Supplementary Materials for 'Selective bivariate copula models using image recognition', published in *ASTIN Bulletin*

Andreas Tsanakas and Rui Zhu

Bayes Business School, City, University of London

A Details of PCA, LDA and SVM

PCA To find the principal component (PC) subspace, we can apply the reduced singular value decomposition (SVD) on the column-centred $\mathbf{X}^M$:

$$(\mathbf{X}^M)^c = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{A.1}$$

where $(\mathbf{X}^M)^c \in \mathbb{R}^{N\times 4096}$ is the column-centred $\mathbf{X}^M$ derived by extracting column means from $\mathbf{X}^M$, $\mathbf{U} \in \mathbb{R}^{N\times r}$ and $\mathbf{V} \in \mathbb{R}^{4096\times r}$ contain left and right singular vectors, $\mathbf{D} \in \mathbb{R}^{r\times r}$ is a diagonal matrix with singular values $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_r \geqslant 0$. The first q ($q \leqslant r$) columns in $\mathbf{V}$, i.e. the first few PCs, are selected to construct the PC subspace.

LDA LDA finds the discriminative subspace by solving the following optimisation problem:

$$\max_{\mathbf{W}} \frac{\det(\mathbf{W}^T\mathbf{S}_B\mathbf{W})}{\det(\mathbf{W}^T\mathbf{S}_W\mathbf{W})}, \tag{A.2}$$

with $\mathbf{S}_W = \sum_{k=1}^{K} \sum_{i \text{ in class } k} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$ and $\mathbf{S}_B = \sum_{k=1}^{K} N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$. Here K is the number of classes, $\mathbf{W} \in \mathbb{R}^{153\times(K-1)}$ contains the bases of the linear discriminant subspace, $\mathbf{x}_i \in \mathbb{R}^{153\times 1}$ denotes each sample in $\mathbf{X}$, $\boldsymbol{\mu}_k \in \mathbb{R}^{153\times 1}$ is the class mean of the k-th class and $\boldsymbol{\mu} \in \mathbb{R}^{153\times 1}$ is the overall mean of $\mathbf{X}$. As the optimisation problem (A.2) involves class information, $\mathbf{W}$ summarises the discriminative information between classes.

SVM SVM aims to find a separating hyperplane $f(\mathbf{x}) = \phi(\mathbf{x}_i^P)^T\mathbf{w} + b$ for classification by maximising the margin M between two classes:

$$\begin{aligned} &\max_{\mathbf{w},b} \ M \\ \text{s.t. } &m_i(\phi(\mathbf{x}_i^P)^T\mathbf{w} + b) \geqslant M(1-\psi_i) \ \ \forall i, \\ &\psi_i \geqslant 0 \ \ \forall i, \ \ \sum_{i=1}^{N} \psi_i \leqslant C, \end{aligned} \tag{A.3}$$

where $M = 1/||\mathbf{w}||_2$ is the shortest distance from the training sample to the classification boundary, $\mathbf{w}$ and b defines the separating hyperplane, $\phi(\cdot)$ is a function that projects $\mathbf{x}_i^P$ to a reproducing kernel Hilbert space, ψ_i is the slack variable that allows violations of the training observations to the margins, and C is a predefined positive integer that controls the trade-off between the goodness-of-fit of the training set to the classifier and the generalisation ability of the classifier on unseen data. The solutions $\mathbf{w}^*$ and b^* are then used to classify a test observation $\mathbf{x}$: if $f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w}^* + b^*$ is positive, then $\mathbf{x}$ belongs to the positive class; otherwise, $\mathbf{x}$ belongs to the negative class.

B Sensitivity analyses

B.1 Robustness tests for the image recognition approach of Section 3

Here we summarise the results of two robustness checks, seeking to evaluate the extent to which classification performance is impacted by potentially arbitrary decisions in the design of our copula selection process.

First, in order to generate heatmaps, a choice of marginal distribution is necessary. (As we are only investigating dependence effects, this choice does not reflect any assumption regarding the marginal distributions of the actual data one may be modelling; in a sense, it is a hyperparameter choice). So far, all heatmap images are generated from bivariate samples with Normal margins. Here, we additionally consider Cauchy, Laplace and Uniform margins. For the $R = 20,000$ bivariate copula samples we generated with $n = 250$ and $\tau = 0.5$, we produce heatmaps using each of those additional margins by slightly modifying the process described in Section 3.1. Subsequently, we extract features and train a SVM to classify those heatmaps in the case of each margin, following the same process and experiment settings as in Section 3.3. The results are summarised in Figure B.1. Clearly, the Normal and Laplace margins have very similar medians and interquartile ranges, indicating that classification performance is similar for those two marginal choices. The Cauchy and Uniform margins show worse classification accuracies,

with medians lower than those of Normal and Laplace margins by around 1.5%. This shows that the choice of margin has a noticeable effect on the final results and that the choice of a Normal margin proved to be a beneficial one.

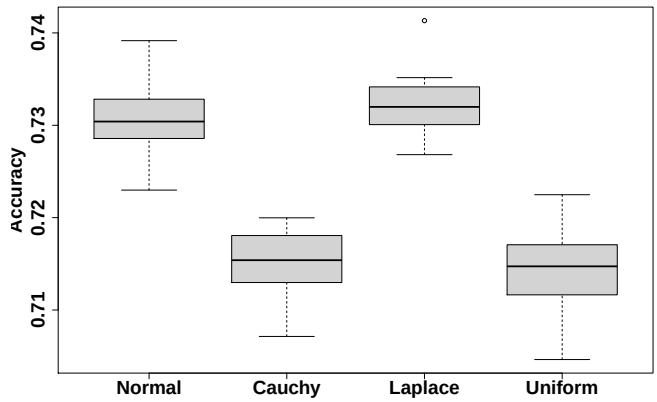

Figure B.1: The classification accuracies for different margins with $n = 250$ and $\tau = 0.5$.

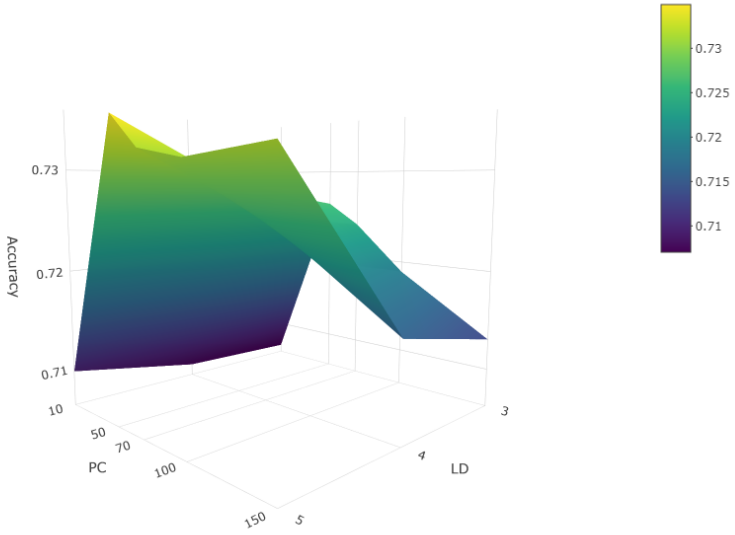

Figure B.2: Classification accuracies for different dimensions of the PC and LD subspaces to classify copula samples with $n = 250$ and $\tau = 0.5$.

Second, the dimensions of the PC and LD subspaces can affect the final classification performance, because they determine the amount of information to be included in the low-dimensional subspaces. Setting the dimensions to small numbers may result in low classification accuracies because of the loss of vital information for classification, while setting them to large numbers

close to the original feature dimensions fails to achieve dimension reduction. In Figure B.2, we show the surface curve of the classification accuracies for different dimensions of the PC and LD subspaces to classify copula samples with $n = 250$ and $\tau = 0.5$. Five dimensions of the PC subspace are tested, {10,50,70,100,150}, while three dimensions of the LD subspace are tested, {3,4,5}. As expected, when the dimensions of the subspaces are low, e.g. the dimension of the PC subspace is 10 and that of the LD subspace is 3 or 4, the classification accuracies are just around 71%. However, when the dimension of the PC subspace is higher than 50 and that of the LD subspace is set to the maximum number of five, we can observe the highest classification accuracies of more than 73%. These results demonstrate that our choices of 150-dimensional PC subspace and 5-dimensional LD subspace are sensible.

B.2 The impact of the statistical and image features on the classification performance

In the image recognition approach, the statistical and image features are combined to provide a more complete description of copula samples. Here we produce results for more covariate combinations, in order to give a fuller picture on the relative contributions of statistical and image features to prediction accuracy. Specifically, we aim to explore the impact of the statistical features and image features on the classification performance separately. The training and test sets with $n \in [100, 250]$ of Section 4.2.2 are used here.

The following three settings are experimented. First, only three statistics, Kendall's rank correlation, skewness and arachnitude, are extracted from the training set to train the SVM classifier. No image features are used. The dimension reduction process of LDA is ignored in this case, because the dataset is only three-dimensional. Second, we add four additional statistics to the first setting, such that we have in total seven statistics, providing a fuller description of the copula samples. Specifically, we use the empirical tail probabilities $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{A_i}$, where A_i is one of $\{u_i \leqslant 0.05, v_i \leqslant 0.05\}$, $\{u_i \leqslant 0.25, v_i \leqslant 0.25\}$, $\{u_i \geqslant 0.75, v_i \geqslant 0.75\}$ or $\{u_i \geqslant 0.95, v_i \geqslant 0.95\}$. Third, only the high-dimensional AlexNet image features are extracted from the training set and fed to PCA and LDA for dimension reduction. Thus no statistical features are used. The image features are used to train the SVM classifier.

Table B.1: Test classification accuracies of the three settings.

	Three statistics	Seven statistics	Image features only
$n \in [100, 250]$	0.5113	0.5367	**0.5793**

From Table B.1, we can observe that involving more statistics can improve the classification performance, compared to the baseline case of using three statistics. However, using image

features only produces clearly higher accuracy compared to using statistical features only. Comparing with Table 1, we further note that using image features only (accuracy: 0.5793) still dominates AIC (accuracy: 0.5688). Furthermore, the classification accuracies in Table B.1 are all lower than those of the image recognition approach of Table 1, which combines image and statistical features. Taken together, these results demonstrate that image features contribute important information that is not captured by statistical features and that a combination of statistical and image features is best for selecting a copula model.

B.3 Sensitivity analysis of the image recognition algorithm of Section 4.2.2

We adapt the scenario weighting and reverse sensitivity framework developed by Pesenti et al. (2019), in the context of stress testing simulation models and implemented in the **R** package *SWIM* (Pesenti et al., 2021). This framework is well suited to situations where it is cumbersome or computationally expensive to repeatedly evaluate the prediction function on new observations.

We apply the sensitivity analysis on the test set, with $\mathbf{x}_t \in \mathbb{R}^{153}$ the feature vector of the tth sampling instance, for $t = 1, \dots, S$, where $S = 10,000$. Furthermore, for each testing instance we also consider the vector $\mathbf{y}_t \in \mathbb{R}^6$, where $y_{t,l}$ represents the number of votes obtained by the lth copula model as part of the majority voting procedure described in Section 3.2. Then, for each model $l = 1, \dots, 6$ we calculate a vector of weights in $\mathbb{R}^S$, such that, under re-weighting the sample $y_{1,l}, \dots, y_{S,l}$, the average number of votes for this model increases by 1. The vector of weights is selected by minimising the Kullback-Leibler divergence; specifically we solve the problem:

$$\begin{cases} \min_{\mathbf{w}\in\mathbb{R}^S} \frac{1}{S}\sum_{t=1}^{S} w_t \log(w_t) \quad \text{s. t.} \\ w_t > 0, \ t = 1, \dots, S \\ \frac{1}{S}\sum_{t=1}^{S} w_t = 1 \\ \frac{1}{S}\sum_{t=1}^{S} w_t y_{t,l} = \frac{1}{S}\sum_{t=1}^{S} y_{t,l} + 1. \end{cases}$$

The solution $\mathbf{w}^* \in \mathbb{R}^S$ (derived originally by Csiszár (1975)) applies a higher weight to those testing instances that drive the increase in the average vote for model l. Subsequently a sensitivity index for the ith feature can be defined as the normalised increase in the average of $x_{t,j}$, $t = 1, \dots, S$, $j = 1, \dots, 153$, over instances, arising from weighting by $\mathbf{w}^*$.

The results of this analysis are shown in Figure B.3, which plots the sensitivity of the majority vote for each of the models in $\mathcal{M}$ to the first 10 principal components of the heatmap images, as well as the statistical features $\hat{\tau}$ (tau) $\hat{\zeta}$ (skew), $\hat{\xi}$ (arach). It can be seen that the sensitivity to skewness $\hat{\zeta}$ is important for radially symmetric models (with a negative effect)

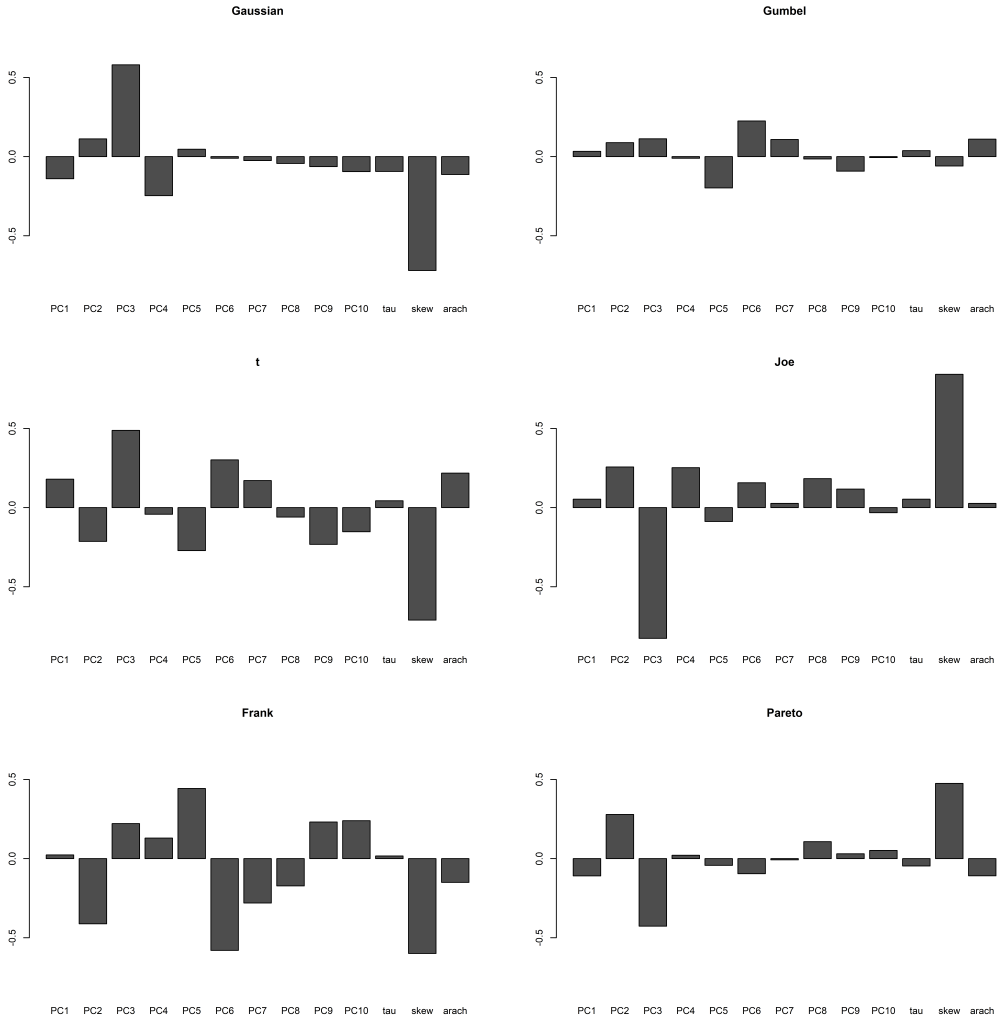

Figure B.3: Sensitivity of majority vote for each copula model, to different features.

and for the Pareto and Joe models (with a positive effect), consistently with the properties of these copulas. Beyond that, the main role in telling apart the different models is played by the image principal components; e.g. we can note the quite different patterns of PC1-PC10, for the 3 radially symmetric models on the left of the plot. On the other hand, for the Joe and Pareto models, which cannot be easily distinguished by the classifier, the PC patterns are rather similar.

References

Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability* 3(1), 146-158.

Pesenti, S. M., Bettini, A., Millossovich, P., Tsanakas, A. (2021). Scenario Weights for Importance Measurement (SWIM)–an R package for sensitivity analysis. *Annals of Actuarial Science* 15(2), 458-483.

Pesenti, S. M., Millossovich, P., Tsanakas, A. (2019). Reverse sensitivity testing: What does it take to break the model? *European Journal of Operational Research* 274(2), 654-670.

C Algorithms

Algorithm 1: Algorithm for generating the image dataset, with given fixed n, τ (Section 3.1).

$i \leftarrow 0$;
while $i < R$ **do**
  Choose randomly copula family $C^{(m)},\ m \in \mathcal{M}$;
  if m *is the* t *copula* **then**
    Choose randomly degrees of freedom ν from $\{3, 4, \ldots, 10\}$;
    Work out parameters θ from (τ, ν);
  else
    Work out parameter θ from τ;
  end
  Simulate n pairs of observations from $(U, V) \sim C^{(m)}(\cdot; \theta)$;
  Transform simulated observations to pseudo-observations $(u_j, v_j)_{j=1,\ldots,n}$;
  From $(u_j, v_j)_{j=1,\ldots,n}$ calculate sample statistics $\hat{\tau}$, $\hat{\zeta}$, $\hat{\xi}$;
  if $\hat{\tau} \geqslant 0$ **then**
    if $\hat{\zeta} \geqslant 0$ **then**
      $KeepData \leftarrow$ *TRUE*
    else
      if $m \in \mathcal{M}_s$ **then**
        Rotate pseudo-observations, $u_j \leftarrow 1 - u_j,\ v_j \leftarrow 1 - v_j,\ j = 1, \ldots, n$;
        $\hat{\zeta} \leftarrow -\hat{\zeta}$;
        $KeepData \leftarrow$ *TRUE*;
      else
        $KeepData \leftarrow$ *FALSE*;
      end
    end
  else
    $KeepData \leftarrow$ *FALSE*
  end
  if $KeepData =$*TRUE* **then**
    $i \leftarrow i + 1$;
    Calculate $\mathrm{AIC}^{(l)}$ from $(u_j, v_j)_{j=1,\ldots,n}$, for each $l \in \mathcal{M}$;
    Save $\hat{\tau}$, $\hat{\zeta}$, $\hat{\xi}$, $\mathrm{AIC}^{(l)}$;
    Transform pseudo-observations to normal $x_j \leftarrow \Phi^{-1}(u_j),\ y_j \leftarrow \Phi^{-1}(v_j),\ j = 1, \ldots, n$;
    Estimate joint density of (x_j, y_j). Create heatmap and save image;
  end
end

Algorithm 2: Algorithm for generating the test set, with variable n, τ and model rotations (Section 4.1).

$i \leftarrow 0$;
while $i < S$ **do**
  Choose randomly $m \in \mathcal{M}$, $n \in [n_1, n_2]$, $\tau \in [\tau_1, \tau_2]$;
  if m *is the* t *copula* **then**
    Choose randomly degrees of freedom ν from $\{3, 4, \ldots, 10\}$;
    Work out parameters θ from (τ, ν);
  else
    Work out parameter θ from τ;
  end
  Choose randomly $r \in \{0, 90, 180, 270\}$;
  if $r \in \{0, 180\}$ **then**
    $\tau_r \leftarrow \tau$;
  else
    $\tau_r \leftarrow -\tau$;
  end
  Simulate n pairs of observations from $(U, V) \sim C^{(m_r)}(\cdot; \theta)$;
  Transform simulated observations to pseudo-observations $(u_j, v_j)_{j=1,\ldots,n}$;
  Calculate $\mathrm{AIC}^{(l)}$, $l \in \mathcal{M}'$;
  Save m, r, τ_r, $(u_j, v_j)_{j=1,\ldots,n}$, and $\mathrm{AIC}^{(l)}$, $l \in \mathcal{M}'$;
  $i \leftarrow i + 1$;
end

Algorithm 3: Algorithm for generating heatmap images from the test set, with variable fixed n, τ and model rotations. (First step of Section 4.2.2.)

$i \leftarrow 0$;
while $i < S$ **do**
  Read pseudo-observations $(u_j, v_j)_{j=1,\dots,n}$, underlying model with (m, r) such that $m_r \in \mathcal{M}'$, and rank correlation τ_r, from the ith instance of the test set;
  Calculate $\hat{\tau}$ from $(u_j, v_j)_{j=1,\dots,n}$;
  Initialise the degree to which data will be rotated, $s \leftarrow 0$;
  if $\hat{\tau} < 0$ **then**
    $s \leftarrow s + 90$;
    $v_j \leftarrow 1 - v_j,\ j = 1, \dots, n$;
    $\hat{\tau}_s \leftarrow -\hat{\tau}$;
  end
  Calculate $\hat{\zeta}$ from $(u_j, v_j)_{j=1,\dots,n}$;
  if $\hat{\zeta} < 0$ **then**
    $s \leftarrow s + 180$;
    $u_j \leftarrow 1 - u_j,\ v_j \leftarrow 1 - v_j,\ j = 1, \dots, n$;
    $\hat{\zeta}_s \leftarrow -\hat{\zeta}$;
  end
  Estimate the copula rotation $\hat{r} \leftarrow 360 - s$;
  Calculate from $(u_j, v_j)_{j=1,\dots,n}$, $\hat{\xi}_s$ and $\mathrm{AIC}_s^{(l)}$ for each $l \in \mathcal{M}$;
  Save $\hat{\tau}_s$, $\hat{\zeta}_s$, $\hat{\xi}_s$, $\mathrm{AIC}_s^{(l)}$;
  Transform pseudo-observations to normal $x_j \leftarrow \Phi^{-1}(u_j),\ y_j \leftarrow \Phi^{-1}(v_j),\ j = 1, \dots, n$;
  Estimate joint density of (x_j, y_j). Create heatmap and save image;
  if $m \in \mathcal{M}_s$ **then**
    if $sign(\hat{\tau}) = sign(\tau_r)$ **then**
      $FirstStep \leftarrow$ *TRUE*
    else
      $FirstStep \leftarrow$ *FALSE*
    end
  end
  if $m \in \mathcal{M}_a$ **then**
    if $\hat{r} = r$ **then**
      $FirstStep \leftarrow$ *TRUE*
    else
      $FirstStep \leftarrow$ *FALSE*
    end
  end
  $i \leftarrow i + 1$;
end