Predicting Image Positions using Graphical Models

Report – Advanced Topics in Decision Theory (WI4139)

Volker Strobel (4524187)

June 23, 2016

Abstract

In this report, the dependence structure between color features of image patches and their x, y-coordinates in a larger map image is analyzed using the linear least squares predictor, graphical Gaussian models, and a vine copula. The dependencies are used to build predictive models: given the mean red, green, and blue value of a patch, the x, y-coordinates should be predicted. The models are compared in terms of their goodness of fit and their predictive power on unseen data.

1 Introduction & Problem Statement

Computer vision tasks, such as object detection, image restoration, or localization, often require high-dimensional feature spaces for adequate representations of the underlying problem. In this report, the following computer vision problem is analyzed:

> At what x, y-coordinate is a given image patch located in a larger map image?

While existing approaches extract keypoints of the current patch and the map image, followed by finding a homography between both keypoint sets, these approaches are usually computationally complex and do not allow for in-depth analyses of the problem (e.g. how to change the map image to improve the fit of the model). In contrast, graphical models are convenient tools for modeling dependence in the joint distribution of the image features.

The goal of this report is to build models from data and compare their ability to predict the x, y-coordinate of unseen image patches. To this end, the linear least squares predictor, graphical Gaussian models, and a vine copula are compared. Using the linear least squared predictor will set the stage for the graphical models. By using graphical Gaussian models insights into the dependence structure of the variables should be obtained. While graphical Gaussian models assume that the data are joint normal, the use of vine copulas provides a more flexible modeling technique and allows for modeling marginal distributions and the dependence structure independently.

The remainder of this report is structured as follows. Section 2 introduces the method for generating the dataset, and for inferring the parameters of the linear least square predictor, the graphical Gaussian model, and the vine copula. Section 3 presents the obtained results and error statistics. In Section 4, the results are discussed, models contrasted and directions of future research are given.

2 Methods

2.1 Dataset Generation

For generating the dataset, a suitable map image of size width × height = 640×480 pixels was chosen. Since the color values of pixels and their x, y-coordinate should be correlated the search term 'rainbow art' was used in Google's image search. The selected image can be seen in Figure 1a. In order to obtain image patches from a given map image, $N = 1000$ different views of the map image were generated (see Figure 1b for an example) using the tool *draug*[1]. The x, y-positions of the image patch centers (represented by the random variables POS_x and POS_y) were sampled from a normal distribution with the following parameters:

$$(POS_x, POS_y) \sim \mathcal{N}(\mu, \Sigma) \tag{1}$$

$$\mu = \begin{pmatrix} 320 \\ 240 \end{pmatrix} \tag{2}$$

$$\Sigma = \begin{pmatrix} 107 & 0 \\ 0 & 80 \end{pmatrix} \tag{3}$$

Therefore, x and y positions were sampled independently. The image patches are 1/10 in size of the map image. The standard deviations of POS_x, POS_y were chosen, such that ca. 99 % of the drawn samples are expected to be in the ranges $[0, 640]$ and $[0, 480]$, respectively for x, and y values. The dataset is split into two parts, such that 500 images are used as training images (construction of the models) and 500 as test images (evaluation of the predictive power of the models).

Three features per image patch are used: the average red, green, and blue color. By construction of the dataset, it is known that the samples are identically and independently distributed (i.i.d.). For demonstration purposes, the first 20 values of the resulting dataset can be found in the appendix. The full dataset can be obtained online[2].

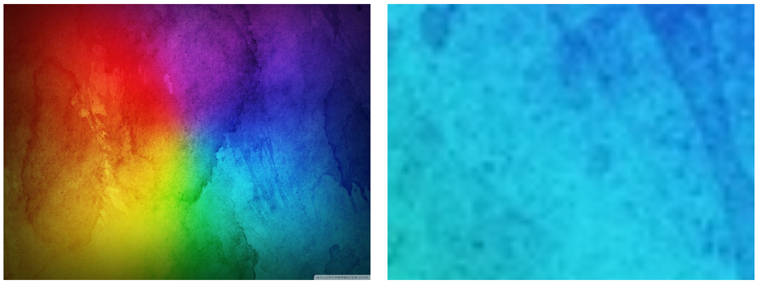
2.2 Test for Normality

For motivating the use of graphical Gaussian models, the Shapiro-Wilk test was used (significance level $\alpha = 0.05$), which tests if the univariate marginal distributions are normally distributed.

If variables were non-normal in the graphical Gaussian model, they were transformed to uniform distributions by converting them to pseudo-observations, and then to normal by applying the inverse standard normal cdf. The transformed dataset was tested for multivariate normality using Henze-Zirkler's test

[1] draug is a tool that I developed during my graduation project. It generates many different views of one single image using a pinhole camera model and parameters for viewing angles. Source code: `https://github.com/Pold87/draug`

[2] `https://github.com/Pold87/decision-theory`



(a) Map image

(b) Random image patch

Figure 1: Figure 1a Shows the map image that was used for generating the image patches. Figure 1b shows one sample of the $N = 1000$ image patches that were generated.

with the null hypothesis that the data is joint normal (significance level $\alpha = 0.05$).

2.3 Linear Least Squares Predictor

The used random vectors are $Y = (POS_x, POS_y)$ (position of the image patch) and $X = (R, G, B)$ (mean red, green, and blue value of the image patch). Given a measurement of the mean color values $X = (r, g, b)^T$, the linear least squares predictor was calculated as follows:

$$\hat{Y}(X) = EY + \operatorname{cov}(Y, X)\operatorname{var}(X)^{-1}((r, g, b)^T - EX)$$

To determine the goodness of fit, the explained variation R^2 for the dependent variables POS_x and POS_y was calculated. The regression coefficients were tested for significance in the model (significance level $\alpha = 0.05$) using the t-statistic with $N - p = 500 - 4$ df (three independent variables and the intercept term). Coefficients were selected using backward elimination. The residuals from the fitted models were visually analyzed and the residual standard errors σ_x and σ_y were calculated. The determined regression coefficients were used to obtain x, y-predictions for the $N = 500$ training images and the $N = 500$ test images.

2.4 Graphical Gaussian Models

Two graphical Gaussian models were constructed from the training set: one with the original non-normal margins and one with transformed normal margins. Two steps were involved in this procedure: model selection, that is, the construction of the independence graph and lieklihood estimation for the variance matrix V.

2.4.1 Model Selection

For selecting the model, sequential backward elimination from a saturated graphical model was used. To test if an edge can be deleted, the likelihood ratio test based on the deviance is used. In each step, the edge with the highest p-value based on the χ^2 test of independence is deleted. If no p-value is significant (significance level $\alpha = 0.05$), the iteration is stopped.

2.4.2 Likelihood Estimation

Based on the graphical model achieved in the model selection step, the maximum likelihood estimate of the variance matrix V is calculated using the IPF algorithm. The IPF algorithm updates the sample variance matrix and fits it to the given graph.

2.4.3 Predictions

The expectation $E_{b|a}(X_b)$ was used to calculate the predictions using the graphical Gaussian model, with $X_a = (R, G, B)$ and $X_b = (POS_x, POS_y)$. By Proposition 6.3.1 (Whittaker 2009), the conditional distribution of X_b given $X_a = a$ is normally distributed with mean

$$E_{b|a}(X_b) = \mu_b + V_{ba}V_{aa}^{-1}(x_a - \mu_a) \tag{4}$$

The equation for $E_{b|a}(X_b)$ is directly related to the linear least squares predictor, with V_{ba} relating to $\mathrm{cov}(Y, X)$ and V_{aa}^{-1} to $\mathrm{var}(X)^{-1}$.

2.5 Error statistics

To determine the goodness of fit of the models, the mean absolute error (MAE) and the standard deviations of the absolute errors were calculated on the training set, separately for x and y-positions. To compare the predictive power of models on unseen data, the same statistics were calculated on the hold-out test set.

2.6 Vine Copula

Finally, the dependence structure between the variables is described using a vine copula. Since the values of R, G and B are known for the training and the test set, they were converted to pseudo-observations using ranking on the combined dataset ($N = 1000$, ties, if any, are broken at random). By construction of the dataset, it is known that POS_x and POS_y are normally distributed. Therefore, instead of ranking the samples of these variables, the cdf for the variables POS_x and POS_y was applied on these variables. The cdfs were estimated using the samples in the training dataset only—since it is assumed the values for x and y are unknown for the testset.

The tree structure of the vine is chosen using forward selection. The selection is performed among all possible copula families in the R package *Vine*

Copula[3]. To this end, the available copulas are fitted using maximum likelihood estimation. Then, bivariate copula families for the edges of the vine are chosen using log-likelihood as selection criterion. First, a saturated model is selected with fitted copulas on all levels of the tree. Then, to simplify the model, independence tests between bivariate distributions were performed (significance level $\alpha = 0.05$). If the null hypothesis of independence could not be rejected, the independence copula was chosen.

For the prediction of the point estimate, the mode of the conditional pdf $f_{POS_x POS_y|RGB} = f_{POS_x POS_y RGB}/f_{RGB}$ was used. Since the denominator of this pdf is constant for given (r, g, b), it suffices to find the argmax of the numerator. To this end, the joint pdf is evaluated at all combinations of x, y, with $x \in \{1, 2, \ldots, 640\}$ and $y \in \{1, 2, \ldots, 480\}$ and fixed (r, g, b) values. The values of x and y were transformed to copula data using the marginal cdfs, which were estimated from the training set. Finally, if the argmax was found, the corresponding x, y-positions were mapped from $[0, 1]$ to $[0, 640]$ and $[0, 480]$ by applying the inverse cdfs.

[3] An overview of the possible families can be found at: `https://cran.r-project.org/web/packages/VineCopula/VineCopula.pdf`

3 Results

3.1 Test for Normality

Figure 2 shows the pairwise correlation plot of the training dataset. It can be seen that the variables R, G, B seem to be non-normally distributed, while POS_x and POS_y are normal. This is underlined by Shapiro-Wilk's Normality Test (Table 1). Therefore, for the construction of the second graphical Gaussian model, the variables R, G, and B were transformed to have standard normal margins. The results of the transformation are shown in Figure 3. Based on Henze-Zirkler's multivariate normality test (conducted after transforming the margins to normal), the null hypothesis that the data were multivariate normal was rejected (HZ: 13.04, $p < 0.001$). Still, the analyses were performed using graphical Gaussian models.

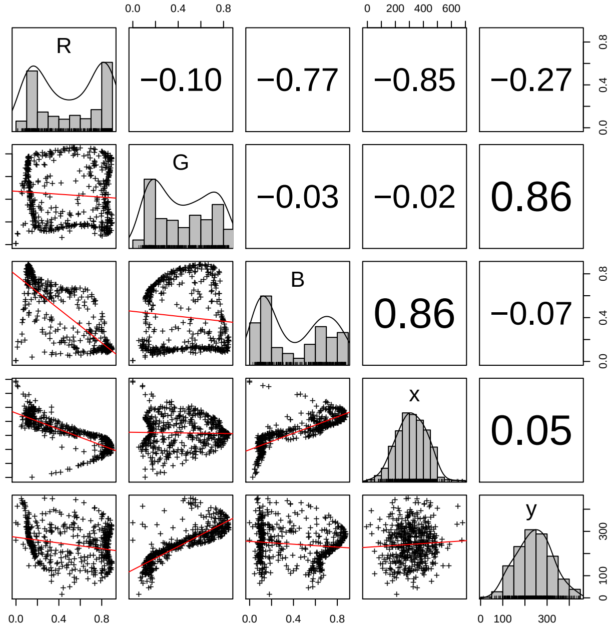


Figure 2: This figure shows a scattermatrix of the training dataset. The upper triangle displays the Spearman correlation coefficients, the diagonal displays the histograms and kernel density estimates, and the lower diagonal shows the bivariate sample distributions with a linear fitting line.

Variable	Statistic	p-value	Normality
R	0.8603	0.0000	no
G	0.9146	0.0000	no
B	0.8562	0.0000	no
POS_x	0.9961	0.2607	yes
POS_y	0.9955	0.1660	yes

Table 1: Shapiro-Wilk's normality test

R	0.093				
G	-0.006	0.062			
B	-0.072	-0.007	0.085		
POS_x	-26.655	-0.663	24.830	12016.115	
POS_y	-5.991	16.311	-2.727	506.025	6591.180
means	0.492	0.430	0.408	317.646	241.764
	R	G	B	POS_x	POS_y

Table 2: The sample variance matrix of the data set. The variances are calculated using the operator $\text{var}_{500}(X, Y)$.

3.2 Linear least squares predictor

Table 2 shows the empirical variance matrix and the mean values of the training dataset. Given a measurement of the average color values $X = (r, g, b)^T$, the linear least squares predictor becomes:

$$\hat{Y}(X) = EY + \text{cov}(Y, X)\,\text{var}(X)^{-1}((r, g, b)^T - EX)$$

$$= \begin{pmatrix} 317.646 \\ 241.764 \end{pmatrix} + \begin{pmatrix} -26.655 & -0.663 & 24.830 \\ -5.991 & 16.311 & -2.727 \end{pmatrix} \begin{pmatrix} 0.093 & -0.006 & -0.072 \\ -0.006 & 0.062 & -0.007 \\ -0.072 & -0.007 & 0.085 \end{pmatrix}^{-1} \begin{pmatrix} r - 0.492 \\ g - 0.430 \\ b - 0.408 \end{pmatrix}$$

$$= \begin{pmatrix} 355.36 - 180.88r - 12.36g + 139.23b \\ 292.55 - 173.80r + 228.36r - 160.24b \end{pmatrix}$$

R	5.935				
G	-1.625	3.418			
B	3.741	-1.440	5.129		
POS_x	1.614	0.095	-1.194	3.222	
POS_y	3.091	-3.324	2.729	-0.006	4.748
	R	G	B	POS_x	POS_y

Table 3: Inverse correlation matrix

In Table 3, the inverse correlation matrix is shown. From the table, the

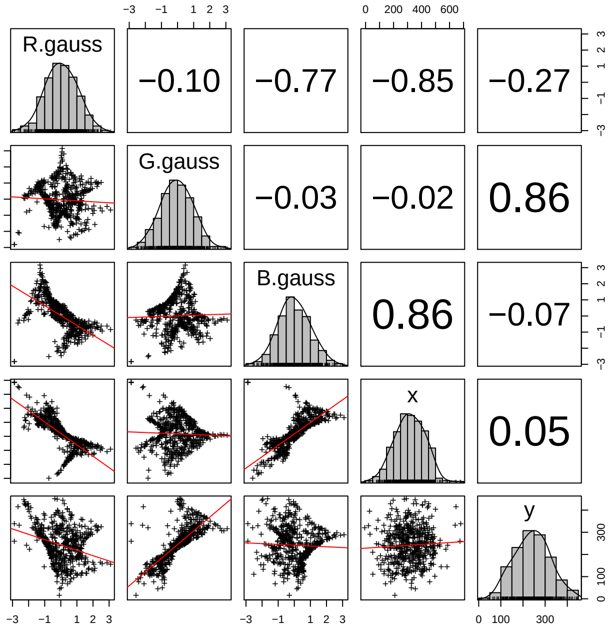


Figure 3: Scattermatrix of the training set after transforming the variables R, G, B to have standard normal margins.

proportion of explained variation for POS_x and POS_y can be calculated:

$$R^2(POS_x; rest) = (3.222 - 1)/3.222 \approx 68.96\,\%$$
$$R^2(POS_y; rest) = (4.748 - 1)/4.748 \approx 78.94\,\%$$

	Estimate	Std. Error	t-value	p	significant
Intercept	355.36	16.35	21.731	< 0.001	yes
r	-180.88	15.89	-11.385	< 0.001	yes
g	-12.36	11.49	-1.075	0.283	no
b	139.23	16.65	8.364	< 0.001	yes

Table 4: Analysis of the coefficients for POS_x

Tables 4 and 5 show the estimated coefficients for the model. From Table 4, it can be seen that the coefficient for the variable G is not significantly different from zero, based on the t-test ($t = -1.075, p = 0.283, df = 496$). Therefore,

	Estimate	Std. Error	t-value	p	significant
Intercept	292.549	9.977	29.32	< 0.001	yes
r	-173.795	9.693	-17.93	< 0.001	yes
g	228.356	7.010	32.58	< 0.001	yes
b	-160.239	10.155	-15.78	< 0.001	yes

Table 5: Analysis of the coefficients for POS_y

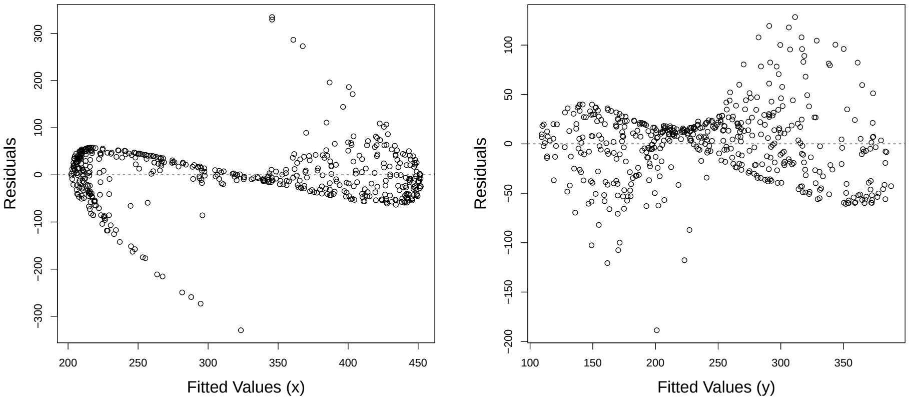


Figure 4: *Left*: Residuals versus fitted plot for POS_x and the linear least squares predictor ($\rho = -0.19$); *Right*: Residuals versus fitted plot for POS_y ($\rho = -0.06$)

another linear least squares model was fit without G as predictor for POS_x. The adapted coefficients are shown in Table 6. The amount of explained variance for POS_x of the new model is $R^2 = 68.89\,\%$. The standard deviation of the residuals are $\sigma_x = 61.26$ on 497 df and $\sigma_y = 37.37$ on 496 df. Figure 4 shows *Residuals versus fitted* plots and Spearman's rank correlations between them.

	Estimate	Std. Error	t-value	p	significant
Intercept	345.68	13.66	25.309	< 0.001	yes
R	-176.29	15.31	-11.517	< 0.001	yes
B	144.13	16.01	9.001	< 0.001	yes

Table 6: Coefficients for the regression without predictor G.

3.3 Graphical Gaussian Model

3.3.1 Model with non-normal margins

The elimination process of the stepwise edge deletion stopped after two iterations. In the first iteration, the edge $\{x, y\}$ was deleted ($\chi^2 = 0.0013, p = 0.972, df = 1$). In the second iteration, the edge $\{G, x\}$ was deleted ($\chi^2 = 1.165, p = 0.281, df = 1$). This process yields the graphical model in Figure 5.

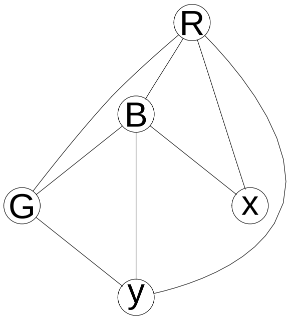


Figure 5: Graphical Gaussian model determined by backward elimination.

R	0.093				
G	-0.006	0.062			
B	-0.072	-0.007	0.085		
POS_x	-26.655	0.041	24.830	12016.115	
POS_y	-5.991	16.311	-2.727	663.141	6591.180
	R	G	B	POS_x	POS_y

Table 7: The fitted matrix V based on maximum likelihood estimation using IPF (non-normal margins).

The maximum likelihood estimation for the variance matrix V using the IPF algorithm is shown in Table 7. The deviance of the simplified model is $D = 2(\hat{l_{sat}} - \hat{l}) = 2(-5216.51 - (-5217.09)) = 1.1$ with 2 df.

3.3.2 Model with transformed normal margins

In this model, in which the variables R, G, B were transformed to have normal margins, the stepwise edge deletion stopped after one iteration and the edge $\{x, y\}$ was deleted ($\chi^2 = 0.003, p = 0.955, df = 1$). The deviance of the simplified model is $D = 2(\hat{l_{sat}} - \hat{l}) = 2(-7351.85 - (-7351.85)) = 0.00$ with 1 df.

tree	edge	family	par	par2	tau	p	indep.
1	4,3	Frank	10.62 (0.50)	-	0.68	< 0.001	no
	1,4	Frank	-9.46 (0.47)	-	-0.65	< 0.001	no
	5,1	Rotated Tawn type 2 270 degrees	-2.70 (0.27)	0.30 (0.02)	-0.24	< 0.001	no
	5,2	Survival BB8	6.00 (0.44)	0.88 (0.02)	0.65	< 0.001	no
2	1,3;4	Rotated Tawn type 1 270 degrees	-1.53 (0.15)	0.20 (0.05)	-0.12	< 0.001	no
	5,4;1	Rotated Tawn type 2 90 degrees	-1.72 (0.09)	0.52 (0.07)	-0.27	< 0.001	no
	2,1;5	Rotated Tawn type 1 180 degrees	1.91 (0.23)	0.32 (0.04)	0.21	< 0.001	no
3	5,3;1,4	Rotated Tawn type 1 270 degrees	-1.78 (0.11)	0.41 (0.05)	-0.24	< 0.001	no
	2,4;5,1	Rotated Tawn type 1 180 degrees	1.19 (0.09)	0.18 (0.14)	0.05	0.86	yes
4	2,3;5,1,4	t	0.04 (0.05)	-7.83 (2.21)	0.02	–	yes

Table 8: Results of the fitting process for vine copulas using the saturated model. 1: R, 2: G, 3: B, 4: POS_x, 5: POS_y

3.4 Vine Copula

The full specifications of the fitted vine copula can be seen in Table 8. Figure 6 shows a matrix of contour plots of the fitted bivariate copula in the vine. The edge $2,4;5,1$ only had a small partial correlation value ($\tau = 0.05$) and the p-value for the dependence test was not significant. Therefore, the edge was set to the independence copula, and, consequentially, the edge $2,3;5,1,4$, as descendant, too. Perspective visualizations of the used bivariate copulas in tree 1 can be found in the appendix (A). Figure 7 shows *Residual versus Fitted* plots and Spearman's rank correlations between them.

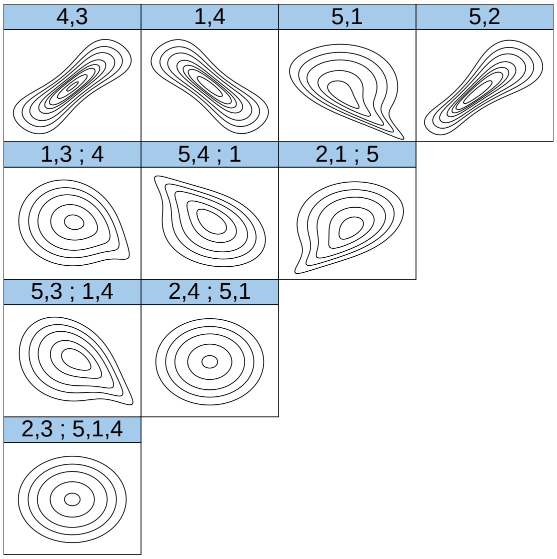


Figure 6: Contour plots of the bivariate copulas in the vine copula (displayed using standard normal margins)

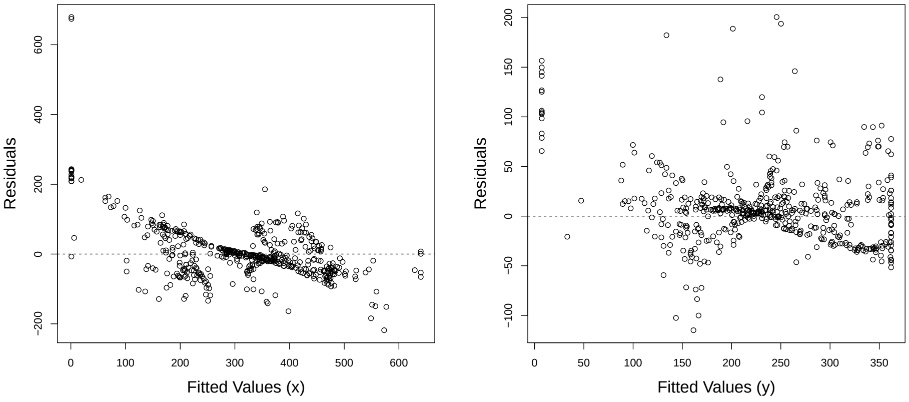


Figure 7: *Left*: Residuals versus fitted plot for POS_x using the vine copula ($\rho = -0.40$); *Right*: Residuals versus fitted plot for POS_y using the vine copula ($\rho = -0.16$)

3.5 Error statistics

In Table 9, the Mean Absolute Errors (MAE) and standard deviations of the residuals for the different models on the are shown for the training set. If there is no information about the values of (r, g, b), then the best predictor for Y is EY. Therefore, the baseline uses $EY = (317.646, 241.76)^T$ for all predictions. Table 10 shows the errors and standard deviations for the test set.

Model	MAE		SD	
	x	y	x	y
Baseline	88.0	65.4	65.3	48.1
Linear least squares predictor	41.1	27.8	45.3	24.8
Graphical Gaussian model	41.1	27.8	45.3	24.8
Transformed graphical Gaussian model	42.5	30.7	42.1	28.7
Vine Copula	51.5	26.4	61.1	30.8

Table 9: Mean Absolute Errors (MAE) and standard deviations for the training set.

Model	MAE		SD	
	x	y	x	y
Baseline	84.8	63.6	63.0	48.6
Linear least squares predictor	40.1	29.3	40.3	29.8
Graphical Gaussian model	40.1	29.3	40.3	29.8
Transformed graphical Gaussian model	38.7	29.7	36.3	30.8
Vine Copula	45.7	23.9	46.6	29.1

Table 10: Mean Absolute Errors (MAE) and standard deviations for the test set

4 Discussion

This report presented different approaches for modeling the dependencies in a five-dimensional regression problem with three independent and two dependent variables. In the following, the achieved results will be discussed with regards to accuracy, model complexity, and possible improvements.

As a general observation, all models performed better than the baseline, both on the training and on the testset. Therefore, there is evidence that the used approaches could (i) capture the dependence structure in the training set and (ii) generalize their predictive power to the unseen test set. Given the simplistic nature of the image features, no over-fitting of the training data was expected, which was underlined by the comparable error statistics for the training and testset. The standard deviations of the errors are similar for the models. However, the vine copula model has a higher standard deviation in x-direction, in particular on the training set.

For the linear least squares predictor, the coefficient for G for the prediction of POS_x was not significantly different from zero and was therefore excluded from the model, which changed the R^2-value only marginally. The values of R^2 (69 % and 79 %) can be considered good values, given the simple image features. While the *Residuals versus fitted* plots (Figure 4) for POS_y shows rather random behavior, a certain interaction between residuals and fitted values for variable POS_x can be seen ($\rho = -0.19$). This interaction should be studied in more detail to ensure that the model makes unbiased predictions.

By sequential edge deletion in the graphical Gaussian model a more parsimonious model could be obtained, with two independence statements: $POS_x \perp\!\!\!\perp POS_y \mid (R, G, B)$ and $G \perp\!\!\!\perp POS_x \mid (POS_y, R, B)$, which rendered the linear least squares predictor and the untransformed graphical Gaussian model congruent. The transformation of the marginals of R, G, and B to standard normal slightly improved the predictive performance and reduced the standard deviation in x-direction, leaving the predictive power for POS_y rather unchanged. Therefore, applying this transformation can be recommended for the given problem. Interestingly, after the transformation, the statement $G \perp\!\!\!\perp POS_x \mid (POS_y, R, B)$ was not valid anymore.

In the vine copula model, only 8 of the 10 copulas had to be fitted due to the insignificant p-value for the edge in tree 3, leading to a simpler model. The vine copula model achieves the best results on the prediction of the y-value on the training and the testset, while the MAE in x-direction is higher than in the other two models. The rather high negative correlation for POS_x between residuals and fitted values ($\rho = -0.40$) shows that model still has much room for improvement. Additionally, high standard deviations of the residuals and errors are evident for the training and testset. Further analysis shows, that they are partly due to the image generation process: while almost all of the image positions were between $[0, 640]$ for POS_x and $[0, 480]$ for POS_y, three images were “out of scope” and resulted in black only images. Another small part of the images was partially black, which shifted the value of the mean color values. While the copula estimated the POS_x positions of these images as $POS_x = 1$, which could result in huge errors—which can be seen in Figure 7—the linear regression was less affected.

In a real-world setting, run-time and computational complexity will be crucial which were left out in this report. However, it should be noted that the predictions of the vine copula model took the longest time due to evaluating the pdf at 640 × 480 = 307200 values. One possibility to speed up this process would be to draw samples.

For the presented problem, the linear least squares predictor and the graphical Gaussian model seem to be most appropriate due to fewer parameters, higher accuracy, and faster prediction time. However, the extracted features for this problem were deliberately simplistic, in order to gain insights into the used models and techniques. Using more sophisticated features, like edge histograms or another map image could lead to dependency structures that can only be captured via the vine copula model.

An interesting research direction will be the inclusion of time dependence: in different real-world scenario, successive images will not be i.i.d. but highly correlated. This could be modeled by a Bayesian belief network, which could take the predictions of the previous image into account.

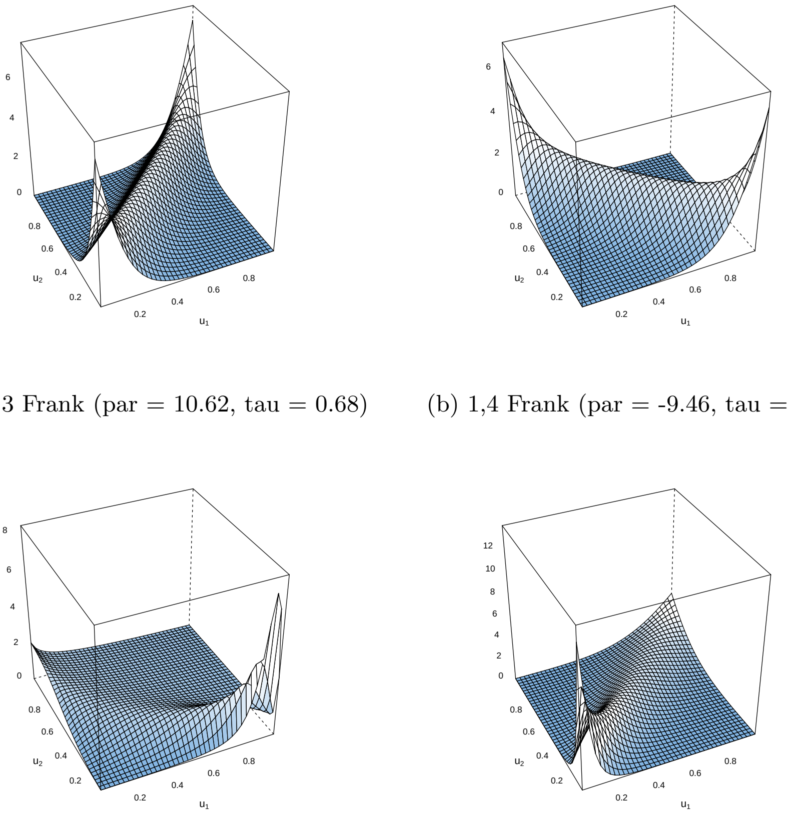
The developed code for this report offers an automated pipeline from image generation to model evaluation. In the future, additional models and different images can be easily defined and compared. To get deeper insights, the tests should be repeated using larger datasets and different map images. Thanks to the presented data generation process, the availability of more data is unproblematic.

References

Whittaker, Joe (2009). *Graphical models in applied multivariate statistics.* Wiley Publishing.

A Visualization of the used copulas

A.1 Tree 1



(a) 4,3 Frank (par = 10.62, tau = 0.68)

(b) 1,4 Frank (par = -9.46, tau = -0.65

(c) 5,1 Rotated Tawn type 2; 270 degrees (par = -2.7, par2 = 0.3, tau = -0.24)

(d) 5,2 Survival BB8 (par = 6, par2 = 0.88, tau = 0.65)

Figure 8: Perspective visualization of the density of the used bivariate copulas for tree 1.

B Dataset with average red, green, and blue values

R	G	B	x	y
0.12	0.69	0.86	405.83	289.16
0.63	0.44	0.09	90.1	350.77
0.17	0.73	0.61	352.62	287.23
0.29	0.59	0.61	341.47	248.59
0.63	0.16	0.63	319.83	138.62
0.87	0.1	0.14	242.07	148.49
0.55	0.79	0.12	286.1	383.25
0.16	0.38	0.77	375.92	232.16
0.55	0.84	0.12	295.22	330.82
0.89	0.18	0.1	205.01	154.38
0.12	0.47	0.85	421.29	253.93
0.11	0.76	0.61	377.32	352.95
0.71	0.16	0.6	299.44	116.19
0.72	0.71	0.2	294.88	257.95
0.64	0.77	0.23	302.34	271.82
0.43	0.14	0.64	374.11	139.15
0.13	0.5	0.8	380.9	248.92
0.83	0.08	0.11	192.47	90.45
0.85	0.52	0.11	203.61	267.03
0.19	0.8	0.37	337.79	335.86
...	...	...	...	...

C Technical Details

The code for generating image patches of a given map image can be found at: `https://github.com/Pold87/draug`. The commented code for extracting the mean color values and generating the models was written in `R`. The used dataset was saved in CSV format. Both can be found at:
`https://github.com/Pold87/decision-theory`

The section on graphical Gaussian models made use of the packages `gRim`, `gRbase` and `Rgraphviz` for the visualization.

The section on VineCopula used the packages `copula` and `VineCopula`.