Semi-Supervised Domain Adaptation with Non-Parametric Copulas

David Lopez-Paz\textsuperscript{1}, Jose Miguel Hernández-Lobato\textsuperscript{2}, Bernhard Schölkopf\textsuperscript{2}

\textsuperscript{1}Max Planck Institute for Intelligent Systems; \textsuperscript{2}Cambridge University

Abstract

We address the problem of domain adaptation [4]. We focus on how to use data from a source task to help estimate another different but related target task, for which limited access to data is assumed.

Our approach consists in:

1. Modeling each of the tasks as a density estimation problem.
2. Estimating the source task as a Regular Vines (R-Vines) Copula Decomposition.
3. Detecting and adapting changes in each of the R-Vine factors across tasks.

Also, we propose a new non-parametric Copula Distribution. Finally, we analyze the performance of the proposed framework in real-world data and compare it to state-of-the-art algorithms.

Copulas

The copula \( c(u) \) of a given density \( p(x) \) describes the dependencies among the u.v. \( x_1, \ldots, x_d \) but contains no information about their marginal effects, since \( p_{\text{d}}(X) \sim U(0,1) \) for any i.r. \( X \) [6]:

\[
p(x) = \frac{1}{u(0,1)^d} \prod_{i=1}^{d} p_i(x_i) \Phi^{-1}(u_i|x_{-i}), \Phi^{-1}(u|x_{-i})
\]

Given a sample \( \{(x_i, y_i)\}_{i=1}^{n} \) from \( p(x, y) \), we can obtain pseudo-samples from its copula \( c \) by mapping each observation to the uniform square using marginal i.i.d. \( U_i \), i.e.

\[
U = \{(u_i, v_i)\}_{i=1}^{n} = \{(P(x_i), P(y_i))\}_{i=1}^{n}
\]

Non-Parametric Bivariate Copulas

KDE using \( U \) is not possible due to its bounded support [5]. Instead, we back-transform the data to have Gaussian marginals using the Gaussian q.h. \( \Phi^{-1} \), and estimate \( c \) as

\[
\hat{c}(u, v) = \frac{\hat{p}_{uv}(\hat{u}, \hat{v}) \Phi^{-1}(\hat{u}) \Phi^{-1}(\hat{v})}{\hat{p}_{u}(\hat{u}) \hat{p}_{v}(\hat{v})} = \frac{\sum_{i=1}^{n} \Phi^{-1}(u_i) \Phi^{-1}(v_i) \hat{p}_i(x_i, y_i)}{\sum_{i=1}^{n} \Phi^{-1}(u_i) \Phi^{-1}(v_i) \hat{p}_i(x_i, y_i)}
\]

Regular Vine Copula Distributions

High-dimensional copulas can be factorized as a collection of bivariate copulas, in a so-called Regular Vine Copula Decomposition or Decomposition [1]:

\[
p(x) = \prod_{i=1}^{d} p_i(x_i) \prod_{j=1}^{d} \prod_{k=j+1}^{d} \left( \sum_{c_{ij} \in \mathcal{C}_{ijk}} p_{ij}(x_{ij}) p_{jk}(x_{jk}) p_{ik}(x_{ik}) p_{ijk}(x_{ijk}) p_{ij}(x_{ij}) p_{jk}(x_{jk}) p_{ik}(x_{ik}) p_{ijk}(x_{ijk}) \right)
\]

These models are formed by a collection of trees, in each of which represent a bivariate copula:

\[
P(231) = \Phi_{231} \cdot \Phi_{23} \cdot \Phi_{31} \cdot \Phi_{21} \cdot \Phi_{31} \cdot \Phi_{31}
\]

- Each bivariate copula forming the vine can belong to a different parametric family.
- The selection of each spanning tree is done by maximizing the sum of the absolute empirical Kendall’s \( \tau \) associated with each edge.
- Conditional c.d.f.s at two points are expressed as partial derivatives of copulas from tree \( i \sim 1 \):

\[
P_j(u_j) = \frac{\partial C_{ij}(u_j, v_{ij})}{\partial v_{ij}} \bigg|_{v_{ij} = 0}
\]

Non-Parametric Regular Vine Copula Distributions

We now introduce the rule in (3) to build non-parametric Regular Vine Copula Distributions that are formed by non-parametric bivariate copulas densities as in (1):

\[
P(x) = \int_{0}^{1} \int_{0}^{1} \prod_{i=1}^{d} \prod_{j=i+1}^{d} N(u_i, \nu_i, \Sigma_i) \Phi \left( \frac{d_i(u)}{\nu_i} \cdot \rho_{ij} \right) du_j
\]

Domain Adaptation with Regular Vines

Assume we are given samples from two different but related tasks, namely the source task data:

\[
D_s = \{(x_s, y_s)\}
\]

and the target task data:

\[
D_t = \{(x_t, y_t)\}
\]

with \( X_s \subset X_t \). Our objective is to use the maximum amount of information from the source task data to infer a better model for the target task data.

We rely on the proposed technique for the two dimensional feature space case. First, we infer a R-Vine decomposition \( p_2 \) from the source data (equation 2), and close it in a second R-Vine distribution \( p_2 \) for the target task, namely:

\[
\hat{p}_1(x, y) = \hat{p}_1(x) \hat{p}_2(x) \hat{p}_2(x, y) \hat{p}_2(x, y) \hat{p}_2(x, y) \hat{p}_2(x, y) \hat{p}_2(x, y)
\]

When comparing pairs of factors using the target task data, several adaptation schemes arise:

1. \( p_1 = \hat{p}_1 = 0 \), \( p_2 = \hat{p}_2 \) and \( \hat{p}_2 \) are restimated using the corresponding samples from \( D_s \).
2. \( p_2 = \hat{p}_2 = 0 \), \( p_1 = \hat{p}_1 \) and \( \hat{p}_1 \) are restimated using the corresponding samples from \( D_t \cup D_s \).
3. \( c_{ij} \neq c_{ij} \), \( c_{ij} \neq c_{ij} \) are restimated using the corresponding samples from \( D_t \cup D_s \).
4. \( c_{ij} \neq c_{ij} \), \( c_{ij} \neq c_{ij} \) are restimated using the corresponding samples from \( D_t \cup D_s \).

Several of the previous can occur, but there is no limitation in addressing them independently.

Detecting Changes

We propose the use of the Maximum Mean Discrepancy (MMD) [5] test. Given samples \( x \) and \( y \) from two distributions \( X \) and \( Y \), MMD will determine \( X \sim Y \) according to distance between the embeddings of the empirical distributions of these two samples in a RKHS is significantly large. The empirical form of this statistic takes the form:

\[
\text{MMD}(x, y) = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \hat{K}(x_i, y_j) - \hat{K}(y_i, x_j) - \hat{K}(x_i, y_j) + \hat{K}(y_i, x_j)
\]

Semi-Supervised Domain Adaptation

Note that unlabeled, unpaired or incomplete data is still helpful to refine the factors of the R-Vine Decomposition \( p_2 \) that depend on it. Please refer to the experiments.

Experimental Results

Average Test Log-Likelihood for Density Estimation:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Auto</th>
<th>Cloud</th>
<th>Housing</th>
<th>Magic</th>
<th>Page-Blocks</th>
<th>Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNPRV</td>
<td>0.86</td>
<td>0.97</td>
<td>0.91</td>
<td>0.72</td>
<td>0.86</td>
<td>0.20</td>
</tr>
<tr>
<td>PRV</td>
<td>0.84</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>0.88</td>
<td>0.73</td>
</tr>
<tr>
<td>KIDR</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>NIPS</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Notes:

1. All features are continuous to ensure copula functional form uniqueness.
2. Our method is abbreviated as NPP (Non-Parametric Regular Vine).
3. Experiments are based on 50 random 2000-sample training/test splits.
4. Only 5% from the target data task was labeled.
5. UNPRV ignores this second labeled data from the target task.

Future Work

We are investigating how to improve the proposed framework in the following directions:

1. Inclusion of discrete features.
2. More advanced techniques of factor refinement, beyond the presented substitution.
3. Domain Adaptation rules for parametric vines: correction of bivariate copula families or parameters across domains.

References