Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

AN IMPORTANCE SAMPLING METHOD FOR PORTFOLIO CVaR ESTIMATION WITH GAUSSIAN COPULA MODELS

Pu Huang

Business Analytics and Math Sciences
IBM Research
Yorktown Heights, NY 10598, USA

Dharmashankar Subramanian

Business Analytics and Math Sciences
IBM Research
Yorktown Heights, NY 10598, USA

Jie Xu

Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, USA

ABSTRACT

We developed an importance sampling method to estimate Conditional Value-at-Risk for portfolios in which inter-dependent asset losses are modeled via a Gaussian copula model. Our method constructs an importance sampling distribution by shifting the latent variables of the Gaussian copula and thus can handle arbitrary marginal asset distributions. It admits an intuitive geometric explanation and is easy to implement. We also present numerical experiments that confirm its superior performance compared to the naive approach.

1 INTRODUCTION

Estimating the risk of a portfolio through Monte Carlo simulation is a fundamental task in risk management. Different measures of risk call for different simulation techniques. Among the very many risk measures that have been proposed in the literature and applied in practice, Value-at-Risk (VaR) is arguably the most popular one. VaR was developed by JP Morgan in 1990s and since then has been widely adopted by the banking industry. The Basel II Accord has made it a mandatory risk measure for determining regulatory capital requirements. Recently, Artzner, Delbaen, Eber, and Heath (1999) argued that a risk measure is necessarily "coherent" to encourage diversification. This concept of coherence later was extended by Follmer and Schied (2002) to a larger set of convex risk measures. It turns out that VaR is not coherent and a coherent alternative is Conditional Value-at-Risk (CVaR). In this paper, we present an importance sampling procedure for CVaR portfolio risk estimation.

Literature on efficient sampling mechanisms for CVaR estimation has been scarce. In their work, Manistre and Hancock (2005) developed an importance sampling method to estimate CVaR for a single normal variable. For a sum of Bernoulli variables, Merino and Nyfeler (2004) showed a method to exponentially tilt the Bernoulli masses to generate importance samples for CVaR estimation. A relevant, but different, line of work is tail probability estimation, which concerns devising efficient simulation procedures to evaluate the extreme probability of exceeding a given threshold. One important application of such procedures is credit risk estimation. Glasserman and Li (2005), and Glasserman, Kang, and Shahabuddin (2008) have developed importance sampling methods to evaluate tail probabilities for the widely used CreditMetrics model.

We assume portfolio loss is driven by a Gaussian copula model. The random potential loss of each asset in the portfolio may follow an arbitrary marginal distribution. The dependency among assets are captured by correlation between a set of latent Gaussian variables. More specifically, we consider the following portfolio

$$L = a^T X = a_1X_1 + a_2X_2 + \ldots + a_nX_n,$$

where $a = [a_1, a_2, \ldots, a_n]^T \in R^n$ is the amount of capital invested in assets $i = 1, 2, \ldots, n$, and $X = [X_1, X_2, \ldots, X_n]^T \in R^n$ is a random vector that represents the potential losses of the assets. Our goal is

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

to evaluate the risk of total loss L of this portfolio, where risk is measured by $\mathrm{CVaR}_\beta(L)$. Recall that the CVaR value of a random variable L is the expectation value of L given that it is greater than its β-quantile. More specifically, let VaR_β denote the β-quantile of L, i.e.,

$$\mathrm{VaR}_\beta = \min\{l : \Pr(L \leq l) \geq \beta\},$$

then $\mathrm{CVaR}_\beta(L)$ can be defined as

$$\mathrm{CVaR}_\beta(L) = \mathbb{E}[L \mid L \geq \mathrm{VaR}_\beta].$$

A naive approach to estimate $\mathrm{CVaR}_\beta(L)$ is as follows: 1) draw a set of i.i.d. sampling points X^j from X, 2) compute $L^j = aX^j$ for each j, 3) use L^j to calculate $\widehat{\mathrm{VaR}}_\beta$, an estimated value of VaR_β, and 4) choose only those L^j's that are greater than $\widehat{\mathrm{VaR}}_\beta$, and average across them. The final result is the conditional sample average and is an estimate of $\mathrm{CVaR}_\beta(L)$.

Clearly, a major shortcoming of this approach is that, roughly speaking, only $1-\beta$ percent of sample points L^j's will eventually be selected to compute the conditional sample average. This usually leads to a large estimation variance because of insufficient sample size. This shortcoming becomes more severe as β approaches one, as less portion of the total sample points will be utilized for estimation.

To reduce estimation variance, we would like to sample from a shifted L to get more importance sample points that are greater than the quantile value VaR_β. However, since we can not directly sample L, let alone from a shifted L, we need to find an approach to draw importance samples Y^j from X, such that $a^T Y^j$ would behave as if they were sampled from a shifted L. This is achieved by properly shifting random vector X.

In the rest of paper, we first present our importance sampling approach in section 2; then we test our approach in section 3; finally in section 4, we discuss potential extensions to our method.

2 IMPORTANCE SAMPLING SCHEME

We use a Gaussian copula to capture the dependency structure among random losses $X_1, \ldots, X_n$. In such a model, random vector $X = [X_1, \ldots, X_n]^T$ is represented by a component-wise nonlinear transformation of a Gaussian vector $Z = [Z_1, \ldots, Z_n]^T$ as,

$$X_i = F_i^{-1}(\Phi(Z_i)), \quad i = 1, \ldots, n, \tag{1}$$

where $Z_i \sim N(0,1)$ is a standard normal random variable, $\Phi(\cdot)$ is the univariate standard normal cdf, $F_i(\cdot)$ is the marginal cdf of X_i, and $F_i^{-1}(\cdot)$ is the inverse of $F_i(\cdot)$. Since Z_i is standard normal, $\Phi(Z_i)$ is uniformly distributed on interval $[0,1]$, and thus the cdf of X_i is the same as $F_i(\cdot)$. To capture the dependency among X_i's, Z_i's are correlated with correlation matrix Σ_Z, i.e., Z follows distribution $N(0, \Sigma_Z)$. Without loss of generality, we assume Σ_Z is positive definite, i.e., Σ_Z has full rank. Otherwise, we can always reduce the dimension of Z to ensure Σ_Z has full rank. In this paper, we assume Σ_Z is given. Practically speaking, it should be estimated from data. The NORTA procedure is well-known for that purpose (Cario and Nelson. 1997).

We engineer an importance sampling scheme for X by shifting the Gaussian vector Z. Let $Y = [Y_1, \ldots, Y_n]^T$ be an random vector generated by applying the same non-linear transformation that generates X to a shifted Z, i.e.,

$$Y_i = F_i^{-1}(\Phi(Z_i + \Delta Z_i)), \quad i = 1, \ldots, n, \tag{2}$$

where ΔZ_i is the amount of shift imposed on Z_i, we want to draw an importance sample from Y to construct an estimator of $\mathrm{CVaR}_\beta(L)$ with reduced variance.

Note that since $Z_1, \ldots, Z_n$ are correlated, shifting a single Z_i will influence other Z_{ii}'s through the correlation matrix Σ_Z. We apply a linear transformation on Z to decouple this cross-influence effect.

Let

$$\Sigma_Z = U^T D U$$

be the singular value decomposition of the correlation matrix Σ_Z. Construct a random vector $V = [V_1, ..., V_n]^T$ as

$$V = D^{-1/2} U Z, \tag{3}$$

then $V_1, ..., V_n$ are n independent standard normal random variables. Since Σ_Z is positive definite, we know that $U^T = U^{-1}$ is an orthogonal matrix and $D^{-1/2}U$ has full rank. Therefore, the inverse map from V to Z can be written as

$$Z = D^{1/2} U^T V. \tag{4}$$

Any shift ΔV on V can be easily transformed to an corresponding shift $\Delta Z = D^{1/2} U^T \Delta V$ on Z through (4).

Let $V^1, ..., V^J$ be an i.i.d. sample of V generated by independently drawn J points from $N(0, I_n)$ (i.e., the standard n-dimensional multivariate normal distribution with n independent and standard normal components). Applying linear transformation (4) on each point V^j, we get a sample $Z^1, ..., Z^J$ of Z. Further applying nonlinear transformation (1) to each point Z^j, we get a sample $X^1, ..., X^J$ of X. Given such a sample of X, we calculate the empirical loss L^j for each point X^j as

$$L^j = a^T X^j. \tag{5}$$

Sort $L^1, ..., L^J$ in an ascending order. Let $L^{(1)}, ..., L^{(J)}$ denote the sorted sequence with $L^{(1)} \leq ... \leq L^{(J)}$, and K denote the largest integer such that $(J-K)/J \geq 1-\beta$, i.e.,

$$K = \max\{j \mid (J-j)/J \geq 1-\beta, \quad j = 1, ..., J\}, \tag{6}$$

then we can estimate the β-VaR value of L as

$$\widehat{\text{VaR}}_\beta(L) = L^{(K)}. \tag{7}$$

Consider operation $a^T X^j$ in (5), it projects a sample point X^j in the support space R^n of random vector X (thus called X-space) to a direction pointed by vector a. Without loss of generality, assume a is an unit vector, i.e., $a^T a = 1$, then under such a projection, sample point $X^{(K)}$ is projected to anther point $L^{(K)} a$ lying in a ray that passes the origin and points to direction a. Let

$$\mathscr{H}_X = \{W \in R^n \mid a^T W = L^{(K)}\}, \tag{8}$$

be the hyper-plane in the X-space that has normal direction a and cuts through point $L^{(K)} a$, then $\mathscr{H}_X$ separates all sample points $X^1, ..., X^J$ in the X-space into two groups: one contains those points with $a^T X^j \geq L^{(K)}$; another contains the rest. Denote these two groups as $\mathscr{C}_X$ and $\bar{\mathscr{C}}_X$ respectively.

Note that since each V^j determines a ***single*** image X^j through a chain of maps (4) and (1), partition $\{\mathscr{C}_X, \bar{\mathscr{C}}_X\}$ in the X-space naturally induces a partition of sample points $V^1, ..., V^J$ in the support space of random vector V (thus called V-space). Denote such an induced partition in the V-space as $\{\mathscr{C}_V, \bar{\mathscr{C}}_V\}$ with $\mathscr{C}_V$ being defined as

$$\mathscr{C}_V = \{V^j \mid X^j \in \mathscr{C}_X, \quad j = 1, ..., J\},$$

i.e., $\mathscr{C}_V$ contains all V^j's whose images X^j's belong to $\mathscr{C}_X$.

In the V-space, we want to find a hyper-plane to separate two groups of points $\mathscr{C}_V$ and $\bar{\mathscr{C}}_V$. This is a classic binary classification problem. Very many different methods exist for such a problem. In the next section, we will numerically test different methods. No matter what classification method we adopt, the output is always a hyper-plane in the V-space that can be represented by

$$\mathscr{H}_V = \{W \in R^n \mid k^T W = b\},$$

where k is an unit normal vector (i.e., $k^T k$=1) of the hyper-plane and $|b|$ is the distance from the origin to the hyper-plane. Note that $\mathscr{C}_V$ and $\bar{\mathscr{C}_V}$ typically can not be completely separated by a hyper-plane in the V-space. What we are looking for is a hyper-plane that separates $\mathscr{C}_V$ and $\bar{\mathscr{C}_V}$ with minimum classification error.

Recall that V follows a standard n-dimensional normal distribution $N(0, I_n)$. We shift the center of such a distribution, which actually is the origin, to the projection point of the origin on hyper-plane $\mathscr{H}_V$, which can be represented by vector bk. In other words, we shift V by

$$\Delta V = bk,$$

and consequently shift Z by

$$\Delta Z = bD^{1/2} U^T k. \tag{9}$$

Intuitively, we shift distribution function $N(0, I_n)$ in the V-space along a direction k corresponding to (approximately) the direction pointed by a in the X-space. And the amount of shift b along direction k in the V-space is such a value that the induced amount of shift along direction a in the X-space is $L^{(K)}$ (approximately), the estimated β-VaR value of L.

After shifting ΔZ, $Z + \Delta Z$ follows a multivariate normal distribution $N(bD^{1/2}U^T k, \Sigma_Z)$. Applying transformation (2) on $Z + \Delta Z$, we get a random vector Y as defined in (2). Let $g_Y(t)$, $t \in R^n$ denote the pdf of Y, and $g_X(t)$, $t \in R^n$ denote the pdf of X, generated by un-shifted Z. We need to calculate the likelihood ratio $g_X(t)/g_Y(t)$ in order to carry out our importance sampling procedure. To that end, let $\phi_Z(t)$, $t \in R^n$ denote the joint pdf of Z, i.e.,

$$\phi_Z(\cdot) = N(0, \Sigma_Z),$$

then $g_X(t) = g_X(t_1, ..., t_n)$ can be written in terms of $\phi_Z(\cdot)$ as

$$g_X(t_1, ..., t_n) \quad = \quad \phi_Z(\Phi^{-1}(F_1(t_1)), ..., \Phi^{-1}(F_n(t_n)))|A|, \tag{10}$$

where A is the Jacobian matrix of transformation

$$Z_i = \Phi^{-1}(F_i(X_i)), \quad i = 1, ..., n,$$

and $|A|$ is the determinant of A. The above transformation is the inverse map of (1), and thus is a component-wise transformation as well. This means that A is a diagonal matrix and we can explicitly write the i-th diagonal entry of A as

$$A_{ii} = \frac{\partial \Phi^{-1}(F_i(t_i))}{\partial t_i} = \frac{f_i(t_i)}{\phi(\Phi^{-1}(F_i(t_i)))},$$

where $f_i(\cdot) = F_i'(\cdot)$ is the marginal pdf of X_i, and $\phi(\cdot) = \Phi'(\cdot)$ is the pdf of the standard normal random variable. Plug this back into equation (10), we write $g_X(\cdot)$ in terms of $\phi_Z(\cdot)$ as

$$g_X(t_1, t_2, ..., t_n) = \phi_Z(\Phi^{-1}(F_1(t_1)), ..., \Phi^{-1}(F_n(t_n))) \prod_{i=1}^{n} \frac{f_i(t_i)}{\phi(\Phi^{-1}(F_i(t_i)))}.$$

Similarly, let

$$\phi_{Z+\Delta Z}(\cdot) = N(bD^{1/2} U^T k, \Sigma_Z),$$

be the joint pdf of $Z+\Delta Z$, then $g_Y(\cdot)$ can be written as

$$g_Y(t_1,t_2,...,t_n) = \phi_{Z+\Delta Z}(\Phi^{-1}(F_1(t_1)),...,\Phi^{-1}(F_n(t_n)))\prod_{i=1}^{n}\frac{f_i(t_i)}{\phi(\Phi^{-1}(F_i(t_i)))}.$$

Therefore, the likelihood function $w(t)$ can be written as

$$w(t) = \frac{g_X(t)}{g_Y(t)} = \frac{\phi_Z(\Phi^{-1}(F_1(t_1)),...,\Phi^{-1}(F_n(t_n)))}{\phi_{Z+\Delta Z}(\Phi^{-1}(F_1(t_1)),...,\Phi^{-1}(F_n(t_n)))}.$$

This likelihood function $w(\cdot)$ will be evaluated at importance sampling points drawn from $g_Y(\cdot)$. We can either recycle existing sample points $Z^1,...,Z^J$ (used to estimate $\text{VaR}_\beta(L)$ as shown in (7)) to generate such points, or create a completely new set of sample points. The latter approach is straight-forward : we first draw points from $\mathcal{N}(bD^{1/2}U^Tk,\Sigma_Z)$, the pdf of $Z+\Delta Z$, and then apply transformation (2) to get a set of sample points of Y. To recycle $Z^1,...,Z^J$, we first shift each point Z^j by $\Delta Z = bD^{1/2}U^Tk$ as shown in (9), then apply transformation (2) on $Z^j + bD^{1/2}U^Tk$ to get points $Y^1,...,Y^J$. From now on, we assume we'll take the recycling approach to generate $Y^1,...,Y^J$ for the simplicity of presentation.

Given sample points $Y^1,...,Y^J$, the likelihood ratio $w^j = w(Y^j)$ of each point Y^j is simply

$$\begin{aligned} w^j &= \frac{\phi_Z(\Phi^{-1}(F_1(Y_1^j)),...,\Phi^{-1}(F_n(Y_n^j)))}{\phi_{Z+\Delta Z}(\Phi^{-1}(F_1(Y_1^j)),...,\Phi^{-1}(F_n(Y_n^j)))} \\ &= \frac{\phi_Z(Z^j+bD^{1/2}U^Tk)}{\phi_{Z+\Delta Z}(Z^j+bD^{1/2}U^Tk)}. \end{aligned} \tag{11}$$

Based on $Y^1,...,Y^J$, we compute the "exaggerated" empirical loss $\tilde{L}^j$ for each j as

$$\tilde{L}^j = a^TY^j, \quad j=1,...,J.$$

We then sort $\tilde{L}^1,...,\tilde{L}^J$ in an ascending order and denote the sorted sequence as $\tilde{L}^{(1)},....,\tilde{L}^{(J)}$ with $\tilde{L}^{(1)} \leq ... \leq \tilde{L}^{(J)}$. Let $w^{(j)}$ be the corresponding likelihood ratio of the j-th smallest element $\tilde{L}^{(j)}$ in the sorted sequence, we then find the largest integer S between 1 and J such that the sum of $w^{(j)}$ from S to J is greater than $J(1-\beta)$, i.e.,

$$S = \max\{s \mid \sum_{j=s}^{J} w^{(j)} \geq J(1-\beta),\ s=1,...,J\}. \tag{12}$$

Finally, β-CVaR value of the portfolio loss L is estimated as

$$\widehat{\text{CVaR}_\beta}(L) = \left(\sum_{j=S}^{J} w^{(j)}a^TY^{(j)}\right) \Big/ \left(\sum_{j=S}^{J} w^{(j)}\right). \tag{13}$$

Equations (12) and (13) warrant some further explanations. Compare equations (12) and (6), we can see that w^j/J can be interpreted as the "real probability" to generate Y^j if it were drawn from $g_X(\cdot)$, the pdf of X. Since we actually generate Y^j from $g_Y(\cdot)$, each sample Y^j needs to be scaled by w^j to get the "real probability" as if they were generated from $f_X(\cdot)$.

We summarize our method as follows

1. Draw i.i.d. sample points $V^1,...,V^J$ from $N(0,I_n)$, get $Z^1,...,Z^J$ through transformation (4), and get $X^1,...,X^J$ through transformation (1).
2. Calculate $L^1,...,L^J$ as in (5), and estimate $\text{VaR}_\beta(L)$ as in (7).

3. Divide sample points $X^1, \ldots, X^J$ into two groups $\mathscr{C}_X$ and $\bar{\mathscr{C}}_X$ in the X-space, and find the corresponding partition $\mathscr{C}_V$ and $\bar{\mathscr{C}}_V$ in the V-space.
4. Find a hyper-plane that separates $\mathscr{C}_V$ and $\bar{\mathscr{C}}_V$ in the V-space and output parameters k and b.
5. compute ΔZ as in (9).
6. generate Y^j by applying transformation (2) to $Z^j + \Delta Z$.
7. compute w^j using (11), and evaluate $\text{CVaR}_\beta(L)$ as shown in (13).

3 COMPUTATIONAL EXPERIMENTS

In this section, we study the computational performance of our importance sampling algorithm on a set of numerical experiments implemented in MATLAB. An instance of a numerical experiment is fully specified in terms of:

1. a set of n marginal densities of $X_1, X_2, \ldots, X_n$;
2. a $n \times n$ correlation matrix Σ_Z ;
3. a vector of n weights $a_1, a_2, \ldots, a_n$;
4. the number of sample points J.

The goal in all experiments below is to estimate CVaR at the 95-percentile of the random variable $L = \sum_{i=1}^{n} a_i X_i$. We use $J = 1000$ in all experiments. A *run* reported in each of the tables below is a set of 100 experiments that we conduct in order to estimate the variance of the CVaR estimator. In all the following results (tables), the percentage of variance reduction in CVaR estimate is computed relative to the variance in CVaR estimate obtained from plain simulation using the same number of samples.

3.1 An illustrative example in two dimensions

We first provide an illustration of our algorithm on a simple two-dimensional instance with the following input data.

Table 1: A two-dimensional example.

Variable	Marginal density	Weight (a)
X_1	`normal(10, 5)`	5
X_2	`exponential (0.9)`	25

The correlation matrix, Σ_Z is taken to be:

$$\Sigma_Z = \begin{bmatrix} 1 & 0.5428 \\ 0.5428 & 1 \end{bmatrix}$$

We tried two variants of the linear classifier in our algorithm, namely linear classification using Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). We used the MATLAB function call `classify` with default settings for the LDA classifier, and LIBSVM (Chang and Lin 2001) with default settings for the SVM classifier. Figure 1 shows the LDA classifier in the Z-space, and Figure 2 shows the same LDA classifier the V-space. Similarly, Figure 3 shows the SVM classifier in the Z-space, and Figure 4 shows the same SVM classifier in the V-space. Note that the SVM classifier pushes the linear decision boundary to the farthest possible separation of the two sample-specific classes, while the LDA classifier is less extreme with respect to the position of the linear boundary (note the number of red points on the blue side of the linear boundary in Figure 1, or Figure 2). Table 2 shows the results for the above example for a *run* across 100 experiments. Both methods lead to variance reduction and the LDA-based method performs better than the SVM-based method for this simple example.

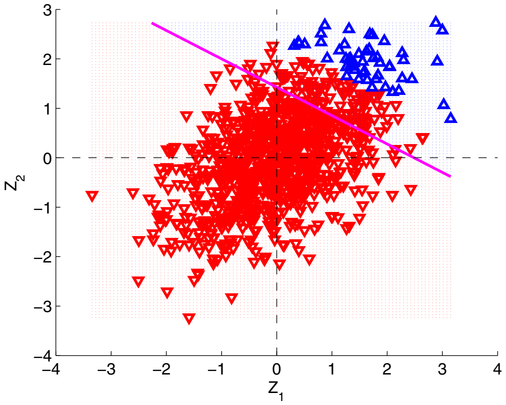

Figure 1: Importance Sampling Shift in the Z-Space using LDA classifier.

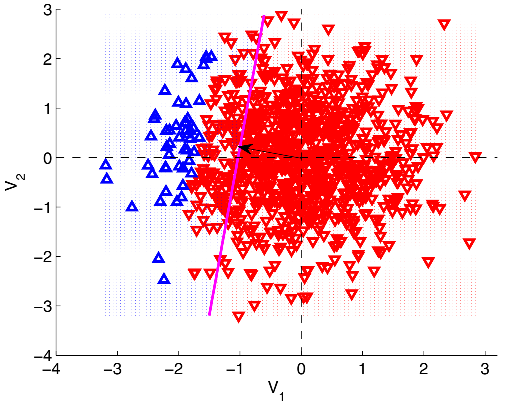

Figure 2: Importance Sampling Shift in the V-Space using LDA classifier.

3.2 The special case where all marginal densities are normal

We next consider the special case where all the marginal densities are normal distributions. In other words, we seek to estimate $\mathrm{CVaR}_\beta(L)$, where $L = \sum_{i=1}^{n} a_i X_i$ and each X_i has a marginal density that is normal, say, $X_i \sim N(\mu_i, \sigma_i)$, each with mean μ_i and standard deviation σ_i. The correlation matrix is ρ_X. The analytical expression for CVaR in this case (for $\beta \geq 0.5$) is given as (Rockafellar and Uryasev 2000),

$$\mathrm{CVaR}_\beta(L) = a^T\mu + \frac{\sqrt{a^T S a}}{\sqrt{2\pi}(1-\beta)\exp\left[\mathrm{erf}^{-1}(2\beta-1)\right]^2} \tag{14}$$

In Equation(14), $S = (\sigma\sigma^T) \bullet \rho_X$ is the covariance matrix corresponding to the multivariate Gaussian random vector X (and the operator $\bullet$ refers to element-by-element multiplication) and erf^{-1} refers to the inverse of error function $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$. This is an interesting case because it allows us to benchmark the CVaR estimate produced by our algorithm against the analytical CVaR. We tested the algorithm for $n = 50$, where all fifty marginal densities were fixed to be normal densities with arbitrarily chosen means and standard deviations. For simplicity, we chose the weighting vector a to be a vector of ones, i.e. $a_i = 1$, for all i. We report the performance of the algorithm when run

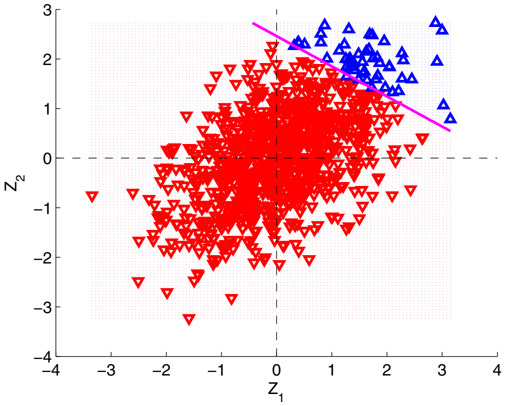

Figure 3: Importance Sampling Shift in the Z-Space using SVM classifier.

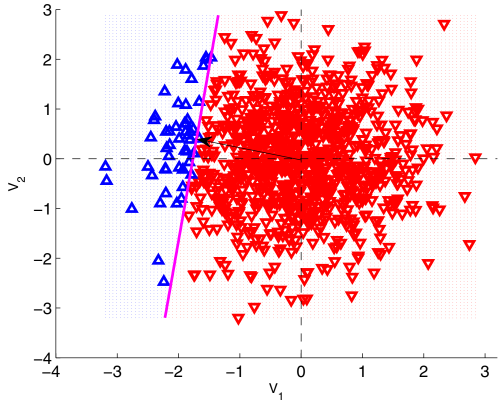

Figure 4: Importance Sampling Shift in the V-Space using SVM classifier.

against a set of 5 randomly chosen correlation matrices in Table 3 (parameters of the 50 marginal densities and the five correlation matrices not shown due to space considerations).

The percentage of CVaR estimation error in Table 4 is computed relative to the corresponding true value of CVaR computed using Equation (14). It can be seen that the LDA-based variant of our algorithm gives better performance in terms of variance reduction in all five instances. The CVaR estimation error is also the least for the LDA-based method.

3.3 A mixture of normal and exponential densities

We conclude the experimentation with the more general case that involves a combination of normal and exponential densities. We test on a reasonably large sized set of instances with $n = 100$, where 50 variables have normal marginal densities and the remaining 50 variables have exponential marginal densities. As in the previous subsection, the parameters for specifying the above 100 marginal densities were arbitrarily chosen, and the weighting vector was fixed at unity. We report the performance of our algorithm when run against a set of 5 randomly chosen correlation matrices in Table 5 (again, parameters of the 100 marginal densities and the five correlation matrices not shown). In Table 5, the column labeled "CVaR*" is a higher-fidelity CVaR estimate that is obtained using plain simulation with 8000 samples, and repeating across 100 such experiments. All experiments were conducted on a Lenovo ThinkPad T60p, with an Intel CPU, T2600 at 2.16 GHz, and 2 GB RAM. Our MATLAB implementation could not simulate more than 8000 samples on the 100-dimensional space due to memory limitations.

Table 2: Results for the two-dimensional example.

Method	CVaR Estimate	Variance	Variance Reduction
Plain	173.91	20.70	-
SVM-based	175.67	13.47	34.93%
LDA-based	175.01	6.56	68.28%

Table 3: Results for the case of 50 normal variables.

Run	CVaR (Plain)	Var (Plain)	CVaR (SVM)	Var (SVM)	CVaR (LDA)	Var (LDA)	True CVaR
1	1895.89	333.27	1897.46	210.47	1897.60	88.57	1898.07
2	1963.26	550.29	1965.51	234.93	1963.94	133.71	1964.83
3	1781.84	208.31	1782.77	94.78	1782.68	57.42	1782.37
4	1834.89	284.22	1839.64	172.72	1838.94	71.06	1838.74
5	1818.11	282.90	1823.12	103.20	1822.23	73.45	1822.44

Table 4: CVaR Estimation error (%) and Variance Reduction (%) for the case of 50 normal variables.

Run	CVaR (Plain)	Var (Plain)	CVaR (SVM)	Var (SVM)	CVaR (LDA)	Var (LDA)
1	-0.11%	-	-0.03%	36.85%	-0.02%	73.42%
2	-0.08%	-	0.03%	57.31%	-0.05%	75.70%
3	-0.03%	-	0.02%	54.50%	0.02%	72.43%
4	-0.21%	-	0.05%	39.23%	0.01%	75.00%
5	-0.24%	-	0.04%	63.52%	-0.01%	74.03%

The percentage of CVaR estimation error in Table 6 is computed relative to CVaR*. It can be seen that for the more general case the SVM-based method gives an overall better performance in terms of both variance reduction and quality of the mean CVaR estimate. The CVaR estimation error is the least for the SVM-based method. The extent of variance reduction obtained from the SVM-based method is comparable to that obtained from the LDA-based method, and both variants lead to greater than 50% reduction in the variance of CVaR estimate. The LDA based method overestimates the CVaR by around 1-1.25%, compared against the higher-fidelity plain simulation estimate.

We conclude our preliminary numerical investigation with the observation that the proposed importance-sampling algorithm appears to be effective in terms of variance reduction in estimating CVaR for weighted sums of correlated random variables. Both variants of the algorithm, namely the LDA-based and the SVM-based methods, are easy to implement with relatively minor computational overhead. The performance of the SVM-based method seems to be more competitive in the more general case that involves non-normal marginal densities. We speculate that there might be an optimal shift, i.e. an optimal positioning of the linear decision boundary that is somewhere in between the LDA-based and the SVM-based decision boundaries, and one may need to tune this for any given application.

4 CONCLUSIONS

We have developed an importance sampling procedure to estimate CVaR with a general Gaussian copula model. Our method has an intuitive geometric explanation. It is fairly easy to implement and involves minimal computational effort. Numerical tests have confirmed its superior performance compared to the naive approach. This work may be extended further in two directions: first, investigate our algorithm from a theoretical perspective and characterize its asymptotic performance; second, devise an method to find the “optimal” shift for arbitrary marginal densities and Gaussian copula correlation matrix.

Table 5: Results for the more general case of 100 mixed variables. CVaR* is an estimate of CVaR with 8000 samples in each of the 100 experiments

Run	CVaR (Plain)	Var (Plain)	CVaR (SVM)	Var (SVM)	CVaR (LDA)	Var (LDA)	CVaR*
1	2072.24	471.51	2075.91	196.46	2102.31	161.23	2076.30
2	2074.90	352.96	2075.14	174.51	2099.95	178.73	2077.31
3	2023.31	297.24	2026.58	141.55	2048.96	146.88	2028.91
4	2090.90	423.76	2094.42	177.08	2121.14	167.35	2096.10
5	2120.59	506.81	2123.92	214.03	2151.42	193.10	2124.37

Table 6: CVaR Estimation error (%) and Variance Reduction (%) for the case of 100 mixed variables.

Run	CVaR (Plain)	Var (Plain)	CVaR (SVM)	Var (SVM)	CVaR (LDA)	Var (LDA)
1	-0.20%	-	-0.02%	58.33%	1.25%	65.80%
2	-0.12%	-	-0.10%	50.56%	1.09%	49.36%
3	-0.28%	-	-0.11%	52.38%	0.99%	50.59%
4	-0.25%	-	-0.08%	58.21%	1.19%	60.51%
5	-0.18%	-	-0.02%	57.77%	1.27%	61.90%

REFERENCES

Artzner, P., F. Delbaen, J.-M. Eber, and D. Heath. 1999. Coherent measures of risk. *Mathematical Finance* 9:203–228.

Cario, M. C., and B. L. Nelson.. 1997. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

Chang, C. C., and C. J. Lin. 2001. Libsvm: a library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Follmer, H., and A. Schied. 2002. Convex measures of risk and trading constraints. *Finance and Stochastics* 6:429–447.

Glasserman, P., W. Kang, and P. Shahabuddin. 2008. Fast simulation of multifactor portfolio credit risk. *Operations Research* 56:1200–1217.

Glasserman, P., and J. Li. 2005. Importance sampling for portfolio credit risk. *Management Science* 51:1643–1656.

Manistre, B. J., and G. H. Hancock. 2005. Variance of the cte estimator. *North American Actuarial Journal* 9:129–156.

Merino, S., and M. A. Nyfeler. 2004. Applying importance sampling for estimating coherent credit risk contributions. *Quantitative Finance* 4:199–207.

Rockafellar, R. T., and S. Uryasev. 2000. Optimization of conditional value-at-risk. *The Journal of Risk* 2:21–41.

AUTHOR BIOGRAPHIES

PU HUANG is a Research Staff Member in the Business Analytics & Mathematical Sciences Department at the IBM T.J. Watson Research Center. His research interests include stochastic optimization, risk analysis and data mining. In the past few years, he has been working closely with various IBM organizations to develop analytical solutions for resource allocation and risk management applications. Dr. Huang got his Ph.D. in Operations Research from the Carnegie Mellon University. His email is puhuang@us.ibm.com.

DHARMASHANKAR SUBRAMANIAN is a Research Staff Member in the Business Analytics & Mathematical Sciences Department at the IBM T.J. Watson Research Center. His interests and expertise include mathematical modeling, mathematical programming with risk measures for stochastic optimization, simulation and risk analysis. At IBM, his specific interests include operational risk modeling, product development portfolio analysis and stochastic optimization with risk measures.

Prior to joining IBM Research, he was a Senior Research Scientist at Honeywell Labs in Minneapolis, where he conducted basic and applied research activities for a diverse set of domains, including supply-side energy portfolio optimization for optimal utility contracts, cogeneration-based integrated energy systems, aircraft trajectory optimization, wireless sensor networks, and nonlinear process control. He holds a Ph.D. in Chemical Engineering from Purdue University, where he focused on combinatorial decision-making under uncertainty. His e-mail address is dharmash@us.ibm.com>.

JIE XU obtained his Ph.D. in Industrial Engineering and Management Sciences from Northwestern University in Evanston, IL and a MS degree in Computer Science from State University of New York at Buffalo. His research interests includes simulation-based optimization, variance reduction techniques for Monte Carlo simulation, revenue management and risk management.