Probabilistic Modeling for Novelty Detection with Applications to Fraud Identification

Rémi Domingues

PhD Defense
EURECOM - Sorbonne University
January 29th 2019

Advisor: Maurizio Filippone
Co-advisor: Pietro Michiardi
Motivations - The Amadeus use case
Motivations - The Amadeus use case
Motivations - Fraud detection

- Compromised user accounts
- Fraudulent bookings
- Payment frauds
- Malicious bots
Anomaly detection

- Supervised learning – The class imbalance problem
- Unsupervised learning – Novelty detection
  - Recognition of anomalies in test data which differ significantly from the training set
  - Estimate the distribution of nominal samples
  - Similar to a one-class classification problem
Novelty detection
Novelty detection
Industrial constraints and research challenges

- Proactive, unlabelled data
- Anomalies in training data
- Continuous scoring
- Numerical & categorical data
- Scalable and distributed
- White-box model
- Little tuning

- Novelty detection
- Robustness
- Probabilistic method
- Variational learning of joint distributions
- Mini-batch learning
- Interpretable
- Nonparametric
Dirichlet Process Mixture Model
Dirichlet Process Mixture Model (DPMM)

- Weighted mixture of multivariate distributions in the exponential family
- Nonparametric Bayesian method: infinite-dimensional parameter space
- Dirichlet Process as nonparametric prior
- A product of exponential-family distributions is in the exponential-family
- Probabilistic, mini-batch training, categorical support, clustering
Dirichlet Process

- Bayesian nonparametric model
- Distribution over distributions

- Consider a Gaussian $G_0$:

$$G \sim DP(\alpha, G_0)$$
Stick-breaking process

- Constructive way of forming $G$

- Weights $\pi_k(v) = v_k \prod_{j=1}^{k-1} (1 - v_j)$, with $v_k \sim \text{Beta}(1, \alpha)$

- $G \sim DP(\alpha, G_0) \iff G = \prod_{k=1}^{\infty} \pi_k \delta_{\theta_k}$
  - $\theta_k^* \sim G_0$
  - $\delta_{\theta_k}$ is the indicator function which evaluates to zero everywhere, except for $\delta_{\theta_k}(\theta_k) = 1$
Dirichlet Process Mixture Model

\[ z_n \quad \xrightarrow{\sim} \quad x_n \quad \quad \begin{array}{c} \nu_k \quad \xrightarrow{\sim} \quad \eta^*_k \quad \xrightarrow{\sim} \quad \lambda \quad \xrightarrow{\sim} \quad \alpha \quad \xrightarrow{\sim} \quad s_0, r_0 \end{array} \]
Dirichlet Process Mixture Model

1. Draw $\alpha | s_0, r_0 \sim \Gamma(s_0, r_0)$

2. Draw the stick length $v_k | \alpha \sim \text{Beta}(1, \alpha)$, yielding the mixing weights $\pi_k(v) = v_k \prod_{j=1}^{k-1} (1 - v_j)$

3. Draw component $\eta_k^* | \lambda \sim G_0$, with $G_0$ conjugate prior in the exponential family, e.g. $p(X|\eta^*)$ multivariate normal, $G_0$ Normal-wishart

4. Assign the data to the components: $z_n | v \sim \text{Mult}(\pi(v))$
   Generate the observations: $x_n | z_n \sim p(x_n|\eta_{z_n}^*)$
DP mixture inference

• Predictive density: $p(x_{N+1} | X, \theta) = \int p(x_{N+1} | W)p(W | X, \theta) dW$

• Intractable posterior over the latent variables $p(W | X, \theta)$

• Approximate inference
  • Markov Chain Monte Carlo techniques, e.g. Gibbs sampling
  • Variational Inference

  • *Variational inference is that thing you implement while waiting for your Gibbs sampler to converge.* — David Blei
Variational inference

- Approximate the **posterior** $p$ by a **tractable approximation** $q$ with variational parameters

- $q$ is from a family of simpler distributions
  
  $q(\mathbf{v}, \eta^*, \mathbf{z}, \mathbf{w}) = q_{\alpha,\beta}(\mathbf{v}) \cdot q_\tau(\eta^*) \cdot q_r(\mathbf{z}) \cdot q_{g_1,g_2}(\mathbf{w})$

  - $q_{\alpha,\beta}(\mathbf{v})$: product of **Beta**
  - $q_\tau(\eta^*)$: product of distributions in the **exponential family**
  - $q_r(\mathbf{z})$: product of **multinomials** on cluster assignment variable $\mathbf{z}$
  - $q_{g_1,g_2}(\mathbf{w})$: $\Gamma$ distribution

- Hyperparameters: $\lambda$, $\alpha$, $s_0$ and $r_0$

- Latent variables: $\mathbf{v}$, $\eta^*$, $\mathbf{z}$ and $\mathbf{w}$

- Variational parameters: $\alpha_k$, $\beta_k$, $\tau_k$, $r_{nk}$, $g_1$ and $g_2$
Variational inference

1. Initialize the model parameters

2. Optimize the **variational parameters** to minimize

\[
D_{KL}(q(W)||p(W|X, \theta)) = \mathbb{E}_{q}[\ln q(W)] - \mathbb{E}_{q}[\ln p(W, X|\theta)] + \ln p(X|\theta)
\]

Equivalent to maximizing \( \ln p(X|\theta) \geq \mathbb{E}_{q}[\ln p(W, X|\theta)] - \mathbb{E}_{q}[\ln q(W)] \)

3. Compute the expectation of \( p(W|X, \theta) \) under \( q(W|X) \), e.g.

\[
\ln q^{*}_{\alpha, \beta}(v) = \mathbb{E}_{\eta^{*}, z, w}[\ln p(X, v, \eta^{*}, z, w)] + c = \prod_{k=1}^{K-1} \text{Beta}(\alpha_k, \beta_k)
\]

4. Compute the **geometric means**

- \( \mathbb{E}[\ln v_k], \mathbb{E}[\ln(1 - v_k)], \mathbb{E}[\eta^{*}], \mathbb{E}[-a(\eta^{*})], \mathbb{E}[z_{nk}], \mathbb{E}[w] \) and \( \mathbb{E}[\ln w] \)
- Update the model parameters to maximize the expectation of \( p(W, X|\theta) \) under \( q(W|X) \)
• Nondecreasing, used for **convergence monitoring**

\[
\ln p(X|\theta) \geq \mathbb{E}_q[\ln p(W, X|\theta)] - \mathbb{E}_q[\ln q(W)] \\
\geq \mathbb{E}_q[\ln p(X, Z, \eta^*, V, W|\theta)] - \mathbb{E}_q[\ln q(Z, \eta^*, V, W)] \\
\geq \mathbb{E}_q[\ln p(X|Z, \eta^*)] + \mathbb{E}_q[\ln p(Z|V)] + \mathbb{E}_q[\ln p(\eta^*|\lambda)] \\
+ \mathbb{E}_q[\ln p(V|W)] + \mathbb{E}_q[\ln p(W|s_0, r_0)] - \mathbb{E}_q[\ln q_{\alpha,\beta}(V)] \\
- \mathbb{E}_q[\ln q_{\tau}(\eta^*)] - \mathbb{E}_q[\ln q_r(Z)] - \mathbb{E}_q[\ln q_{g_1,g_2}(W)]
\]
Predictive distribution

\[ p(x_{N+1} | X, \theta) = \int \sum_{k=1}^{\infty} \pi_k(\nu) p(x_{N+1} | \eta_k^*) d\nu(\nu, \eta^* | X, \theta) \]

\[ \approx \sum_{k=1}^{K} E_q[\pi_k(\nu)] E_q[p(x_{N+1} | \eta_k^*)]. \]  

- **Analytically,** we obtain \( E_q[\pi_k(\nu)] = \frac{\alpha_k}{\alpha_k + \beta_k} \prod_{i=1}^{k-1} \left( 1 - \frac{\alpha_i}{\alpha_i + \beta_i} \right) \)

- **Monte Carlo sampling** is used to estimate the density
  1. Draw \( m \) samples from \( q_{\tau}^*(\eta^*) \)
  2. Compute each \( p(x_{N+1} | \eta^*) \)
  3. Average the resulting \( m \) likelihoods
Experimental survey
Algorithms

Gaussian Mixture Model
Dirichlet Process Mixture Model
Robust Kernel Density Estimation
Least-Squares Anomaly Detection
Probabilistic PCA

Probabilistic

Mahalanobis
Local Outlier Factor
Angle-Based Outlier Detection
Subspace Outlier Detection

Distance-based

Nearest neighbors

One-class SVM

Domain-based

Grow When Required network

Neural networks

Kullback-Leibler Divergence

Information theoretic

Isolation Forest

Isolation
Results

Average outlier detection performances on 15 datasets (5 runs)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ROC AUC</th>
<th>PR AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>IForest</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>RKDE</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>PPCA</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>OCSVM</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>KL</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>GMM</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>Mahia</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>SOD</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>DPMM</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>DPGMM</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>LOF</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>ABOD</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>GWR</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>LSA</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
### Results - No classification datasets

#### Average outlier detection performances on 10 datasets (5 runs)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>GMM</th>
<th>DPGMM</th>
<th>DPMM</th>
<th>RKDE</th>
<th>PPCA</th>
<th>LSA</th>
<th>MAHA</th>
<th>LOF</th>
<th>ABOD</th>
<th>SOD</th>
<th>KL</th>
<th>GWR</th>
<th>OCSVM</th>
<th>IFOREST</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR AUC</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>14</td>
<td>9</td>
<td>12</td>
<td>11</td>
<td>1</td>
<td>8</td>
<td>13</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>ROC AUC</td>
<td>13</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>9</td>
<td>10</td>
<td>14</td>
<td>4</td>
<td>12</td>
<td>11</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>
Area under the ROC and Precision-Recall curves

- 10 frauds, 990 normal transactions (i.e. 1% positives, 99% negatives)
- Prediction 1: 5 frauds correctly labelled, all normal transaction correctly labelled
- Prediction 2: 5 frauds correctly labelled, 20 normal transactions incorrectly labelled

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>ROC</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction 1</td>
<td>0.75</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>Prediction 2</td>
<td>0.74</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>

- The ROC AUC downplays the impact of false positives when negative observations are over-represented
### Scalability

- **Runtime** and **memory** scalability
- Stability, robustness, resistance to the curse of dimensionality
- Datasets of increasing size, dimensionality and noise

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Training/prediction time</th>
<th>Mem. usage</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>🍃 Samples 🌃 Features</td>
<td>🍃 Samples 🌃 Features</td>
<td>🌳 Noise 🇺🇸 High dim. 🇺🇸 Stability</td>
</tr>
<tr>
<td>GMM</td>
<td>Low/Low</td>
<td>Medium/Medium</td>
<td>Low</td>
</tr>
<tr>
<td>BGM</td>
<td>Low/Low</td>
<td>Medium/Medium</td>
<td>Low</td>
</tr>
<tr>
<td>DPGM GMM</td>
<td>Medium/Low</td>
<td>High/High</td>
<td>Low</td>
</tr>
<tr>
<td>RKDE</td>
<td>High/High</td>
<td>High/High</td>
<td>High</td>
</tr>
<tr>
<td>PPCA</td>
<td>Low/Low</td>
<td>High/Low</td>
<td>Low</td>
</tr>
<tr>
<td>LSA</td>
<td>Low/Medium</td>
<td>Low/Low</td>
<td>Medium</td>
</tr>
<tr>
<td>MAHA</td>
<td>Low/Medium</td>
<td>Medium/Low</td>
<td>Low</td>
</tr>
<tr>
<td>LOF</td>
<td>High/High</td>
<td>Low/Low</td>
<td>High</td>
</tr>
<tr>
<td>ABOD</td>
<td>Low/High</td>
<td>Low/Medium</td>
<td>Low</td>
</tr>
<tr>
<td>SOD</td>
<td>High/High</td>
<td>Low/Medium</td>
<td>High</td>
</tr>
<tr>
<td>KL</td>
<td>Low/Medium</td>
<td>Low/Medium</td>
<td>Low</td>
</tr>
<tr>
<td>GWR</td>
<td>Medium/Medium</td>
<td>Medium/Low</td>
<td>Low</td>
</tr>
<tr>
<td>OCSVM</td>
<td>High/High</td>
<td>Low/Low</td>
<td>Low</td>
</tr>
<tr>
<td>IFOREST</td>
<td>Low/Medium</td>
<td>Low/Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>
Contours - Old Faithful dataset

GMM  DPGMM  RKDE  PPCA  LSA
Maha  LOF    ABOD  SOD   KL
GWR   OCSVM  IForest
Deep Gaussian Process
autoencoder
Deep Gaussian Process autoencoder

- Unsupervised and probabilistic
- Suitable for any type of data
- Training only requires tensor products
- Inference through stochastic variational inference
- Mini-batch learning
Autoencoders

- Learn a compressed representation of the training data by minimizing the error between the input data and the reconstructed output.
Deep Gaussian Process autoencoders

- Deep probabilistic models
- Composition of functions

\[ f(x) = \left( h^{(N_h-1)} \left( \theta^{(N_h-1)} \right) \circ \ldots \circ h^{(0)} \left( \theta^{(0)} \right) \right)(x) \]
Inference requires calculating the marginal likelihood:

\[
p(X|\theta) = \int p\left( X|F^{(N_L)}, \theta^{(N_L)} \right) \times p\left( F^{(N_L)}|F^{(N_L-1)}, \theta^{(N_L-1)} \right) \times \ldots \times \]
\[
p\left( F^{(1)}|F^{(N_0)}, \theta^{(0)} \right) \, dF^{(N_L)} \ldots dF^{(1)}
\]
DGPs with Random Features

- GPs are single-layered Neural Nets with an infinite number of hidden units

- Weight-space view of a GP

  \[ F = \Phi W \]

- The priors over the weights are

  \[ p(W) = \mathcal{N}(0, I) \]
Random Feature Expansion of Kernels

- Low-rank approximation of GP covariance functions
- The **RBF kernel** can be approximated using trigonometric functions
  \[
  \Phi_{\text{RBF}} = \sqrt{\frac{\sigma^2}{N_{\text{RF}}}} \left[ \cos (F\Omega), \sin (F\Omega) \right]
  \text{ with } p(\Omega \cdot j | \theta) = \mathcal{N} \left(0, \Lambda^{-1}\right)
  \]
- The first order **Arc-Cosine kernel** can be approximated using Rectified Linear Units (ReLU)
  \[
  \Phi_{\text{ARC}} = \sqrt{\frac{2\sigma^2}{N_{\text{RF}}}} \max (0, F\Omega)
  \text{ with } p(\Omega \cdot j | \theta) = \mathcal{N} \left(0, \Lambda^{-1}\right)
  \]
- Approximated multivariate GPs are **Bayesian linear models**
DGP-AEs with RFs (2 layers)

\[ X \xrightarrow{\phi^{(0)}} F^{(1)} = Z \xrightarrow{\phi^{(1)}} F^{(2)} \xrightarrow{\Omega^{(1)}} W^{(1)} \xrightarrow{\Omega^{(0)}} \Theta^{(0)} \xrightarrow{\Theta^{(1)}} X \]
• Define $\psi = (\Omega^{(0)}, \ldots, \Omega^{(L)}, W^{(0)}, \ldots, W^{(L)})$

• Lower bound on the marginal likelihood:

$$\log [p(X|\theta)] \geq \mathbb{E}_{q(\psi)} (\log [p(X|\psi, \theta)]) - D_{KL} [q(\psi)\|p(\psi)]$$

where $q(\psi)$ approximates $p(\psi|X, \theta)$

• $D_{KL}$ computable analytically if $q$ and $p$ are Gaussian

• We assume an approximate factorized Gaussian distribution $q(\psi)$
DGPs with RFs - Stochastic variational inference

- **Stochastic unbiased** estimate of the expectation term
  
  - **Mini-batch**
    
    $\mathbb{E}_{q(\psi)} (\log [p(X|\psi, \theta)]) \approx \frac{n}{m} \sum_{k \in \mathcal{I}_m} \mathbb{E}_{q(\psi)} (\log [p(x_k|\psi, \theta)])$
  
  - **Monte Carlo sampling**
    
    $\mathbb{E}_{q(\psi)} (\log [p(x_k|\psi, \theta)]) \approx \frac{1}{N_{MC}} \sum_{r=1}^{N_{MC}} \log [p(x_k|\tilde{\psi}_r, \theta)]$
  
    with $\tilde{\psi}_r \sim q(\psi)$

- The derivative of the estimate yields a **stochastic gradient**
Reparameterization trick

\[
(\tilde{W}_r^{(l)})_{ij} = s_{ij}^{(l)} \epsilon_{rij}^{(l)} + m_{ij}^{(l)}
\]

with \( \epsilon_{rij}^{(l)} \sim \mathcal{N}(0, 1) \)
• Predictive distribution

\[ p(x_*|X, \theta) = \int p(x_*|\psi, \theta)p(\psi|X, \theta)\,d\psi \]

• Approximation

\[ p(x_*|X, \theta) \approx \int p(x_*|\psi, \theta)q(\psi)\,d\psi \approx \frac{1}{N_{MC}} \sum_{r=1}^{N_{MC}} p(x_*|\tilde{\psi}_r, \theta) \]
• Model inference for mixed-type features

- Normal: \( p(x_{[G]}|f^{(N_L)}) = \mathcal{N}(x_{[G]}|f_{[G]}^{(N_L)}, \sigma_{[G]}^2) \)

- Softmax: \( p((x_{[C]})_j|f^{(N_L)}) = \frac{\exp[f_{[C]}^{(N_L)}]_j}{\sum_i \exp[f_{[C]}^{(N_L)})_i]} \)

- Combined likelihood: \( p(x|f^{(N_L)}) = \prod_k p(x_{[k]}|f^{(N_L)}) \)
DGP-AE Evaluation
- **Isolation Forest**: IFOREST (Liu et al. 2008)
- **Robust Kernel Density Estimation**: RKDE (Kim and Scott 2012)
- **Feedforward Autoencoders**: AE-1, AE-5
- **Variational Autoencoders**: VAE-1, VAE-2 (Kingma and Welling 2014)
- **Variational Auto-Encoded DGP**: VAE-DGP-2 (Dai et al. 2016)
- **Neural Autoregressive Distribution Estimator**: NADE-2 (Uria et al. 2016)
Method comparison

- 11 datasets, mean area under the precision-recall curve (MAP)
- Some datasets contain over 3 millions samples and 100 features
- DGP-AE achieves the best results for novelty detection
- Softmax accurately models categorical variables

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G-1</td>
<td>G-2</td>
<td>GS-1</td>
<td>GS-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAMMOGRAPHY</td>
<td>0.222</td>
<td>0.183</td>
<td>0.222</td>
<td>0.183</td>
<td>0.221</td>
<td>0.118</td>
<td>0.075</td>
<td>0.119</td>
<td>0.148</td>
<td>0.193</td>
<td>0.231</td>
<td>0.244</td>
</tr>
<tr>
<td>MAGIC-GAMMA-SUB</td>
<td>0.260</td>
<td>0.340</td>
<td>0.260</td>
<td>0.340</td>
<td>0.235</td>
<td>0.253</td>
<td>0.125</td>
<td>0.230</td>
<td>0.305</td>
<td>0.398</td>
<td>0.402</td>
<td>0.290</td>
</tr>
<tr>
<td>WINE-QUALITY</td>
<td>0.224</td>
<td>0.203</td>
<td>0.224</td>
<td>0.203</td>
<td>0.075</td>
<td>0.106</td>
<td>0.042</td>
<td>0.064</td>
<td>0.124</td>
<td>0.102</td>
<td>0.051</td>
<td>0.059</td>
</tr>
<tr>
<td>MUSHROOM-SUB</td>
<td>0.811</td>
<td>0.677</td>
<td>0.940</td>
<td>0.892</td>
<td>0.636</td>
<td>0.725</td>
<td>0.331</td>
<td>0.758</td>
<td>0.479</td>
<td>0.596</td>
<td>0.839</td>
<td>0.546</td>
</tr>
<tr>
<td>CAR</td>
<td>0.050</td>
<td>0.061</td>
<td>0.043</td>
<td>0.067</td>
<td>0.045</td>
<td>0.044</td>
<td>0.032</td>
<td>0.071</td>
<td>0.050</td>
<td>0.030</td>
<td>0.034</td>
<td>0.041</td>
</tr>
<tr>
<td>GERMAN-SUB</td>
<td>0.066</td>
<td>0.077</td>
<td>0.106</td>
<td>0.098</td>
<td>0.113</td>
<td>0.065</td>
<td>0.103</td>
<td>0.104</td>
<td>0.062</td>
<td>0.118</td>
<td>0.109</td>
<td>0.079</td>
</tr>
<tr>
<td>PNR</td>
<td>0.190</td>
<td>0.172</td>
<td>0.190</td>
<td>0.172</td>
<td>0.201</td>
<td>0.059</td>
<td>0.107</td>
<td>0.100</td>
<td>0.106</td>
<td>0.006</td>
<td>0.146</td>
<td>0.124</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>0.756</td>
<td>0.752</td>
<td>0.810</td>
<td>0.835</td>
<td>0.509</td>
<td>0.563</td>
<td>0.510</td>
<td>0.532</td>
<td>0.760</td>
<td>0.373</td>
<td>0.585</td>
<td>0.564</td>
</tr>
<tr>
<td>SHARED-ACCESS</td>
<td>0.692</td>
<td>0.738</td>
<td>0.692</td>
<td>0.738</td>
<td>0.668</td>
<td>0.546</td>
<td>0.766</td>
<td>0.471</td>
<td>0.527</td>
<td>0.239</td>
<td>0.783</td>
<td>0.746</td>
</tr>
<tr>
<td>PAYMENT-SUB</td>
<td>0.173</td>
<td>0.173</td>
<td>0.168</td>
<td>0.168</td>
<td>0.137</td>
<td>0.157</td>
<td>0.129</td>
<td>0.175</td>
<td>0.143</td>
<td>0.101</td>
<td>0.180</td>
<td>0.142</td>
</tr>
<tr>
<td>AIRLINE</td>
<td>0.081</td>
<td>0.079</td>
<td>0.081</td>
<td>0.079</td>
<td>0.060</td>
<td>0.063</td>
<td>0.059</td>
<td>0.068</td>
<td>0.074</td>
<td>0.064</td>
<td>-</td>
<td>0.069</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>0.344</td>
<td>0.338</td>
<td>0.366</td>
<td>0.370</td>
<td>0.284</td>
<td>0.264</td>
<td>0.222</td>
<td>0.262</td>
<td>0.270</td>
<td>0.216</td>
<td>0.336</td>
<td>0.284</td>
</tr>
</tbody>
</table>
Convergence monitoring - Networks

- MAP and mean log-likelihood (MLL). The higher the better

- DGP-AE shows the best likelihood
- MAP quickly stabilizes while the likelihood is continuously refined
• Correlation between a higher test likelihood and a higher MAP

• Moderately deep networks capture the complexity of data without an important convergence overhead
Convergence monitoring - GPs

- Dimensionality reduction capabilities of a DGP-AE-G-2

- Increasing the number of GPs results in a slower convergence
- 5 GPs achieve good novelty detection performance despite a significant dimensionality reduction
Latent representation

- Meaningful low-dimensional representations, comparable with state-of-the-art manifold learning methods
Conclusions
Conclusions

- Novel probabilistic models for novelty detection
  - DPMM
    - Interpretable, fast and accurate modeling of mixed-type features
    - Clustering, not suitable for numeric-only data
  - DGP-AE
    - Competitive with SoA and DNN-based novelty detection methods
    - Good dimensionality reduction abilities
    - Tractable and scalable inference
    - Suitable to model mixed-types features

- Experimental surveys for novelty detection
  - Numerical, mixed-type and temporal data
  - No clear winner
  - Metric comparison
  - Recommendations based on datasets’ characteristics
  - Highlighted scalability pitfalls
Industrial contributions

- Generic **benchmarking platform**

- Comparative study used internally

- Thousands of DPMMs running to **raise alerts**

- **Recommendations** for action sequences + **integration** ready
Research contributions

- Journals
  Under review

- Journal & conference
  Presented at *ECML-PKDD*, 2018

- Workshop
Future work

- Mini-batch training for DPMM
- **Generative DGP-AE**
- Model discrete event sequences with **structured DGP-AE**
- Image-based novelty detection
- **Distributed** and **GPU** computing, **streaming** data
Thank you
Exponential family of distributions

- **Density**
  - $h(x)$ function
  - $\eta^*$ natural parameter
  - $T(x)$ sufficient statistics
  - $a(\eta^*)$ normalization factor

  $$p(x|\eta^*) = h_l(x) \exp \left( \eta^*^T T(x) - a_l(\eta^*) \right)$$ (2)

- **Conjugate prior, based on the previous likelihood**

  $$p(\eta^*|\lambda) = h_p(\eta^*) \exp \left( \lambda_1^T \eta^* + \lambda_2(-a_l(\eta^*)) - a_p(\lambda) \right),$$ (3)

  - Same dimensionality for $\lambda$ and $\eta^*$, $\lambda_2$ is a scalar

- **Posterior**

  $$p(\eta^*|\tau) = h_p(\eta^*) \exp \left( \tau_1^T \eta^* + \tau_2(-a_l(\eta^*)) - a_p(\tau) \right).$$ (4)
Dirichlet Process Mixture Model
● Mean-field variational inference
  ● The optimal solution $q_j^*$ for each of the factors $q_j$ is:

  $$\ln q_j^*(\textbf{w}_j|\textbf{X}) = \mathbb{E}_{i\neq j}[\ln p(\textbf{X}, \textbf{W})] + \text{const}$$  \hspace{1cm} (5)

● Truncated representation of a DP mixture
  ● $\pi_k(v) = v_k \prod_{j=1}^{k-1} (1 - v_j)$
  ● $\pi_k(v) = 0$ for $k > K$, which is achieved by setting $v_K = 1$
  ● $q_{\alpha_K, \beta_K}(v_K = 1) = 1$