A dependent partition-valued process for multitask clustering and evolving networks

Konstantina Palla, David A. Knowles, Zoubin Ghahramani

Amsterdam, 12th June 2013
Partition-valued processes

Related work

The dependent partition-valued process (DPVP)

Models

Experiments
**Partition-valued Processes**

<table>
<thead>
<tr>
<th>$\pi_1$</th>
<th>$\pi_2$</th>
<th>$\pi_3$</th>
<th>$\pi_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
</tbody>
</table>

Let:

- $[n]$ denote the natural numbers \( \{1, \ldots, n\} \).
- Each $\pi_\tau$ is a set of disjoint non-empty clusters indexed by a covariate (time/location) $\tau \in \mathcal{T}$

How do we induce dependency among the partitions?
**Related Work**

- **Dependent Dirichlet processes (DDP)**
  \[ G^{(\tau)} = \sum_{k=1}^{\infty} w_k^{(\tau)} \delta_{\theta_k^{(\tau)}} \]
  e.g.: MacEachern [1999, 2000]

- **Dependent partitions**
  e.g Sudderth and Jordan [2008], Duan et al. [2007]

Another related line of work uses fragmentation-coagulation processes [Bertoin, 2006, Teh et al., 2011].
Basic Idea

How should we introduce dependency on $\tau$ among the partitions, $\pi_\tau$?

- stick breaking to generate a partition $\pi$
- generating Bernoulli by thresholding Gaussians
- using Gaussian processes to model dependency on $\tau$
STICK BREAKING TO GENERATE A PARTITION

\[
G = \sum_{k=1}^{\infty} w_k \delta_{\theta_k}
\]

\[
w_k = u_k \prod_{l=1}^{k-1} (1 - u_l) \quad \forall k \in \mathbb{Z}
\]

\[
u_k \sim Beta(1, \alpha) \quad \forall k \in \mathbb{Z}
\]

For each object \( n \in [N] \)

1. Set \( k := 1 \)
2. Sample branching variable \( a_{nk} \sim Bernoulli(u_k) \)
3. If \( a_{nk} \) then assign \( c_n := k \), else increment \( k \) and go to 2.
Bernoulli by thresholding a Gaussian

To generate, $a_{nk} \sim \text{Bern}(u_k)$ we can

$$f_{nk} \sim \mathcal{N}(0, \sigma^2)$$

$$a_{nk} = \mathbb{I}[f_{nk} < \phi^{-1}(u_k|0, \sigma^2)]$$

$\phi(.|\mu, \sigma^2)$: Gaussian cumulative distribution function
Gaussian Processes to Model Dependency on $\tau$

Coupling Gaussians $f_{nk}$ over $\tau$:

- Extend $f_{nk}$ to random functions
  $f_{nk}(\tau), \tau \in T$
- Gaussian prior on $f_{nk} \rightarrow$ Gaussian process
  $f_{nk}(\cdot) \sim_{iid} \text{GP}(0, \Sigma) \ \forall n \in [N], k \in \mathbb{Z}$
- The covariance function $\Sigma(\tau, \tau')$ captures dependence on $T$

Example of 3 samples from a GP. Each plot represents $f_{nk}(\cdot)$
• Given functions \( \{f_{nk}\} \) and stick lengths \( u_k \), decide branching variables

\[
\begin{align*}
  u_k &\sim_{\text{iid}} \text{Beta}(1, \alpha) \quad \forall k \in \mathbb{Z} \\
  f_{nk}(\cdot) &\sim_{\text{iid}} \text{GP}(0, \Sigma) \quad \forall n \in [N], k \in \mathbb{Z} \\
  a_{nk}(\tau) &= \mathbb{I}[f_{nk}(\tau) < \phi^{-1}(u_k|0, \Sigma(\tau, \tau))] \\
  &\quad \forall n \in [N], k \in \mathbb{Z}
\end{align*}
\]

• Set cluster assignments \( c_n(\tau) = \min\{k : a_{nk}(\tau) = 1\} \)

• Note that \( f_{nk}(\tau), f_{nk}(\tau') \) correlated with covariance \( \Sigma(\tau, \tau') \Rightarrow a_{nk}(\tau), a_{nk}(\tau') \) correlated too!
**Dependent Partition Valued process (DPVP)**

- $DPVP(\alpha, \tau, \Sigma)$, $\alpha$ is the concentration parameter, $\tau$ is the vector of covariate locations, $\Sigma$ is the covariance function
- CRP marginals at each covariate value $\tau$
- Global mixing proportions $u_k$
- Pitman-Yor marginals; $u_k \sim_{iid} Beta(1 - d, \alpha + kd)$, where $\alpha > 0$ and $0 \leq d < 1$
MODELS

- Clustering multiple data sources
- Evolving networks
**Clustering Multiple Sources**

- \( N \) objects, \( L \) data sources
- At each \( \tau \); \( \pi_\tau \) is the partitioning, \( Y^\tau \) is data with dimension \( N \times D^\tau \)
- The model is then

\[
\begin{align*}
c_{1:n}^\tau & \sim DPVP(\alpha, t, \Sigma) \\
\theta_k^\tau & \sim P^\tau \\
y_n^\tau | c_{1:n}^\tau, \theta_k^\tau & \sim F^\tau(\theta_{c_{1:n}^\tau}^\tau)
\end{align*}
\]

\( \theta_k^\tau \) are cluster parameters, \( P^\tau \) are priors on the cluster parameters, \( F^\tau \) are data likelihoods.
Evolving Community Structure Model

- $N$ objects. At each $\tau$; $Y^\tau$ is a $N \times N$ binary matrix denoting presence or absence of links
- Probability of a link between $n$ and $n'$ at time $\tau$ depends on the cluster assignments $c_n^\tau$ and $c_{n'}^\tau$ (see e.g. Infinite Relational Model of Kemp et al. [2006]).

\[
c_n^\tau \sim DPVP(\alpha, \tau, \Sigma)
\]
\[
\theta_{kk'}^\tau \sim \text{Beta}(\beta, \beta)
\]
\[
y_{nn'}^\tau | c^\tau, \theta^\tau \sim \text{Bernoulli}(\theta_{c_n^\tau c_{n'}^\tau})
\]

where $\theta_{kk'}^\tau$ is the link probability between clusters $k$ and $k'$ and $y_{nn'}^\tau$ denotes the presence of a link between objects $n$ and $n'$ at time $\tau$
**Choice of Kernel**

- Kernel $\Sigma$ determines the dependency among the data sources
- GP offers flexibility in choice; depends on the application at hand
- Choices:
  1. **Squared exponential**: When known covariate value e.g. time or spatial location.
     \[
     \Sigma(\tau, \tau') = \exp \left( -\frac{(\tau - \tau')^2}{2r^2} \right)
     \]
     where $r$ is the lengthscale.
  2. **Similarity kernel**: When no or little prior knowledge about clustering similarity among the sources. We learn the kernel from data.
     \[
     \Sigma(\tau, \tau') \sim \text{Uniform}[-1, 1], \tau \neq \tau'
     \]
  3. **Tree structured dependency**. See next slide!

Note: In all cases $\Sigma(\tau, \tau) = 1$. DPVP invariant to scaling of the covariance matrix
Interpret the known tree over the data sources as a graphical model:

\[ N(a|0, v_a) \]

\[ N(b|a, v_{ab}) \]

\[ N(f_1|b, v_{b1}) \]

\[ N(f_2|b, v_{b2}) \]

\[ N(f_3|a, v_{a3}) \]

Constrain the marginal variance to be one, so

\[ v_{b1} = v_{b2} := 1 - v_a - v_{ab} \]

\[ v_{a3} := 1 - v_a \]

Integrating out \( a \) and \( b \) we have

\[
\begin{bmatrix}
  f_1 \\
  f_2 \\
  f_3
\end{bmatrix}
\sim N
\begin{pmatrix}
  0 \\
  0 \\
  0
\end{pmatrix}
, \begin{pmatrix}
  1 & v_a + v_{ab} & v_a \\
  v_a + v_{ab} & 1 & v_a \\
  v_a & v_a & 1
\end{pmatrix}
\]
Cancer cell line encyclopedia [CCLE, Barretina et al., 2012]

- $N = 432$ cancer lines. $L = 4$ different data sources (measurements):
  Drug sensitivity (SENS, $D=24$), Gene expression (GE, $D=1000$), Copy number variation (CNV, $D=1000$), Oncogene mutations (ONCO, $D=1600$)
- Task: hold out 10% of SENS data, attempt imputation.
- Compare to:
  1. DP mixture (DPM) independently at each data source,
  2. DPM with common clustering to all data sources
Cancer cell line encyclopedia - Results

<table>
<thead>
<tr>
<th></th>
<th>Independent</th>
<th>Shared</th>
<th>DPVP model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$-3.221 \pm 0.552$</td>
<td>$-1.109 \pm 0.069$</td>
<td>$-0.902 \pm 0.097$</td>
</tr>
</tbody>
</table>

Table: CCLE results: log predictive likelihood

Figure: Learnt correlation matrix
Experiments - Clustering Multiple Data Sources

Hapmap gene expression data

- $N = 1000$ genes from $D = 618$ individuals
- $L = 7$ different populations
- Each data source $Y^\tau$ is $N \times D^\tau$, where $D^\tau \subset D$, $\tau = 1, \cdots, L$
- Target: learn population specific clusterings of the genes

Prior Knowledge

Figure: Tree structure of human populations in HapMap.
Hapmap gene expression data - Results

<table>
<thead>
<tr>
<th></th>
<th>Independent</th>
<th>Shared</th>
<th>DPVP model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$-1.357 \pm 0.055$</td>
<td>$-1.134 \pm 0.013$</td>
<td>$-1.277 \pm 0.016$</td>
</tr>
</tbody>
</table>

**Table:** HapMap results: log predictive likelihood
van de Bunt’s students [Van De Bunt et al., 1999]

- $N = 52$ university friends
- $L = 7$ time points
  - first 4 are 2 weeks apart
  - last 4 are 3 weeks apart
- At each time point $\tau$: $Y^\tau$ is a $N \times N$ binary link matrix
- Target: learn evolution of clustering & predict heldout links
- Compare to:
  1. Independent IRM at each time point
  2. Shared clustering but independent cluster link probabilities across all time points
Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Independent</th>
<th>Shared</th>
<th>DPVP model</th>
</tr>
</thead>
<tbody>
<tr>
<td>van de Bunt</td>
<td>$-0.530 \pm 0.025$</td>
<td>$-0.502 \pm 0.022$</td>
<td>$-0.095 \pm 0.017$</td>
</tr>
</tbody>
</table>

**Table**: van de Bunt’s dataset results: log predictive likelihood
CONCLUSION

- DPVP; a process that defines a distribution over partitions
- Two models;
  1. Clustering multiple datasources
  2. Evolving community structure model
- Encouraging results on real world datasets
Thank you!


**APPENDIX**

van de Bunt’s students

<table>
<thead>
<tr>
<th>Model</th>
<th>ind. IRM</th>
<th>Shared</th>
<th>ECS</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster assign.</td>
<td>$c_n^T \sim CRP(\alpha)$</td>
<td>$c_n \sim CRP(\alpha)$</td>
<td>$c_n^T \sim DPVP(\alpha, \tau, \Sigma)$</td>
</tr>
<tr>
<td>cluster link prob.</td>
<td></td>
<td>$\theta_{kk'} \sim Beta(\beta, \beta)$</td>
<td></td>
</tr>
<tr>
<td>Likelihood</td>
<td>$\prod_{\tau=1}^{L} \prod_{n=1}^{N} \prod_{n'=1}^{N} \theta_{c_n^\tau c_n'^\tau}^{y_{nn'}} (1 - \theta_{c_n^\tau c_n'^\tau})^{(1-y_{nn'})}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>