not enough city specific data for this purpose.

Assume that Germany has fraud probability of 30% with crime ranking of 20.965 (the fraud probability is estimated from Dataset 3). Now consider Dataset 5 with data on propensity crimes in India. We see that the ratio of all recovered cases to all stolen cases has mean of 37.72% and median of 34.2%. Thus, we can just assume for simplicity that fraud probability in India is 35%, i.e. by 16.667% higher than in Germany.

We assume the following simplest model:

$$\frac{\Delta p}{p} = \kappa \Delta r,$$

where p is the fraud probability, Δp is the change of fraud probability, Δr is the change of ranking, and κ is the step size. We estimate κ as follows:

$$\kappa = \frac{(35\% - 30\%)}{(39.81 - 20.96) * 30} = \frac{16.667\%}{(39.81 - 20.96)} = 0.8844\%.$$

Using this simple reasoning and Dataset 4 data, we can define the multiplicator for fraud probability for various countries. We set it to 1 for Germany with crime ranking 20.965, by definition. For Japan with crime ranking of 13.91 the fraud probability is decreased by 6.23%, for Russia with crime ranking of 55.075 the fraud probability is increased by 30.17%, finally, for Senegal with crime ranking of 78.13, the fraud probability is increased by 50.56%.

We call this multiplicator **Model F.** We suggest to use this model, in order to adjust the fraud probability obtained by **Model A** identifying by the geographic coordinates the country in which the phone was when the contract was signed. Note that this change is not drastic even for extreme cases like Japan or Senegal. Also note that the more the crime ranking for the original country is, the more is the fraud probability adjustment, which is quite natural.