Supervised classification using copula and mixture copula

Norou Diawara
with: Sumen Sen

Department of Mathematics and Statistics
Old Dominion University

Outline

What is Supervised classification ?

- The term “pattern” or “feature” to denote the p-dimensional data vector $\boldsymbol{x} = (x_1, x_2....x_p)^T \in \boldsymbol{X}$ of measurements.
- There are G classes or groups, denoted by $\Omega = \{\omega_1, \omega_2.....\omega_G\}$.
- Goal is to classify $\boldsymbol{x}$ into one of the G groups.
- Build the classifier using a labeled training data.
- Classifier is a function $f : \boldsymbol{X} \rightarrow \Omega$

Bayes' decision rule

Assume G classes, $\omega_1, \omega_2, ..., \omega_G$ with a prior probability for each class $p(\omega_1), p(\omega_2), ..., p(\omega_G)$.
Then the Bayes' minimum error rule is to assign the unknown pattern vector $\boldsymbol{x}$ to ω_j if

$$p(\omega_j|\boldsymbol{x}) > p(\omega_k|\boldsymbol{x}),$$
$$\forall\ k \neq j;\ j, k \in \{1, 2, ..., G\}.$$

which is equivalent to:

$$p(\omega_j)p(\boldsymbol{x}|\omega_j) > p(\omega_k)p(\boldsymbol{x}|\omega_k),$$
$$\forall\ k \neq j;\ j, k \in \{1, 2, ..., G\}.$$

Classical methods

- ***Quadratic Discriminant Analysis (QDA):*** Most widely used classifier based on normal distribution. The class conditional density is assumed to be multivariate normal distribution:

$$p(\boldsymbol{x}|\omega_j) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_j|^{\frac{1}{2}}} exp\left[\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right].$$

- ***Linear Discriminant Analysis (LDA):*** Assumes the pattern vector **x** is normally distributed in each class but unlike quadratic discriminant analysis this approach assumes $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_G$.

Classical methods cont.

- ***Regularized Discriminant Analysis:*** Proposed by Friedman (1989), this method is applicable when sample size is small and dimension is high.
- ***Naive Bayes (NB):*** A simple probabilistic classifier based on strong independence assumption.

$$p(\boldsymbol{x}|\omega_k) = \prod_{i=1}^{p} f_i^k(x_i),$$

  where f_i marginal distribution of x_i.
  NB is useful for non-normal patterns. This model is very useful even the features are highly correlated (Zhang 2004)
- ***Logistic Regression (LR):*** Logistic regression is a well known classification tool when the features are discrete or mixed.
  In the literature many authors have compared the classification power of LR, LDA and QDA (Efron. 1975, Pohar et al. 2004).

Limitations

All the classical methods described in the previous slides has limitations:

- Normality assumption(LDA,QDA,RDA).
- Independent assumption (NB).
- Logistic regression ignores the distribution of features.

Goal is to build a classifier that does not assume normality or independence. We use copula and mixture copula models to build classifier.

Copula

A p-dimension copula is a function $C : [0,1]^p \rightarrow [0,1]$ with the following properties:

- $C(1, \ldots, u_i, \ldots, 1) = u_i$ for all $i = 1, 2, \ldots, p$ *and* $u_i \in [0, 1]$.
- $C(u_1, u_2, \ldots, u_p) = 0$ *if* at least one $u_i = 0$ for $i = 1, 2, \ldots, p$.
- For any $u_{i1}, u_{i2} \in [0, 1]$ with $u_{i1} \leq u_{i2}$, for $i = 1, 2, \ldots, p$,

$$\sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \ldots \sum_{j_p=1}^{2} (-1)^{j_1+j_2+\cdots+j_p} C(u_{1j_1}, u_{2j_2} \ldots, u_{pj_p}) \geq 0.$$

Likelihood function

Based on a sample size *n* the complete likelihood function is given by:

$$l(\Theta|\boldsymbol{x}) = \sum_{i=1}^{n} log \left[\sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x_i}|\theta^{\boldsymbol{j}}, R^j(\boldsymbol{r})) \right]. \quad (16)$$

with $x_i = (x_{1i}, \ldots, xpi)$, and

$$f_j(\boldsymbol{x}_i|\theta^j, R^j(\boldsymbol{r})) = c_\Phi \left(F_1(x_{1i}|\boldsymbol{\theta}_{1j}), F_2(x_{2i}|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_{pj})|R^j(\boldsymbol{r}) \right) \prod_{k=1}^{p} f_k(x_{ki}|\boldsymbol{\theta}_{kj}). (17)$$

The likelihood is complicated.
Numerical estimation methods do not converge,
and EM algorithm is too slow to converge.

Likelihood function

We build the likelihood function for continuous margins by introducing a latent variable defined as:

$$z_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \in j^{th}\text{class } j = 1, 2, ..., M,\ i = 1, 2, \ldots, n. \\ 0 & \text{otherwise},\ \text{with } \boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{pi}) \in \mathbb{R}^p. \end{cases} \tag{18}$$

Likelihood function using latent variable:

Based on a random sample of size n and the latent unobserved variable Z_{ij} the log-likelihood function of the complete data can be written as:

$$
\begin{aligned}
l(\Theta|\boldsymbol{x}) &= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + logf_j(\boldsymbol{x}_i|\boldsymbol{\theta}^j)\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + \sum_{k=1}^{p} logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\} \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}log\{c\left(F_1(x_{1i}|\boldsymbol{\theta}_{1j}), F_2(x_{2i}|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_{pj})|R^j\right)\} \\
&= l_1 + l_2 + \cdots + l_p + L_c + \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{(1-p)log\pi_j\}, \qquad (19)
\end{aligned}
$$

where

$$
\begin{aligned}
l_k &= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}\{log\pi_j + logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\} \quad k = 1, 2, \ldots, p; \text{and} \qquad (20) \\
L_c &= \sum_{i=1}^{n}\sum_{j=1}^{M} z_{ij}log\{c\left(F_1(x_{1i}|\boldsymbol{\theta}_{1j}), F_2(x_{2i}|\boldsymbol{\theta}_{2j}), \ldots, F_p(x_{pi}|\boldsymbol{\theta}_{pj})|R^j\right)\}. \qquad (21)
\end{aligned}
$$

Two stage estimation algorithm

1. First apply EM algorithm to each l_k defined in Equation (20).
   At E step calculate:

$$E(z_{ij}|x_{ki}) = T_{ijk}^{(l)}(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)}) = \frac{\pi_j^{(l)} f_k(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)})}{\sum_{j=1}^{M} \pi_j^{(l)} f_k(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)})}, \text{ at each } i = 1, 2, \ldots, n. \quad (22)$$

   In M step set:

$$\hat{\boldsymbol{\theta}_{kj}}^{(l+1)} = argmax \left[\sum_{i=1}^{n} \sum_{j=1}^{M} T_{ijk}^{(l)}(x_{ki}|\boldsymbol{\theta}_{kj}{}^{(l)}) \{log\pi_j + logf_k(x_{ki}|\boldsymbol{\theta}_{kj})\} \right]. \quad (23)$$

   Repeat the process until convergence, to obtain $\hat{\boldsymbol{\theta}}^j = (\hat{\boldsymbol{\theta}_{1j}}, \hat{\boldsymbol{\theta}_{2j}}, ..., \hat{\boldsymbol{\theta}_{pj}})$
2. Now use $\hat{\boldsymbol{\theta}}^j$ and maximize the likelihood function given bellow:

$$(\widehat{R^j}, \hat{\pi}) = argmax\{\sum_{i=1}^{n} log \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x_j}|\hat{\boldsymbol{\theta}}^{\boldsymbol{j}}, R^j\}. \quad (24)$$

Score functions

In order to maximize the likelihood function define in Equation (24), we need to obtain the score functions.
Derivative of the likelihood function (continuous case) w.r.t π_j is given by:

$$\frac{\partial l}{\partial \pi_j} = \sum_{i=1}^{n} \frac{\prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) c_\Phi \left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r}) \right)}{\sum_{j=1}^{M} \pi_j \prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) c_\Phi \left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r}) \right)} \tag{25}$$

Derivative w.r.t $R_j(\boldsymbol{r})$ is given by:

$$\frac{\partial l}{\partial R^j(\boldsymbol{r})} = \sum_{i=1}^{n} \frac{\prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) \frac{\partial}{\partial R^j(\boldsymbol{r})} c_\Phi \left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r}) \right)}{\sum_{j=1}^{M} \pi_j \prod_{k=1}^{p} f_k(x_k|\boldsymbol{\theta}_{kj}) c_\Phi \left(F_1(x_{1i}|\widehat{\boldsymbol{\theta}}_{1j}), \ldots, F_p(x_{pi}|\widehat{\boldsymbol{\theta}}_{pj})|R^j(\boldsymbol{r}) \right)}, \tag{26}$$

where,

$$\begin{aligned}
\frac{\partial}{\partial R} c_\Phi (q_1, \ldots, q_p|R) &= \frac{\partial}{\partial R} log(c_\Phi (q_1, \ldots, q_j|R)) c_\Phi (q_1, \ldots, q_p|R) \\
&= -\frac{1}{2} \frac{\partial}{\partial R} \{log|R| + \boldsymbol{q}^{\boldsymbol{T}}(R^{-1} - I)\boldsymbol{q}\} c_\Phi (q_1, \ldots, q_p|R) \\
&= -\frac{1}{2} \frac{\partial}{\partial R} \{log|R| + \boldsymbol{q}^{\boldsymbol{T}} R^{-1} \boldsymbol{q} - \boldsymbol{q}^{\boldsymbol{T}} I \boldsymbol{q}\} c_\Phi (q_1, \ldots, q_p|R) \\
&= -\frac{1}{2} \{R^{-1} + R^{-1} \boldsymbol{q}\boldsymbol{q}^{\boldsymbol{T}} R^{-1}\} c_\Phi (q_1, \ldots, q_p|R),
\end{aligned} \tag{27}$$

Mixture gamma with equi-correlation structure

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	SE	Estimates	SE
α_{11}=2.3	2.2587	0.1679	2.3127	0.1120
β_{11}=3.2	3.3299	0.2771	3.1883	0.1809
α_{12}=12.2	12.4921	1.0459	12.3864	0.7860
β_{12}=13.3	13.0526	1.1356	13.1493	0.8534
α_{21}=5.9	5.8759	0.4057	5.9114	0.2821
β_{21}=1.2	1.2120	0.0797	1.1996	0.0571
α_{22}=10.5	10.3701	1.0525	10.4207	0.8441
β_{22}=11.3	11.567	1.1667	11.4593	0.9544
α_{31}=8.9	8.7293	0.7253	8.8855	0.5987
β_{31}=4.2	4.3436	0.4130	4.2187	0.3365
α_{32}=16.5	17.8022	2.7847	16.8612	1.3325
β_{32}=7.2	6.0330	1.1217	7.0655	0.5038
r_1=0.60	0.6012	0.0271	0.6031	0.0231
r_2=0.20	0.2102	0.0493	0.1932	0.0369
π_1=0.57	0.5699	0.0012	0.5701	0.0001

Tri-variate gamma mixture density with equi-correlation structure.

Mixture gamma with with unstructured correlation.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	SE	Estimates	SE
α_{11}=2.3	2.2825	0.2031	2.2945	0.0756
β_{11}=3.2	3.2718	0.3535	3.2218	0.1330
α_{12}=12.2	12.1802	1.0728	11.9768	1.0493
β_{12}=13.3	13.4178	1.2736	13.6659	1.0271
α_{21}=5.9	5.8371	0.4792	5.8305	0.3147
β_{21}=1.2	1.2209	0.0958	1.2197	0.0623
α_{22}=10.5	10.6501	0.9398	10.6447	0.6551
β_{22}=11.3	11.2384	1.0139	11.2097	0.7106
α_{31}=8.9	8.9600	0.9211	9.0320	0.6605
β_{31}=4.2	4.2264	0.4531	4.1573	0.3457
α_{32}=16.5	17.8474	2.7001	16.6805	1.4286
β_{32}=7.2	6.8295	1.0694	7.1861	0.6408
r_{12}^{1}=0.60	0.6048	0.0431	0.5978	0.0317
r_{13}^{1}=0.40	0.3965	0.0468	0.3966	0.0297
r_{23}^{1}=0.50	0.5091	0.0497	0.5047	0.0292
r_{12}^{2}=0.20	0.1981	0.0718	0.1934	0.0446
r_{13}^{2}=0.15	0.1484	0.0773	0.1513	0.0527
r_{23}^{2}=0.33	0.3371	0.0588	0.3226	0.0467
π_1=0.57	0.5706	0.0117	0.5712	0.0027

Tri-variate gamma mixture density with unstructured correlation.

Finite mixture of discrete distributions

The idea behind the two stage algorithm is to apply EM algorithm to each marginal and estimate the parameters and using those estimates maximize the full likelihood function to obtain estimates of mixing proportion and association matrix. For discrete random variable $\boldsymbol{X} = (x_1, x_2, \ldots, x_p)$, the M component mixture density is given by:

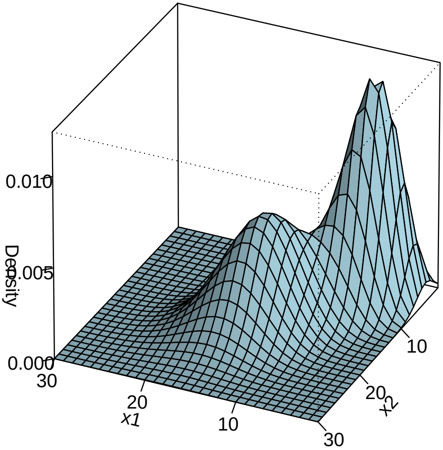
$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}|\theta^j, R^j(\boldsymbol{r})), \tag{28}$$

where

$$\begin{aligned} f_j(\boldsymbol{x}|\theta^{\boldsymbol{j}}, R^j(\boldsymbol{r})) &= P(X_1 = x_1, X_2 = x_2, ..., X_p = x_p|\boldsymbol{\Theta_j}, R^j(\boldsymbol{r})) \\ &= \sum_{t_1=1}^{2}\sum_{t_2=1}^{2}\cdots\sum_{t_p=1}^{2}(-1)^{t_1+t_2+\cdots+t_p} C_\Phi\left(u^j_{1t_1}, u^j_{2t_2}, \ldots,\right. \\ &\quad \left. \ldots, u^j_{pt_p}|R^j(\boldsymbol{r})\right) \end{aligned} \tag{29}$$

with, $C_\Phi(.)$ is the Gaussian copula with association matrix $R(\boldsymbol{r})$, and $u^j_{k1} = F^j_k(x_k|\boldsymbol{\theta}^{kj})$, $u^j_{k2} = F^j_k(x_k - |\boldsymbol{\theta}_{kj})$, for $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, M$. In the above expression $F_j(x_k - |\boldsymbol{\theta}^{kj})$ is the left-hand limit the distribution function F_j at x_k.

Bivariate mixture Poisson.



Application of two stage algorithm in Poisson mixture

For a p-variate mixture poisson model, based on a sample of size *n*, two stage estimation process is given bellow:

Step 1: First use EM algorithm, with initial estimates $\lambda_{kj}^{(l)}$ and $\pi_j^{(l)}$, calculate:

$$T_{ijk}^{(l)} = \frac{\pi_j^{(l)} \frac{\lambda_{kj}^{(l)} e^{-\lambda_{kj}^{(l)}}}{x_k!}}{\sum_{j=1}^{M} \pi_j^{(l)} \frac{\lambda_{kj}^{(l)} e^{-\lambda_{kj}^{(l)}}}{x_k!}}, \quad j = 1, 2 \ldots M \text{ and } k = 1, 2, \ldots, p. \tag{30}$$

Using the above value update the set of estimates by:

$$\widehat{\lambda}_{kj}^{(l+1)} = \frac{\sum_{i=1}^{n} x_{ki} T_{ijk}^{(l)}}{\sum_{i=1}^{n} T_{ijk}^{(l)}}, \quad j = 1, 2, \ldots, M \text{ and } k = 1, 2, \ldots, p. \tag{31}$$

Continue the process until convergence to obtain final set of estimates $\widehat{\lambda}_{kj}$.

Srep 2: Obtain $\widehat{R^j(\boldsymbol{r})}$ and $\widehat{\pi}_j$ by:

$$(\widehat{R(\boldsymbol{r}^j)}, \widehat{\pi}_j) = argmax \sum_{i=1}^{n} log \left(\sum_{j=1}^{M} \pi_j \sum_{t_1=1}^{2} \sum_{t_2=1}^{2} \cdots \sum_{t_p=1}^{2} (-1)^{t_1+t_2+\cdots+t_p} C_\Phi \left(\widehat{u}_{i1t_1}^j, \ldots, \right.\right.$$
$$\left.\left. \ldots, \widehat{u}_{ipt_p}^j | R^j(\boldsymbol{r}) \right) \right), \tag{32}$$

where $\widehat{u}_{ik1}^j = F_k^j(x_{ki}|\widehat{\lambda}_{kj})$, $u_{ik2}^j = F_k^j(x_{ki} - 1|\widehat{\lambda}_{kj})$, for $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, M$.

Equi-correlation structure.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	SE	Estimates	SE
λ_{11}=5	4.6073	0.1699	4.6253	0.1027
λ_{21}=3	2.7637	0.1309	2.7639	0.0856
λ_{31}=2	1.7920	0.1066	1.7937	0.0761
λ_{12}=8	8.1545	0.1486	8.1530	0.1164
λ_{22}=9	8.7404	0.2215	8.7244	0.1131
λ_{32}=7	6.6452	0.1522	6.8257	0.1316
r^1=0.55	0.5891	0.0235	0.582	0.0216
r^2=0.25	0.2620	0.0362	0.2764	0.0284
π_1=0.54	0.5560	0.0133	0.5579	0.0092

Trivariate poisson mixture density with equi-correlation structure.

Unstructured correlation

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	SE	Estimates	SE
λ_{11}=5	4.9539	0.1538	4.8333	0.0872
λ_{21}=3	2.9574	0.1081	2.9844	0.0764
λ_{31}=2	1.9395	0.0809	1.9599	0.0631
λ_{12}=8	8.2912	0.1508	8.2619	0.1175
λ_{22}=9	9.1283	0.2216	9.1349	0.1369
λ_{32}=7	7.1258	0.1988	7.1395	0.1455
r_{12}^{1}=0.55	0.5641	0.0318	0.5749	0.0286
r_{13}^{1}=0.30	0.4063	0.0435	0.3605	0.0344
r_{23}^{1}=0.36	0.4507	0.0298	0.3998	0.0287
r_{12}^{2}=0.20	0.1706	0.0735	0.1929	0.0592
r_{13}^{2}=0.40	0.3933	0.0649	0.4171	0.0448
r_{23}^{2}=0.65	0.6388	0.0415	0.6436	0.0299
π_1=0.54	0.5935	0.0168	0.5759	0.0112

Tri-variate poisson mixture density with unstructured correlation.

Finite mixture of joint discrete and continuous distributions

For a p dimensional random variable $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$, where first p_1 random variables are continuous and rest $p_2 = p - p_1$ of them are discrete. Finite mixture density can be written as:

$$f_{mix}(\boldsymbol{x}|\Theta) = \sum_{j=1}^{M} \pi_j f_j(x|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})), \tag{33}$$

where

$$\begin{aligned} f_j(\boldsymbol{x}|\boldsymbol{\theta}^j, R^j(\boldsymbol{r})) &= \prod_{k=1}^{p_1} f_k(x_k|\boldsymbol{\theta}_{kj}) \sum_{l_{p_1+1}=1}^{2} \cdots \sum_{l_p=1}^{2} (-1)^{l_{p_1+1}+\cdots+l_p} \\ &\times \; C_{\Phi}^{p_1}(F_1(x_1|\boldsymbol{\theta}_{1j}), \ldots, F_{p_1}(x_{p_1}|\boldsymbol{\theta}_{p_1 j}), u^j_{p_1+1, l_{p_1+1}}, \ldots, u^j_{p, l_p}), \end{aligned} \tag{34}$$

with, $u^j_{k1} = F_k(x_k|\boldsymbol{\theta}_{kj})$ and $u^j_{k2} = F_k(x_k - |\boldsymbol{\theta}_{kj})$ and,

$$C_{\Phi}^{p_1}(\boldsymbol{u}) = \frac{\partial^{p_1}}{\partial u_1, \ldots, \partial u_{p_1}} C_{\Phi}(u_1, u_2, \ldots, u_{p_1}, \ldots, u_p).$$

Applications to mixture of Gamma and Poisson.

	Simulation (p=3,M=2)			
	Sample Size=500		*Sample size=1000*	
Parameters	Estimates	SE	Estimates	SE
α_{11}=3.3	3.3464	0.2901	3.3135	0.2019
β_{11}=1.2	1.1916	0.1152	1.1993	0.0790
α_{12}=11.3	11.2245	1.1271	11.3085	0.8268
β_{12}=4.3	4.3342	0.4264	4.2885	0.3239
λ_{21}=2	1.9978	0.1223	2.0079	0.0894
λ_{22}=8	7.9481	0.2378	8.0130	0.1851
r^1=0.60	0.5813	0.0358	0.5958	0.0258
r^2=0.35	0.3424	0.0567	0.3451	0.0396
π_1=0.54	0.5504	0.0011	0.5504	0.0008

Bivariate Gamma and Poisson mixture density.

Method

Salinas et al. (2011) used Gaussian copula classifier for continuous pattern vectors.
Here, we first generalize the approach for discrete and mixed features.
Steps to build the copula classifier are:

1. Based on a training sample from each group ω_j estimate marginal densities.
2. Model the class conditional density $p(\boldsymbol{x}|\omega_j)$ using the gaussian copula and estimate parameters using MLE/IFM.
3. Use Bayes' rule to build classifier.

All discrete features

For discrete patterns, model the class conditional density as:

$$p(x|\omega_k) = \sum_{j_1=1}^{2}\sum_{j_2=1}^{2}\dots\sum_{j_p=1}^{2}(-1)^{j_1+j_2+\dots+j_p} C_\Phi(u^k_{1j_1}, u^k_{2j_2}\dots, u^k_{pj_p}),$$

with $u^k_{j1} = F^k_j(x_j)$ and $u^k_{j2} = F^k_j(x_j - 1)$. Assuming support of F^k_j is an integer. Parameters can be estimated based on a sample of observation $\{\boldsymbol{x}^k_1, \boldsymbol{x}^k_2, \dots, \boldsymbol{x}^k_{n_k}\}$ ($\boldsymbol{x}^k_i \in R^p$) from class ω_k.

Mixed features

If among the p features, p_1's are continuous and rest of them($p_2 = p - p_1$) are discrete. Then, model the class conditional density as :

$$\begin{aligned} p(\boldsymbol{x}|\omega_k) &= \prod_{j=1}^{j=p_1} f_j^k(x_j) \sum_{j_{p_1+1}=1}^{2} \cdots \sum_{j_p=1}^{2} (-1)^{j_{p_1+1}+\ldots+j_p} \\ &\times C_\Phi^{p_1}(F_1^k(x_1), ..., F_{p1}^k(x_{p_1}), u_{p_1+1,j_{p_1+1}}^k, ..., u_{p,j_p}^k), \end{aligned}$$

where

$$C_\Phi^{p_1}(\boldsymbol{u}) = \frac{\partial^{p_1}}{\partial u_1 ... \partial u_{p_1}} \Phi(u_1, u_2 ... u_{p_1} ... u_p).$$

Performance of a classifier

Random Sub-sampling:

1. Total number of sample from class ω_j is N_j.
2. $k_j (< N_j)$ observations are chosen randomly from N_j.
3. Classifier is designed by those k_j samples.
4. Then proportion of correct classification A_i is estimated using remaining $N_j - k_j$ sample from each group.
5. Repeat this process k times and accuracy $A = \frac{1}{k}\sum_{i=1}^{k} A_i \times 100\%$.

Simulation example

- Class 1
  $X_1 \sim Gamma(\alpha_1 = 3.2, \beta_1 = 1.3)$
  $X_2 \sim Poisson(\lambda_1 = 5)$
  $X_3 \sim Geometric(p_1 = 0.28)$
  Training sample 80.
  Test sample 20.
- Class 2
  $X_1 \sim Gamma(\alpha_1 = 2.3, \beta_1 = 4.3)$
  $X_2 \sim Poisson(\lambda_1 = 4)$
  $X_3 \sim Geometric(p_1 = 0.32)$
  Training sample 80.
  Test sample 20.

We assume a equi-correlation structure for the association matrix $R^j(\boldsymbol{r}), j = 1, 2$.

Simulation results

Misclassification rates:

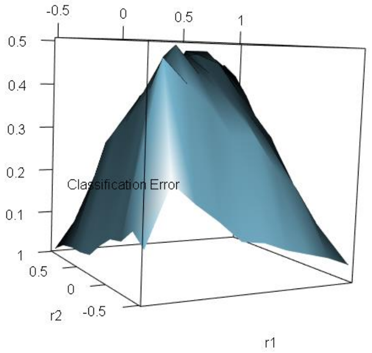
$\rho_1 = 0.1$

ρ_2	Copula	LR	NB
0.1	0.4183	0.4168	0.4213
0.2	0.4132	0.4157	0.4188
0.3	0.4052	0.4155	0.4180
0.4	0.3851	0.4137	0.4180
0.5	0.3603	0.4096	0.4158
0.6	0.3292	0.4077	0.4138
0.7	0.2908	0.4036	0.4127
0.8	0.2366	0.3987	0.4090
0.9	0.1681	0.3885	0.4074

$\rho_2 = 0.5$

ρ_1	Copula	LR	NB
0.1	0.3651	0.4129	0.4281
0.2	0.3881	0.4130	0.4265
0.3	0.3992	0.4118	0.4260
0.4	0.4077	0.4091	0.4253
0.5	0.4061	0.4042	0.4234
0.6	0.393	0.3997	0.4237
0.7	0.3676	0.3925	0.4205
0.8	0.3203	0.3847	0.4181
0.9	0.239	0.3706	0.4175

Classification error plot

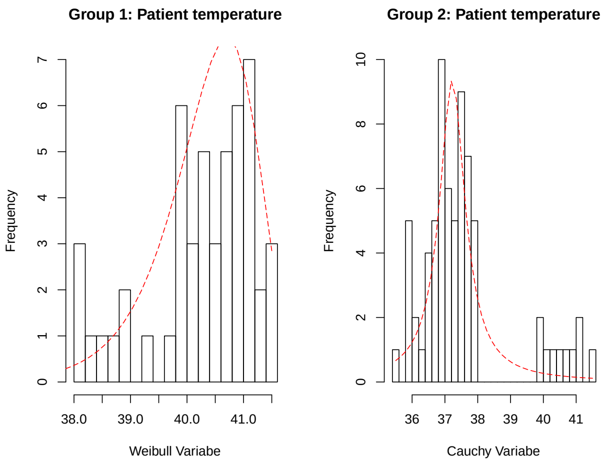


Application to real life data

We applied this copula classification model to a real data.

- *Acute inflammations data* is available online in UCI machine learning archive.
- Three features were chosen
  (1) Continuous need for urination (Yes/No).
  (2) Micturition pains (burning sensation on urinating)(Yes/No).
  (3) Temperature.
- *Nephritis (inflammation of the kidneys)* was the class variable.
- $N_1 = 50$ of the patients had nephritis and $N_2 = 70$ of them did not have nephritis.
- Random sub-sampling method was used to obtain misclassification error.

Histograms


Group 1: Patient temperature
Frequency
0
1
2
3
4
5
6
7
38.0
39.0
40.0
41.0
Weibull Variabe
Group 2: Patient temperature
Frequency
0
2
4
6
8
10
36
37
38
39
40
41
Cauchy Variabe

Misclassification rates comparison

	Copula(UN)	Copula(Equi)	LR	NB
Two binary variables:		29.82%	50.8%	51.24%
Continuous variable included:	6.6%	13.32%	11.88%	13.75%

1. Under this approach, model the class conditional density as:

$$p(\boldsymbol{x}^g|\omega_g) = \sum_{j=1}^{M} \pi_j f_j(\boldsymbol{x}^g|\boldsymbol{\theta}_g^j, R_g^j(\boldsymbol{r})). \quad (35)$$

2. Estimate parameters using proposed two stage algorithm.
3. Use Bayes' rule to build classifier.

Continuous case:

Simulation Setup:

Sample size=1000 (p=2,M=2)	
Class-1	Class-2
α_{11}=2.3	α_{11}=5.1
β_{11}=3.4	β_{11}=1.2
α_{12}=12.2	α_{12}=17.3
β_{12}=1.3	β_{12}=4.3
α_{21}=5.9	α_{21}=3.9
β_{21}=1.2	β_{21}=2.2
α_{22}=10.5	α_{22}=13.5
β_{22}=4.3	β_{22}=7.3
r_1=0.65	r_1=0.25
r_2=0.55	r_2=0.35
π_1=0.57	π_1=0.57

Misclassification errors

We simulated 1000 samples from each group, and randomly choose 800 sample for training and the rest 200 samples to estimate misclassification error. We repeat this process 20 times to obtain average misclassification error rate. Results are given bellow:

Mixture Copula	QDA	LDA	IM
0.1996	0.3057	0.3260	0.5369

Misclassification errors of mixture copula, QDA, LDA and IM model.

Discrete features:

Sample size=1000 (p=2, M=2)	
Class-1	Class-2
λ_{11}=2	λ_{11}=15
λ_{12}=10	λ_{12}=3
λ_{21}=3	λ_{21}=12
λ_{22}=12	λ_{22}=4
r_1=0.62	r_1=0.55
r_2=0.33	r_2=0.25
π_1=0.55	π_1=0.55

Parameter sets for simulation.

Misclassification errors

Mixture Copula	LR	IM
0.2812	0.3315	0.4259

Misclassification errors of mixture copula, LR and IM model.

Mixed features:

Simulation Setup:

Sample size=1000 (p=2, M=2)	
Class-1	Class-2
α_{11}=2.3	α_{11}=12.3
β_{11}=0.2	β_{11}=0.3
α_{12}=10.2	α_{12}=5.1
β_{12}=3.5	β_{12}=2.2
λ_{12}=2	λ_{12}=3
λ_{22}=7	λ_{22}=9
r_1=0.60	r_1=0.65
r_2=0.45	r_2=0.15
π_1=0.55	π_1=0.55

Parameter sets for simulation.

Misclassification errors

Mixture Copula	LR	IM
0.052	0.320	0.091

Misclassification errors of mixture copula, LR and IM model.

Wilt data set

Mixture copula method is implemented to Wilt data set in Johnson et al. (2013).

1. Class: “w” (diseased trees), “n” (all other land cover).
2. GLCM_Pan: GLCM mean texture (Pan band).
3. Mean_G: Mean green value.
4. Mean_R: Mean red value.
5. Mean_NIR: Mean NIR value.
6. SD_Pan: Standard deviation (Pan band).

Misclassification errors

MIxture Copula(Equi)	LDA	QDA
0.19	0.37	0.23

Misclassification error of mixture Copula, LDA and QDA methods.

Conclusion

- New parametric method for supervised pattern reorganization.
- Can be used for discrete and mixed types of features.

- Other copula estimation process.
- Mixed copula models.

- G.M. James and T.J. Hastie. Functional linear discriminant analysis for irregularly samples curves. Journal of the Royal Statistical Society Series B, 63 (3):533-550, 2001.
- G. J. McLachlan The bias of the apprent error rate in discriminant analysis: Biometrica vol. 63, pp. 239-272, 1976.
- H. Joe. Asymptotic efficiency of the two-stage estimation method for copula-based models. Journal of Multivariate Analysis, 94:401419, 2005.
- H. Zhang "The Optimality of Naive Bayes". FLAIRS2004 conference. (available online: PDF (http://www. cs. unb. ca/ profs/ hzhang/publications/ FLAIRS04ZhangH. pdf)).
- J.H. Friedman. Regularized discriminant analysis. Journal of the American Statistical Association, 84:165-175, 1989.
- M. Pohar, M. Blas, and S. Turk. Comparison of Logistic Regression and Linear Discriminant Analysis. MetodoloÅ¡ki zvezki, Vol. 1, No. 1, pp. 143-161, 2004.
- P. X.K. Song,Y. FAN, and J. D.Kalbfleisch. Maximization by parts in likelihood inferance: Journal of the American Statistical Association, Vol. 100, No. 472 pp.1145-1167, 2005.
- R. Salinas-GutiÂ´errez, A. HernÂ´andez-Aguirre, M. J. J. Rivera-Meraz, and E. R. Villa-Diharce. Using Gaussian Copulas in Supervised Probabilistic Classification: Soft Computing for Intelligent Control and Mobile Robotics Studies in Computational Intelligence Volume 318, pp 355-372, 2011.
- R. B. Nelsen. An introduction to copulas. Spinger series in statistics. 2006.
- B.A Johnson, R. Tateishi, N.T. Hoan, A hybrid pansharpening approach and multiscale object based image analysis for mapping diseased pine and oak

Thank You