Modeling Dependency in Prediction Markets

Nikolay Archak
narchak@stern.nyu.edu

Panagiotis G. Ipeirotis
panos@nyu.edu

Department of Information, Operations, and Management Sciences
Leonard N. Stern School of Business, New York University

ABSTRACT

In the last decade, prediction markets became popular forecasting tools in areas ranging from election results to movie revenues and Oscar nominations. One of the features that make prediction markets particularly attractive for decision support applications is that they can be used to answer "what if" questions and estimate probabilities of complex events. Traditional approach to answering such questions involves running a combinatorial prediction market, what is not always possible. In this paper, we present an alternative, statistical approach to pricing complex claims, which is based on analyzing co-movements of prediction market prices for basis events. Experimental evaluation of our technique on a collection of 51 InTrade contracts representing the Democratic Party Nominee winning Electoral College Votes of a particular state shows that the approach outperforms traditional forecasting methods such as price and return regressions and can be used to extract meaningful business intelligence from raw price data.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Economics, Theory

Keywords

Prediction markets, correlation, forecasting, decision support systems, copula, dependency model

1. INTRODUCTION

In the last decade, prediction markets became popular forecasting tools in areas ranging from election results [2] to movie revenues and Oscar nominations [12]. The empirical performance of such markets is remarkably high, and they often outperform traditional forecasting tools, such as polls [2]. This success led many corporations (e.g., HP [5], BestBuy, and Google [8]) to adopt internal prediction markets for forecasting product launch dates, sales volumes, and other important variables. Reports indicate that prediction markets can serve as excellent information aggregation mechanisms, allowing participation of employees, independent of rank, in the decision process.

One of the features that make prediction markets particularly attractive for decision support applications is that they can be used to answer "what if" questions. For instance, Berg and Rietz show that Iowa Political Markets for 1996 Presidential elections could have been used to infer that Dole was a weak candidate *against Clinton as the Democratic nominee* [3]. In a hypothetical scenario, if the Republican party could correctly assess the probability of Dole winning against Clinton, they might have chosen a different nominee (Powell). In general, there are two ways to assess such probability using a prediction market. The first ("direct") approach is to let traders buy and sell combinatorial claims, i.e., claims on combination of more than one event. For instance, the probability of event B conditional on event A can be derived from prices of two contracts - a simple contract on the event A and a joint contract on both events:

$$P(B \| A) = \frac{P(A \wedge B)}{P(A)}.$$

Unfortunately, the vast majority of prediction markets handle only non-combinatorial bids, although there are some notable exceptions such as Yoopick [10]. One of the reasons for that is the complexity of the auctioneer's matching problem, which was shown to be NP-complete, if traders' orders are assumed to be indivisible [9]. While the matching problem becomes tractable and can be solved efficiently using a linear programming approach [9], if the orders are allowed to be partially accepted, it is not yet clear what effect on liquidity and price formation can such partial acceptance have, in particular, when the solution of the corresponding linear program is not unique. Finally, even if such prediction market is "up and running", there is no guarantee that the demand for complex contracts will be sufficient to determine probability of the corresponding scenario. In the worst case, there might be demand for the claim A and the claim B but not for the joint contract $A \wedge B$, therefore the market will fail to establish the price of the joint claim. We conjecture that, while combinatorial prediction markets have significant potential for decision support applications, there is also strong need for an alternative, statistical technique to pricing complex claims.

The main goal of this paper is to present a framework for construction and evaluation of such statistical techniques, as well as to demonstrate that they can be used to extract meaningful business intelligence. Our framework provides a template for building models of contemporaneous dynamics of both claim prices as new information is revealed to the market. The observed dynamics provides information on the latent dependency between two events, which in turn can be used to determine the joint probability of both events happening.

Example 1.0.1 *Consider two markets for presidential elections. The first market tracks the probability of party A winning the election. The second market tracks the probability of candidate B being the nominee for party A. If the first market moves negatively every time the second market moves upwards, then it is likely that the chances of party A winning the election are decreased if they nominate candidate B.*

Co-movements of prediction market claims were studied by some researchers in the past. For instance, Berg and Rietz present regressions of the price of the Clinton's Presidential elections contract on the corresponding prices of Dole's and Powell's contracts [3]. In this paper, we build on prior research in finance, which shows that complicated dynamics of stock returns is best modeled by separating the marginal distributions of each stock, from the model of dependency [13]. Thus, marginal-free dependency models, also known as copulas [15], are the corner-stones of our framework. In recent years, copulas gained significant popularity for dependency modeling in areas ranging from option pricing [6] to decision and risk analysis [7]. As the main contribution of our paper, we show how copulas can be used to learn dependency between prediction markets and how the dependency learned can be used to forecast future market evolution. Our experimental results show that copula based techniques significantly outperform traditional forecasting approaches such as price regressions.

The rest of the paper is organized as follows. Section 2 presents our framework for building dependency models and gives generic algorithms for inference and forecasting in the framework. Section 3 presents our experimental results obtained for a collection of InTrade prediction markets, evaluates predictive performance of the copula-based models and shows that such models can be used to extract useful business intelligence. Finally, Section 4 concludes the paper with a short summary of the results.

2. FRAMEWORK

For simplicity of presentation and evaluation, we will restrict consideration by a hypothetical scenario with two base events A and B; all models presented in the paper can be easily extended to the case of more dimensions and complex Boolean expressions. It is essential for our model that both events A and B are contemporaneous, i.e., whether the events happen or not is realized at the same moment of time $T > 0$. The prediction market for corresponding claims starts at time zero; we will use p_t^A to denote the time t price of a claim that pays \$1 at time T if the event A is realized and zero otherwise; p_t^B represents corresponding claim for the second event. Formally, these statements can be written as

$$p_T^A = \mathbb{I}(A), \quad p_T^B = \mathbb{I}(B),$$

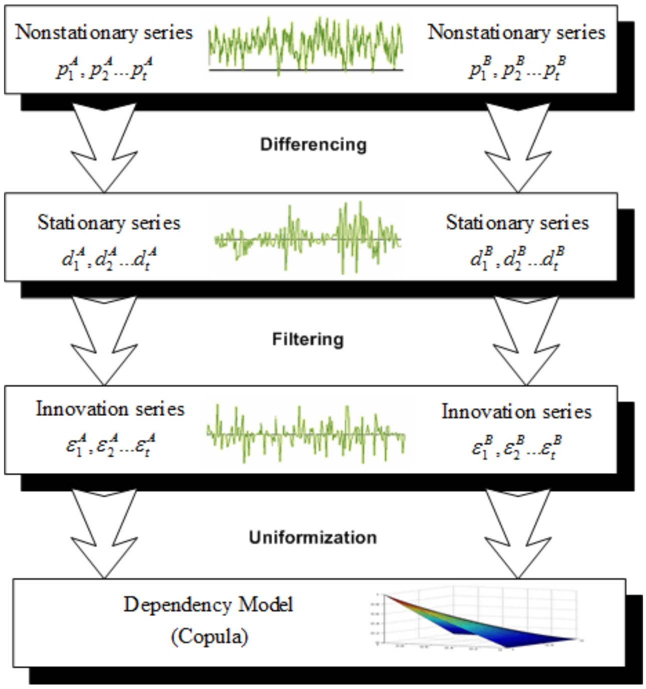

Figure 1: Framework for dependency modeling

where $\mathbb{I}$ is the indicator function.

Our model is based on a simple idea that the observed history of price movements will provide information on the latent dependency between two events, which in turn can be used to determine the joint probability of both events happening. Of course, no conclusions can be made unless sufficient structure is placed onto the problem, and the choice of structure always involves tradeoff between expressiveness of the model and the amount of data needed to learn the model relationships. Considering relatively short duration of an average prediction market contract, our paper proposes the framework displayed in Figure 1. The framework consists of four basic components: differencing, filtering, uniformization and dependency modeling. The components are designed to be independent and pluggable, thus the framework can easily be customized for needs of a particular application.

2.1 Differencing

Granger and Newbold's seminal paper on spurious regressions [11] describes the following experiment. Two hundred of independent random walks $X_t^i, (i = 1..200, t = 1..50)$, each of length 50, are generated. Using the traditional t test with the 95% confidence level, the authors tested the null hypothesis of no relationship in the regression of X^{2i} on X^{2i+1} for each $i = 1..100$. Amazingly, the null hypothesis was rejected (wrongly) on approximately threequarters of all occasions, instead of the anticipated 5%. One can see, that the autoregressive behavior of observations makes significance tests, based on the standard errors from the OLS estimator, invalid. Moreover, even if standard errors are corrected properly, forecasts based on the regression equations are likely to be sub-optimal due to inefficient use of data [11].

As prices of prediction market contracts are likely to have properties of a unit root process like the random walk, we argue that using prices directly as modeling units is not a good idea: statistical analysis of prices may produce results that are spurious and decisions, based on analysis of prices rather than price changes, should be made with caution. Therefore, the first step of our framework is to transform the price series $p_1^{A,B}, ..., p_t^{A,B}$ to series of stationary observations *and* remove autoregressive behavior. We call this step "Differencing" because of similarity to traditional stationarization approaches that require taking differences to obtain series of price changes

$$a_t = p_t - p_{t-1}$$

or returns

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}}.$$

Unfortunately, even if the prediction market is efficient [1], the transformation of price series p_t to return series $r_t$is not sufficient to achieve stationarity. The reason for that is the finite time horizon of the market: at time T the contract will expire. There is significant theoretical and empirical evidence that unconditional distribution of returns in the prediction market depends on the time to contract expiration: in particular, out of two contracts having the same current price, the one closer to expiration is usually more volatile [1]. Explicit dependency of the second moment of the distribution on the time to expiration is a violation of stationarity and must be taken into account before our framework can be applied successfully.

The solution we propose in this paper is to use the prediction market volatility model of Archak and Ipeirotis [1], who found that instantaneous volatility of prices [2] in binary prediction markets is of the following form

$$\Sigma_t = \frac{1}{\sqrt{T-t}} \phi\left(\Phi^{-1}(p_t)\right), \quad (1)$$

where Φ is the distribution function and ϕ is the density function of a standard normal distribution. Instead of usual price differences or return series, we suggest using normalized differences [3]:

$$d_t = \frac{p_t - p_{t-1}}{\Sigma_{t-1}}. \quad (2)$$

Finally, less liquid prediction markets may exhibit return predictability, so additional steps might be necessary to ensure absence of autoregressive behavior. For instance, one may consider AR(p) regression of series d_t on its lag:

$$d_t = c + \sum_{i=1}^{p} \varphi_i d_{t-i} + \omega_t,$$

and use residuals from the model (ω_t) as output of the Differencing step.

[1]On a related note, Leigh and Wolfers [14] provide statistical evidence that Australian betting markets for 2004 Australian elections were at least weakly efficient.

[2]Note that this expression gives the price not the return volatility. The return volatility can be obtained as $\frac{\Sigma_t}{p_t}$

[3]If the time interval between observations is significant compared to the time to contract expiration, one should not normalize by the instantaneous volatility Σ_{t-1} but by the forecast of the average expected volatility between times $t-1$ and t. Refer to Theorem 3 and Equation 8 in [1].

2.2 Filtering

Stationarity does not imply independence and stationary time series can exhibit significant dependency between adjacent observations. In markets, such dependency usually comes through conditional volatility effects, such as volatility clustering: large returns are often followed by other large returns (not necessarily of the same sign); nevertheless, our framework does not forbid modeling other types of relationships.

We think of the market behavior in each period as being affected by two different components: the prior history of the market evolution (in our example, the history defines volatility of new return as a function of prior volatility and prior returns) and the innovation component $\varepsilon_t^{A,B}$ representing the new shock to the market in the period t. We hypothesize, that, in order to use the available data efficiently, one should model dependency directly between the innovation components ε_t^A and ε_t^B for both prediction markets, rather than between returns that represent the mix of innovation shocks and prior history. The main challenge here is being able to extract the innovation component, which is not directly observable. Consistent with statistical literature, we call this step "Filtering", as it generally requires estimating the latent state of a dynamic system with a stochastic component.

Example 2.2.1 *Assume that normalized differences d_t follow GARCH(1,1) process:*

$$d_t \,|d_{t-1}, d_{t-2}... = h_t \varepsilon_t,$$

$$h_t^2 = \alpha_0 + \alpha d_{t-1}^2 + \beta h_{t-1}^2,$$

where h_t represents volatility, $\varepsilon_t \sim \mathcal{N}(0,1)$ are i.i.d. residuals (innovations). The maximum likelihood estimator of the model [4] will produce estimates of the coefficients $\hat{\alpha}_0, \hat{\alpha}, \hat{\beta}$ and the latent volatility $\hat{h}_t$. The goal of the Filtering step is to extract the innovation terms, which can be estimated as:

$$\hat{\varepsilon}_t = \frac{d_t}{\hat{h}_t}.$$

We emphasize that volatility h_t desrcibed in Example 2.2.1 is different from volatility Σ_t that was used in the Differencing step: Σ_t represents the *unconditional* volatility of the prices [4] while h_t represents the conditional component of the volatility designed to capture volatility clustering. Thus, the actual volatility comes as an interplay between these two components.

2.3 Uniformization and Dependency Modeling

In applied literature, it is common to measure the level of dependency between two random variables by the correlation between them: the higher the correlation coefficient, the more dependent the variables are. A good example of such way of thinking is linear regression; indeed, the OLS regression coefficient is nothing but the correlation coefficient after proper scaling. While there are numerous cases, when thinking of dependency as correlation is well justified, there are also situations when a different measure of dependency is needed. The main issue with using the correlation

[4]Only the current price and time to the expiration are known, but not the prior price history.

coefficient is that it is not invariant to nonlinear transformations of the data. For instance, while any random variable x is perfectly correlated with itself, the correlation will break if one considers x and x^3, although the dependency is still perfect: each random variable perfectly predicts the other one.

An alternative approach is to think of the dependency model between two random variables x and y as being separate from the marginal distributions of both variables: if strictly increasing transformations f and g are applied to both variables ($x \Rightarrow f(x)$, $y \Rightarrow g(y)$) the dependency should not change as the process can easily be reverted. The idea can be developed further by noting that, if x and y have cumulative density functions Φ_1 and Φ_2, one can always transform them to uniformly distributed random variables u and v by $x \Rightarrow u = \Phi_1(x)$ and $y \Rightarrow v = \Phi_2(y)$ while still preserving our alternative definition of the dependency. The cumulative density function of the resulting joint distribution with uniform marginals is known in statistics as copula [15].

Definition 1. A bivariate copula is a cumulative density function C of a joint distribution defined on the twodimensional unit cube $[0,1]^2$ such that both marginal distributions are uniform on the interval $[0,1]$.

The observation, that any joint distribution of random variables can be decomposed to its marginal distributions and its copula, is due to Sklar [15].

Theorem 1 ***(Sklar's Theorem)*** *For any bivariate distribution function $H(x,y)$, let $F(x)$ and $G(y)$ be the univariate marginal probability distribution functions. Then there exists a copula C such that*

$$H(x,y) = C(F(x), G(y)).$$

Moreover, if marginal distributions, $F(x)$ and $G(y)$, are continuous, the copula function C is unique.

In recent years, copulas gained significant popularity for dependency modeling in areas ranging from option pricing [6] to decision and risk analysis [7]. In our model, copula is used to specify dependency between contemporaneous innovations for both prediction markets ε_t^A and ε_t^B. Before innovations can be passed to copula, they should be stripped of the marginal distributions and cast to the $[0,1]$ interval. We call this step "Uniformization" as its goal is to ensure that the distribution of innovations is close to uniform. Typically, a normal distribution of innovations can be assumed, therefore uniformization can be performed by passing innovations through the distribution function of a standard normal distribution. In some cases, the market can exhibit significant "fat tails" inconsistent with normality assumption. If so, we suggest replacing the normal distribution with the Student's t-distributition with ν degrees of freedom, where the number of degrees of freedom can be estimated from the data using the maximum likelihood approach.

Once the innovations are cast to the $[0,1]$ interval, a copula can be applied to determine the probability of co-movement. The following copulas are frequently used in the applied work:

1. *Product copula* is a non-parametric copula specifying independent evolution of the innovations:

$$C(u,v) = uv.$$

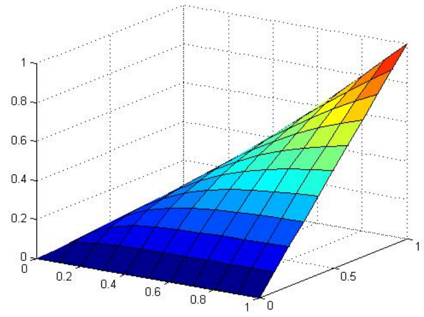

Figure 2: Clayton copula, $\theta = 2.0$

If the product copula is adopted in our model, the evolution of both prediction markets will proceed independently, therefore it is equivalent to assuming that $P(A \wedge B) = P(A)P(B)$.

2. *Gaussian copula* $C_\rho(u,v)$ is a copula of a bivariate normal distribution with correlation ρ:

$$C_\rho(u,v) = \Phi_\rho(\Phi^{-1}(u), \Phi^{-1}(v)).$$

Note that adopting the Gaussian copula together with the Gaussian marginal distributions of ε_t^A and ε_t^B is equivalent to assuming joint normality of both innovations. Nevertheless, the Gaussian copula need not be used with the Gaussian marginals and can, for instance, be applied when the marginals are t-distributions.

3. *Clayton copula* is another popular type of a parametric copula:

$$C_\theta(u,v) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-\frac{1}{\theta}}, \quad \theta > 0.$$

It is typically used to model asymmetric dependency between innovations: as this copula exhibits greater dependency in the negative tail of the distribution than in the positive one, the effects of positive news on the market behavior will be less synchronous than the effects of negative news.

4. Clayton copula can be inverted to model greater dependency in the positive tail of the distribution:

$$C_\theta(u,v) = 1 - \left((1-u)^{-\theta} + (1-v)^{-\theta} - 1\right)^{-\frac{1}{\theta}}, \quad \theta > 0.$$

2.4 Inference

Any model, adhering to our framework, can be estimated using a simple top-down approach: starting from the top block, estimate its parameters and pass the residuals down to the next block. The following example illustrates this idea.

Example 2.4.1 *Assume the data on evolution of both contracts until time $\tau < T$ is available.*

1. *Differencing step. Transform all prices $p_t^{A,B}$ into normalized differences $d_t^{A,B} = \frac{p_t^{A,B} - p_{t-1}^{A,B}}{\Sigma_{t-1}^{A,B}}$, where $\Sigma_t^{A,B} =$*

$\frac{1}{\sqrt{T-t}}\phi\left(\Phi^{-1}(p_t^{A,B})\right)$. Perform AR(1) regression to remove potential autoregressive behavior

$$d_t^{A,B} = c + \varphi d_{t-1}^{A,B} + \omega_t^{A,B}.$$

Estimated residuals from the regression $\hat{\omega}_t^{A,B}$ are passed to the next step.

2. *Filtering step. Run standard GARCH(1,1) estimator on the input $\omega_t^{A,B}$ to remove volatility clustering.*

$$\omega_t^{A,B} = h_t^{A,B}\varepsilon_t^{A,B},$$

$$h_t^{A,B^2} = \alpha_0 + \alpha\omega_{t-1}^{A,B^2} + \beta h_{t-1}^{A,B^2}.$$

Estimated standardized GARCH residuals $\hat{\varepsilon}_t^{A,B}$ are passed to the next step.

3. *Uniformization step. No estimation necessary. Assuming normal distribution of innovations $\varepsilon_t^{A,B}$ cast them to uniform:*

$$u_t = \Phi\left(\varepsilon_t^A\right), \quad v_t = \Phi\left(\varepsilon_t^B\right).$$

4. *Dependency model. Assuming that dependency between contemporaneous observations is specified by Clayton's copula $C_\theta(u,v) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-\frac{1}{\theta}}$, the log-likelihood function is*

$$\log\mathscr{L} = \tau\log(1+\theta) -$$

$$\sum_{t=1}^{\tau}\left[(\theta+1)\log(u_t v_t) + \left(\frac{1}{\theta}+2\right)\left(u_t^{-\theta} + v_t^{-\theta} - 1\right)\right].$$

The dependency parameter θ can be estimated using any optimization method. Figure 2 shows a sample Clayton copula.

Note that the model configuration used in Example 2.4.1 represents only one of many possible setups. The components of the framework are designed to be independent and pluggable, thus the framework can easily be customized for needs of a particular application.

2.5 Forecasting

After the model parameters have been estimated, one needs to forecast into the future, in order to estimate the probability of the joint event. In most cases, the closed form representation of the forecast might not be available, therefore we propose a simulation based approach. The algorithm is given by Algorithm 1. To simplify understanding, we have formulated the algorithm in terms of Example 2.4.1, however it can be easily extrapolated to any other model.

The idea of Algorithm 1 is to run the data generation process from the last observed moment of time τ until the closing date of the prediction market T. The process is repeated L times and the estimate of the joint probability is obtained as a fraction of times a "hit" occurred - price trajectories for both claims A and B converged to 1. Each price trajectory is constructed step by step. The process starts by generating a pair of uniformly distributed innovations u_t and v_t, such that the dependency between them is given by the copula C. The innovations are transformed to have proper marginal distributions by reversing the Uniformization step and fed to the Filtering module to update volatilities (or other latent parameters) and reconstruct the difference series (or any other series that are the output of the Differencing step). Finally, the Differencing step is reversed to reconstruct the new price observation for both contracts.

We emphasize, that for practical application of the algorithm, it is essential to ensure that simulated prediction market prices converge to 0.0 or 1.0 at time T. Using GARCH or any other volatility model for stocks alone will not be enough, as GARCH prices need not converge to anything, however, if the normalization factor $\Sigma_t = \frac{1}{\sqrt{T-t}}\phi\left(\mathscr{N}^{-1}(p_t)\right)$ is used to model unconditional volatility, convergence will happen for sufficiently small simulation step.

Algorithm 1 Estimating the probability of the joint event using Monte-Carlo

```
L ⇐ number of trajectories to generate
H ⇐ 0 {Total number of hits}
for l = 1 to L do
  for t = τ + 1 to T do
    u_t ⇐ uniform random sample on [0, 1]
    v_t ⇐ sample from C(v|u_t) {Now the joint distribution
      of (u_t, v_t) is given by the copula C}
    u_t ⇐ Φ^{-1}(u_t)
    v_t ⇐ Φ^{-1}(v_t) { Reverse the Uniformization step }
    h_t^A ⇐ sqrt(α_0 + α ω_{t-1}^A^2 + β h_{t-1}^A^2)
    h_t^B ⇐ sqrt(α_0 + α ω_{t-1}^B^2 + β h_{t-1}^B^2) {Calculate condi-
      tional volatilities}
    ω_t^A ⇐ h_t^A u_t
    ω_t^B ⇐ h_t^B v_t {Reverse the Filtering step}
    d_t^A ⇐ φ d_{t-1}^A + ω_t^A
    d_t^B ⇐ φ d_{t-1}^B + ω_t^B {Reconstruct the new normalized
      differences}
    p_t^A ⇐ p_{t-1}^A + d_t^A Σ_{t-1}^A
    p_t^B ⇐ p_{t-1}^B + d_t^B Σ_{t-1}^B {Reconstruct the new prices}
    Σ_t^A ⇐ 1/sqrt(T-t) φ(N^{-1}(p_t^A))
    Σ_t^B ⇐ 1/sqrt(T-t) φ(N^{-1}(p_t^B)) {Calculate unconditional
      volatilities for the next step}
  end for
  if p_T^A > 1 - ε_0 AND p_T^B > 1 - ε_0 then
    H ⇐ H + 1 {Increase hit count}
  end if
  P(A ∧ B) ⇐ H/L
end for
```

Algorithm 1 assumes that one is able to sample from conditional distribution $C(v|u)$. For that purpose it is sufficient to know the conditional density of v, which for continuous density copulas is given by

$$\frac{\frac{\partial^2 C}{\partial u\partial v}(u,v)}{\int\frac{\partial^2 C}{\partial u\partial v}(u,v)du} = \frac{\partial^2 C}{\partial u\partial v}(u,v).$$

Here we used the fact that, by definition, all copulas have uniform marginal distributions. For instance, for Clayton copula, the distribution function of v conditional on u is given by

$$C(v|u) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-\frac{1}{\theta}-1}u^{-\theta-1},$$

and the sampling can be performed by inversion of the distribution function (sample z uniformly on $[0,1]$ and return

Variable	Min	Max	Mean	Median	S.Dev.
Daily dev.	-0.882	0.88	0.0003	0	0.018
Returns	-0.967	29.333	0.005	0	0.247
Price	0.02	0.983	0.494	0.506	0.348

Table 1: Descriptive Statistics for 51 InTrade contracts

$C^{-1}(\cdot|u)(z))$.

3. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of our model on a set of InTrade prediction market contracts.

3.1 Data

Our dataset contained daily observations for a collection of InTrade prediction market contracts. Intrade is an online Dublin-based trading exchange website founded in 2001. The trading unit on InTrade is a contract with a typical settlement value of \$10, which is measured on a 100 points scale. To be consistent with our convention that the winning contract pays \$1, we renormalized all price data to be in $[0, 1]$ range, so, for example, a 50 points price of InTrade contract is represented by 0.5 in our dataset. The full dataset included daily closing price and volume data for a collection of 901 InTrade contracts obtained by periodic crawling of InTrade's website. [5] Detailed description and analysis of the dataset are provided in [1].

In order to evaluate the quality of the predictions of our model, the first step was to select a subset of contracts which can be expected to have sufficient dependency between the outcomes. In this paper, we report our test for a subsample of 51 InTrade contracts, where each contract represents the Democratic Party Nominee winning Electoral College Votes of a particular state [6] in the 2008 Presidential Election. Descriptive statistics for this set of contracts are given in Table 1. Each contract had exactly 306 daily observations with the last observation made on November 1st, 2008, several days before the Election.

We tested all contracts for weak efficiency by performing AR(1) regression of daily returns on a constant and the lag of the return. In 16 out of 51 contracts we found statistically significant negative serial correlation with coefficient on the lagged return varying from -0.14 (Louisiana) to -0.35 (Alaska). Further observation has shown that this is an artefact of low liquidity of contracts sufficiently far from expiration: if the illiquid tail of contracts (observations further than 100 days from the contract expiration) are removed from consideration, the effect disappears.

3.2 Testing Predictive Power of the Model

The most straightforward approach to testing our model would be to check if the predictions of our Monte-Carlo algorithm match the reality better than predictions of some baseline model, like the model assuming that events A and B are independent. Unfortunately, in such approach we have only one realized outcome for each event (whether it happened or not), thus making the sample size too small to make reasonable conclusions. We propose an alternative test which is based on forecasting the conditional probability $P(A|B)$ rather than the joint one.

Assume the price trajectories for both claims are observed until time τ. Moreover, one also observes the price trajectory of the second claim until time $\tau+K < T$: $p^B_{\tau+1}, ..., p^B_{\tau+K}$. If our dependency model fits the data well, it can be used to predict the price $p^A_{\tau+K}$ better than models based on price trajectory of the first claim alone. As the model can be tested with different values of τ for each pair of contracts, we now have significantly more observations to estimate performance of the model.

In the following, it will be more convenient to count time relative to the Election day (November 4th, 2008), which is set to be day zero ($T = 0$). For instance, $t = -3$ will represent November 1st, the last day in our dataset. The data was split into the training sample and the test sample based on the time variable: observations $t \in [-100, -50]$ were in the training set, $t \in [-49, -3]$ were in the test set [7]. There is significant empirical evidence that people tend to overvalue small probabilities and undervalue near certainties, i.e., the so called "favorite-longshot bias" [16], therefore, we further eliminated contracts that, as of day -50, had price greater than 0.9 or smaller than 0.1 [8]. After the elimination only 19 "uncertain" contracts were left. We emphasize that for the rest of the contracts, the price as of day -50 exactly predicted the outcome of the contract: for contracts with $p > 0.9$ the Democratic Nominee won, and for contracts with $p < 0.1$ lost.

The following baseline models were considered:

1. Martingale model (**M**): price of each contract is a martingale *conditional on prices of both contracts*, i.e.,

$$E\left\{p^A_{\tau+K}\,\middle|\,p^A_0, ..., p^A_\tau, p^B_0 ... p^B_{\tau+K}\right\} = p^A_\tau.$$

   Note that this model implies independent evolution of both prediction markets, in particular, that

$$P(A \wedge B) = p(A)p(B).$$

2. Price regression (**P**): Regress p^A_t on constant and p^B_t for $t \le \tau$. Predict

$$p^A_{\tau+K} = \beta_0 + \beta_1 p^B_{\tau+K}.$$

   This is essentially the approach of [3].

3. Return regression (**R**): Regress $r^A_{t,K}$ on constant and $r^B_{t,K}$ for $t \le \tau$, where $r_{t,K}$ is the K-day return:

$$r_{t,K} = \frac{p_t - p_{t-K}}{p_{t-K}}.$$

   Predict

$$p^A_{\tau+K} = (1 + \beta_0 + \beta_1 r^B_{\tau+K})p^A_\tau.$$

The baseline models were compared against the following three models from our framework: model with the Gaussian

[5] InTrade keeps historical data for each contract on the website, however expired contracts disappear from the website after certain amount of time.

[6] More precisely, one the fifty states or Washington D.C.

[7] The reason that we excluded observations before the day -100 from consideration is that the market was very inactive and illiquid that far from the contract expiration, therefore all models actually perform better if trained on a reduced dataset.

[8] Behavior of such contracts is also significantly affected by microstructure and transaction costs that complicate extraction of innovation terms.

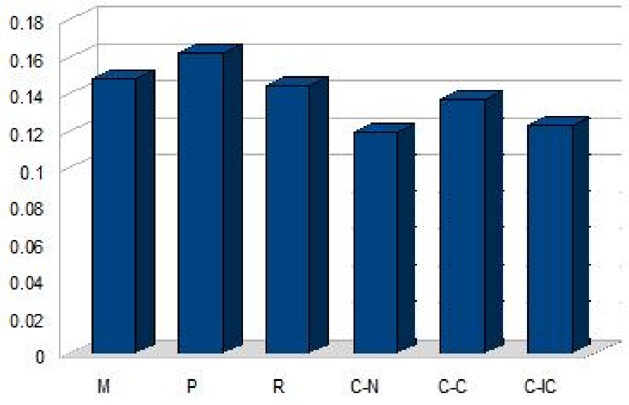

Figure 3: Mean-squared Error on Test Set

copula (**C-N**), model with the Clayton copula (**C-C**) and model with the inverse Clayton copula (**C-IC**). All three models used normalized price differences in the Differencing step, GARCH(1, 1) in the Filtering step and normal marginals in the Uniformization step. To ensure sufficient liquidity (reduce number of no-trade observations), training of these models was performed using 3-day returns.

To make forecasting problem more interesting than just forecasting the price tomorrow, we used $K = 15$, i.e., all models were asked to forecast the price fifteen days later. Out of 19 interesting contracts, we have composed all possible pairs of (different) contracts for which raw correlation between returns on the training set had an absolute value of at least $\rho_0 = 0.3$. With these values of parameters, the resulting test set had $1,498$ observations. To ensure reasonable performance of Algorithm 1, we used $L = 100$, i.e., for each prediction a hundred sample paths were generated.

3.3 Discussion

Mean-squared errors of all six models on the test test are shown in Figure 3. Using mean-absolute error instead, results in the same ranking of all six methods. As expected, the price regression (**P**) consistently performed the worst, with mean-squared error of 0.1624 on the test sample, which is even worse than the result of a naive martingale predictor **M** (0.1483). This result provides confirmation for our argument that price regressions for prediction markets are spurious and should be avoided because of weak out-of-sample performance.

The return regression (**R**) has slightly outperformed the martingale predictor with mean-squared error of 0.1443. The copula models have performed significantly better with the best result for the Gaussian copula **C-N** (0.1193) which has given almost 20% reduction in the mean-squared error compared to the martingale predictor.

We have tested our results for robustness by trying different aggregation techniques and values of ρ_0. Similar but slightly worse performance results were obtained with copula models trained on daily returns or 5-day returns. The issue with daily returns is that there are more zero volume days in the training set than in the test set as the test set is closer to the contract expiration. Difference in liquidity between training and test sets decreases performance of the copula model on the test set, therefore we recommend using certain level of aggregation. Using too high level of aggregation is also undesirable as it reduces information available to the copula model; for instance, with $K = 15$ and 3-day aggregation level, the model will use 5 observations for each single prediction, while at 5-day aggregation level it will use only 3. Nevertheless, with both daily and 5-day returns, copula models outperformed the regression of returns and the martingale predictor. If value of the correlation threshold (ρ_0) is decreased, i.e., the test set is extended to include pairs of contracts with low correlations, performance of the martingale predictor is not significantly affected, while performance of the copula models naturally deteriorates as, on average, each contract in the pair has less information about its counterpart. For instance, with $\rho_0 = 0.2$, the mean-squared error of the Gaussian copula increases to 0.1252. Across many different settings, the Gaussian copula model consistently outperformed two models with asymmetric copulas ($\mathbf{C} - \mathbf{C}$ and $\mathbf{C} - \mathbf{IC}$), what makes us believe that asymmetric dependencies frequently observed in financial markets, were not present in our set of prediction market contracts. We also tried relaxing some of the framework components. In particular, we found that GARCH(1,1) component had no significant effect on performance of the copula models and could be omitted.

3.4 Extracting Business Intelligence

The experiment we described above shows that our framework can be used to construct models outperforming traditional forecasting techniques such as price or return regressions. In this section, we show that it also can be used to extract interesting and potentially useful business intelligence. Using the same sample of Presidential Election contracts, we ask the question: “Which pairs of states are most similar in the sense that winning one state significantly increases the probability of winning the other one?”. A natural ranking criteria for this question is lift:

$$L(A|B) = \frac{P(A|B)}{P(A)} = \frac{P(A \wedge B)}{P(A)P(B)}.$$

We used Gaussian copula from the previous experiment to estimate dependencies between pairs of states from full historical price data until November 1st, 2008. All contracts were included except for those that had price smaller than 0.1 or greater than 0.9 as of November 1st. Because some included contracts were not very liquid, we used weekly aggregation level. Using estimated dependency parameters, we forecast from November 1st until the date of Elections to estimate the value $P(A \wedge B)$. Top lift values obtained are given in Table 2. One can see, for instance, that winning South Dakota lifts probability of winning North Dakota by a factor of two.

Additionally, we repeated the experiment by running all state contracts against PRESIDENT2008.DEM, the contract on the Democratic Nominee winning the 2008 Presidential Elections. Results are given in Table 3. The market identified Missouri and Georgia as key predictors, followed by Montana and Florida.

4. CONCLUSION

In the last decade, prediction markets became popular decision support tools, employed by corporations to forecast product launch dates, sales volumes, and other important variables. Decision support applications of prediction markets frequently require estimating probabilities of complex, combinatorial events, and running a combinatorial prediction market is not always feasible. This paper presents a

Contract A	P(A)	Contract B	P(B)	P(A|B)	Lift
sth.dakota	0.1	nth.dakota	0.29	0.21	2
nth.dakota	0.29	sth.dakota	0.1	0.58	2
sth.carolina	0.11	sth.dakota	0.1	0.19	1.83
sth.dakota	0.1	sth.carolina	0.11	0.19	1.83
sth.carolina	0.11	nth.dakota	0.29	0.18	1.73
nth.dakota	0.29	sth.carolina	0.11	0.5	1.73
westvirginia	0.13	sth.carolina	0.11	0.21	1.63
sth.carolina	0.11	westvirginia	0.13	0.17	1.63
sth.dakota	0.1	westvirginia	0.13	0.17	1.61
westvirginia	0.13	sth.dakota	0.1	0.21	1.61
westvirginia	0.13	georgia	0.29	0.2	1.53
georgia	0.29	westvirginia	0.13	0.44	1.53
sth.dakota	0.1	montana	0.25	0.15	1.49
montana	0.25	sth.dakota	0.1	0.37	1.49
nth.dakota	0.29	montana	0.25	0.4	1.38
montana	0.25	nth.dakota	0.29	0.34	1.38
georgia	0.29	sth.carolina	0.11	0.39	1.36
sth.carolina	0.11	georgia	0.29	0.14	1.36
sth.carolina	0.11	montana	0.25	0.14	1.34
montana	0.25	sth.carolina	0.11	0.33	1.34
...	...	...	...	...	...

Table 2: Pairs of Election Contracts Ranked by Lift Values

Contract A	P(A)	Contract B	P(B)	P(A|B)	Lift
president2008	0.84	missouri	0.57	0.93	1.1
president2008	0.84	georgia	0.29	0.92	1.09
president2008	0.84	montana	0.25	0.9	1.07
president2008	0.84	florida	0.69	0.89	1.06
president2008	0.84	arizona	0.19	0.89	1.06
president2008	0.84	indiana	0.38	0.88	1.05
president2008	0.84	ohio	0.82	0.88	1.04
president2008	0.84	westvirginia	0.13	0.88	1.04
president2008	0.84	nevada	0.82	0.88	1.04
president2008	0.84	nth.carolina	0.67	0.88	1.04

Table 3: Top 10 state contracts giving the highest lift for probability of winning Presidential Elections

framework for an alternative, statistical approaches to pricing complex combinatorial claims. The framework is based on a simple idea that the observed history of price movements will provide information on the latent dependency between two events, which in turn can be used to determine the joint probability of both events happening. The framework consists of four basic components: differencing, filtering, uniformization and dependency modeling. The first three components are designed to extract innovations driving the price movements of both contracts, while the last step connects contemporaneous innovations via a copula. All components of the framework are designed to be independent and pluggable, thus the framework can easily be customized for needs of a particular application. The paper proposes a simple top-down approach to estimating parameters of the model from historical data and a Monte-Carlo based forecasting algorithm that simulates future evolution of the contracts until the expiration date. Experimental evaluation of our technique on a collection of 51 InTrade contracts representing the Democratic Party Nominee winning Electoral College Votes of a particular state shows that the approach outperforms traditional forecasting methods such as price and return regressions and can be used to extract meaningful business intelligence from raw price data.

5. ACKNOWLEDGMENTS

The authors would like to thank the participants at the TUILES seminar at New York University, Jason Ruspini, and David Pennock for their helpful comments on earlier versions of this paper.

6. REFERENCES

[1] N. Archak and P. G. Ipeirotis. Modeling volatility in prediction markets, Sept. 2008. Working Paper.

[2] J. Berg, R. Forsythe, F. Nelson, and T. Rietz. Results from a dozen years of election futures markets research, 2003. Working Draft, The Handbook of Experimental Economics Results.

[3] J. Berg and T. Rietz. Prediction markets as decision support systems. *Information Systems Frontiers*, 5(1):79–93, 2003.

[4] T. Bollerslev. Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*, 31:307–327, 1986.

[5] K.-Y. Chen and C. Plott. Information aggregation mechanisms: Concept, design and field implementation for a sales forecasting problem, 2002.

[6] U. Cherubini and E. Luciano. Bivariate option pricing with copulas. *Applied Mathematical Finance*, 9:69–85, June 2002.

[7] R. T. Clemen and T. Reilly. Correlations and copulas for decision and risk analysis. *Management Science*, 45(2):208–224, 1999.

[8] B. Cowgill, J. Wolfers, and E. Zitzewitz. Using prediction markets to track information flows: Evidence from Google, 2008. Working Paper.

[9] L. Fortnow, J. Kilian, D. M. Pennock, and M. P. Wellman. Betting boolean-style: a framework for trading in securities based on logical formulas. *Decision Support Systems*, 39(1):87 – 104, 2005.

[10] S. Goel, D. Pennock, D. M. Reeves, and C. Yu. Yoopick: A combinatorial sports prediction market. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1880–1881, 2008.

[11] C. W. J. Granger and P. Newbold. Spurious regressions in econometrics. *Journal of Econometrics*, 2(2):111–120, July 1974.

[12] HSX. Hollywood stock exchange traders hit 80% of oscar nominations for the 80th annual academy awards. Available at `http://www.hsx.com/about/press/080123.htm`, 2008.

[13] E. Jondeau and M. Rockinger. The copula-garch model of conditional dependencies: An international stock market application. *Journal of International Money and Finance*, 25(5):827 – 853, 2006.

[14] A. Leigh and J. Wolfers. Competing approaches to forecasting elections: Economic models, opinion polling and predictionmarkets. *The Economic Record*, 82(258):325–340, 2006.

[15] A. Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Publications de l'Institut Statistique de l'Université de Paris*, volume 8, pages 229–231. 1959.

[16] R. Thaler and W. Ziemba. Anomalies: Parimutuel betting markets: Racetracks and lotteries. *Journal of Economic Perspectives*, 2(2):161–174, 1988.