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

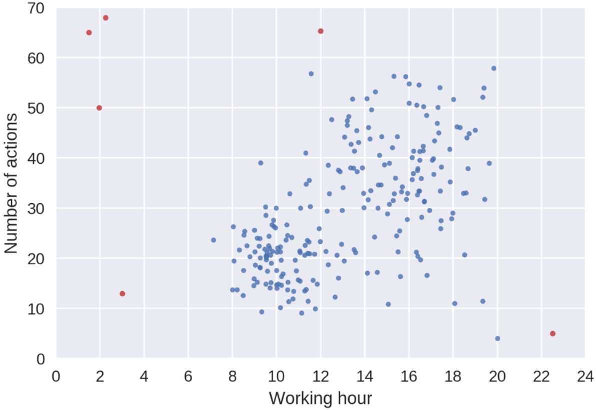
Payment frauds

Malicious bots

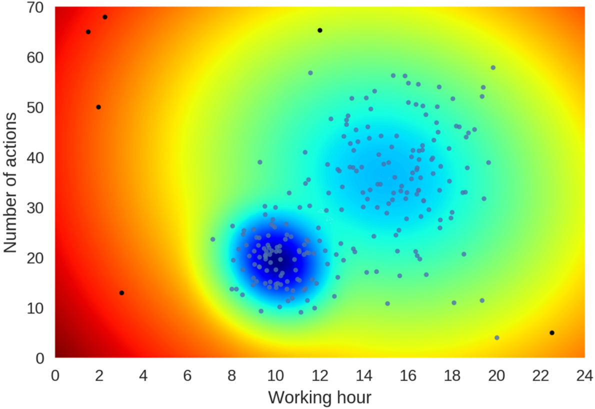
Anomaly detection

- 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



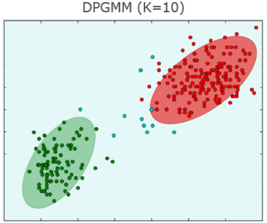
Industrial constraints and research challenges

Proactive, unlabelled data	Novelty detection
Anomalies in training data	Robustness
Continuous scoring	Probabilistic method
Numerical & categorical data	Variational learning of joint distributions
Scalable and distributed	Mini-batch learning
White-box model	Interpretable
Little tuning	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

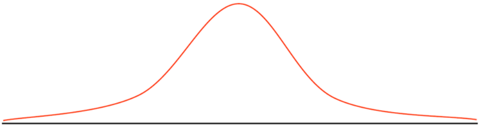


- **A product of exponential-family distributions is in the exponential-family**
- Probabilistic, mini-batch training, categorical support, clustering

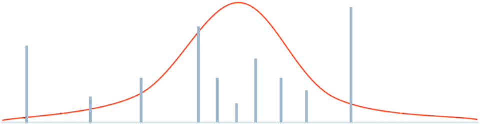
Dirichlet Process

- Bayesian nonparametric model
- **Distribution over distributions**

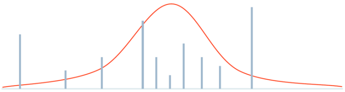
- Consider a Gaussian G_0:



- $G \sim DP(\alpha, G_0)$

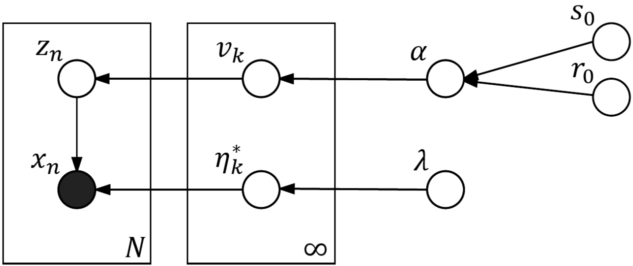


Stick-breaking process

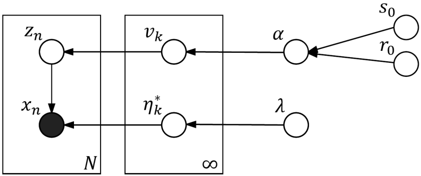


- Constructive way of forming G
- Weights $\pi_k(\boldsymbol{v}) = \boldsymbol{v_k} \prod_{j=1}^{k-1}(1 - v_j)$, with $\boldsymbol{v_k} \sim \mathrm{Beta}(1, \alpha)$
- $G \sim DP(\alpha, G_0) \Leftrightarrow G = \prod_{k=i}^{\infty} \pi_k \delta_{\theta_k}$
  - $\theta_k^* \sim G_0$
  - δ_{θ_k} is the indicator function which evaluates to zero everywhere, except for $\delta_{\theta_k}(\theta_k) = 1$

Dirichlet Process Mixture Model



Dirichlet Process Mixture Model



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

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

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

4. **Assign** the data to the components: $z_n | \boldsymbol{v} \sim \mathrm{Mult}(\pi(\boldsymbol{v}))$
   Generate the observations: $\boldsymbol{x}_n | z_n \sim p(\boldsymbol{x}_n | \eta^*_{z_n})$

DP mixture inference

- Predictive density: $p(\boldsymbol{x}_{N+1}|\boldsymbol{X},\theta) = \int p(\boldsymbol{x}_{N+1}|\boldsymbol{W})p(\boldsymbol{W}|\boldsymbol{X},\theta)d\boldsymbol{W}$
- Intractable posterior over the latent variables $p(\boldsymbol{W}|\boldsymbol{X},\theta)$
- 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
  $q(\boldsymbol{v}, \eta^*, \boldsymbol{z}, w) = q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{v}) \cdot q_{\tau}(\eta^*) \cdot q_r(\boldsymbol{z}) \cdot q_{g_1,g_2}(w)$
  - $q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{v})$: product of **Beta**
  - $q_{\tau}(\eta^*)$: product of distributions in the **exponential family**
  - $q_r(\boldsymbol{z})$: product of **multinomials** on cluster assignment variable $\boldsymbol{z}$
  - $q_{g_1,g_2}(w)$: Γ distribution
- Hyperparameters: $\boldsymbol{\lambda}$, α, s_0 and r_0
- Latent variables: $\boldsymbol{v}$, η^*, $\boldsymbol{z}$ and w
- Variational parameters: α_k, β_k, $\boldsymbol{\tau_k}$, r_{nk}, g_1 and g_2

Variational inference

1. Initialize the model parameters

2. Optimize the **variational parameters** to minimize
   $D_{KL}(q(\boldsymbol{W})||p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{\theta})) = \mathbb{E}_q[\ln q(\boldsymbol{W})] - \mathbb{E}_q[\ln p(\boldsymbol{W},\boldsymbol{X}|\boldsymbol{\theta})] + \ln p(\boldsymbol{X}|\boldsymbol{\theta})$
   Equivalent to maximizing $\ln p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \mathbb{E}_q[\ln p(\boldsymbol{W},\boldsymbol{X}|\boldsymbol{\theta})] - \mathbb{E}_q[\ln q(\boldsymbol{W})]$

3. Compute the **expectation** of $p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{\theta})$ under $q(\boldsymbol{W}|\boldsymbol{X})$, e.g.
   $\ln q^*_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{v}) = \mathbb{E}_{\boldsymbol{\eta}^*,\boldsymbol{z},w}[\ln p(\boldsymbol{X},\boldsymbol{v},\boldsymbol{\eta}^*,\boldsymbol{z},w)] + c = \prod_{k=1}^{K-1} \mathrm{Beta}(\alpha_k,\beta_k)$

4. Compute the **geometric means**
   - $\mathbb{E}[\ln \boldsymbol{v_k}]$, $\mathbb{E}[\ln(1-\boldsymbol{v_k})]$, $\mathbb{E}[\boldsymbol{\eta}^*]$, $\mathbb{E}[-a(\boldsymbol{\eta}^*)]$, $\mathbb{E}[z_{nk}]$, $\mathbb{E}[w]$ and $\mathbb{E}[\ln w]$
   - Update the model parameters to maximize the expectation of $p(\boldsymbol{W},\boldsymbol{X}|\boldsymbol{\theta})$ under $q(\boldsymbol{W}|\boldsymbol{X})$

Lower bound

- Nondecreasing, used for **convergence monitoring**

$$
\begin{aligned}
\ln p(\boldsymbol{X}|\boldsymbol{\theta}) &\geq \mathbb{E}_q[\ln p(\boldsymbol{W}, \boldsymbol{X}|\boldsymbol{\theta})] - \mathbb{E}_q[\ln q(\boldsymbol{W})] \\
&\geq \mathbb{E}_q[\ln p(\boldsymbol{X}, \boldsymbol{z}, \eta^*, \boldsymbol{v}, w|\boldsymbol{\theta})] - \mathbb{E}_q[\ln q(\boldsymbol{z}, \eta^*, \boldsymbol{v}, w)] \\
&\geq \mathbb{E}_q[\ln p(\boldsymbol{X}|\boldsymbol{z}, \eta^*)] + \mathbb{E}_q[\ln p(\boldsymbol{z}|\boldsymbol{v})] + \mathbb{E}_q[\ln p(\eta^*|\boldsymbol{\lambda})] \\
&\quad + \mathbb{E}_q[\ln p(\boldsymbol{v}|w)] + \mathbb{E}_q[\ln p(w|s_0, r_0)] - \mathbb{E}_q[\ln q_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{v})] \\
&\quad - \mathbb{E}_q[\ln q_{\boldsymbol{\tau}}(\eta^*)] - \mathbb{E}_q[\ln q_{r}(\boldsymbol{z})] - \mathbb{E}_q[\ln q_{g_1,g_2}(w)]
\end{aligned}
$$

Predictive distribution

$$
\begin{aligned}
p(\boldsymbol{x}_{N+1}|\boldsymbol{X}, \boldsymbol{\theta}) &= \int \sum_{k=1}^{\infty} \pi_k(\boldsymbol{v}) p(\boldsymbol{x}_{N+1}|\boldsymbol{\eta}_k^*) dp(\boldsymbol{v}, \boldsymbol{\eta}^*|\boldsymbol{X}, \boldsymbol{\theta}) \\
&\approx \sum_{k=1}^{K} \mathbb{E}_q[\pi_k(\boldsymbol{v})] \mathbb{E}_q[p(\boldsymbol{x}_{N+1}|\boldsymbol{\eta}_k^*)].
\end{aligned} \tag{1}
$$

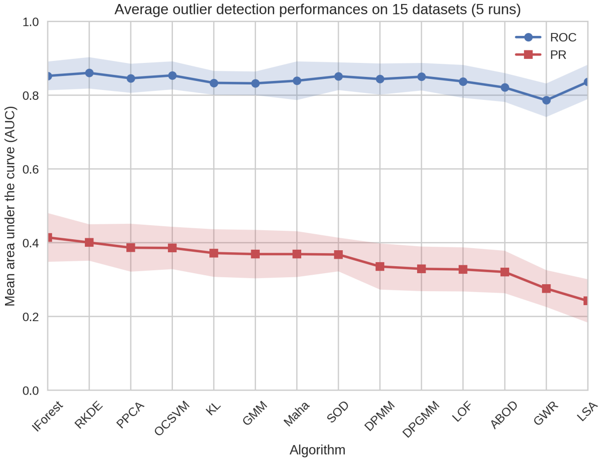
- Analytically, we obtain $\mathbb{E}_q[\pi_k(\boldsymbol{v})] = \frac{\alpha_k}{\alpha_k+\beta_k} \prod_{i=1}^{k-1} \left(1 - \frac{\alpha_i}{\alpha_i+\beta_i}\right)$
- **Monte Carlo sampling** is used to estimate the density
  1. Draw m samples from $q_{\boldsymbol{\tau}}^*(\boldsymbol{\eta}^*)$
  2. Compute each $p(\boldsymbol{x}_{N+1}|\boldsymbol{\eta}^*)$
  3. Average the resulting m likelihoods

Experimental survey

Algorithms

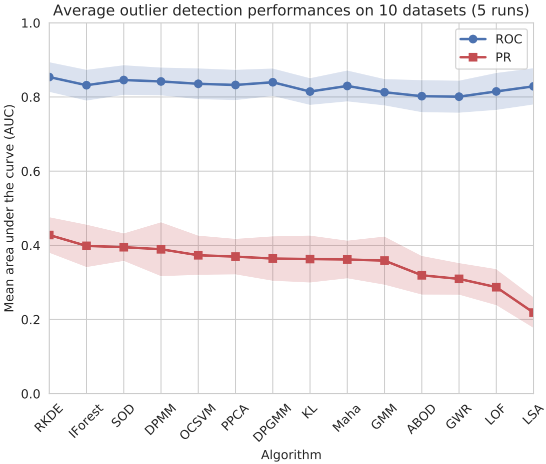
Gaussian Mixture Model Dirichlet Process Mixture Model Robust Kernel Density Estimation Least-Squares Anomaly Detection Probabilistic PCA	Probabilistic
Mahalanobis	Distance-based
Local Outlier Factor Angle-Based Outlier Detection Subspace Outlier Detection	Nearest neighbors
One-class SVM	Domain-based
Grow When Required network	Neural networks
Kullback-Leibler Divergence	Information theoretic
Isolation Forest	Isolation

Results



Algorithm	GMM	DPGMM	DPMM	RKDE	PPCA	LSA	MAHA	LOF	ABOD	SOD	KL	GWR	OCSVM	IFOREST
PR AUC	6	12	9	2	5	14	7	11	10	4	8	13	3	1
ROC AUC	11	4	5	1	7	9	6	10	13	8	12	14	3	2

Results - No classification datasets



Algorithm	GMM	**DPGMM**	**DPMM**	RKDE	PPCA	LSA	MAHA	LOF	ABOD	SOD	KL	GWR	OCSVM	IFOREST
PR AUC	5	**4**	**3**	2	6	14	9	12	11	1	8	13	10	7
ROC AUC	13	**5**	**2**	1	6	9	7	10	14	4	12	11	3	8

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

AUC	ROC	PR
Prediction 1	0.75	0.50
Prediction 2	0.74	0.10

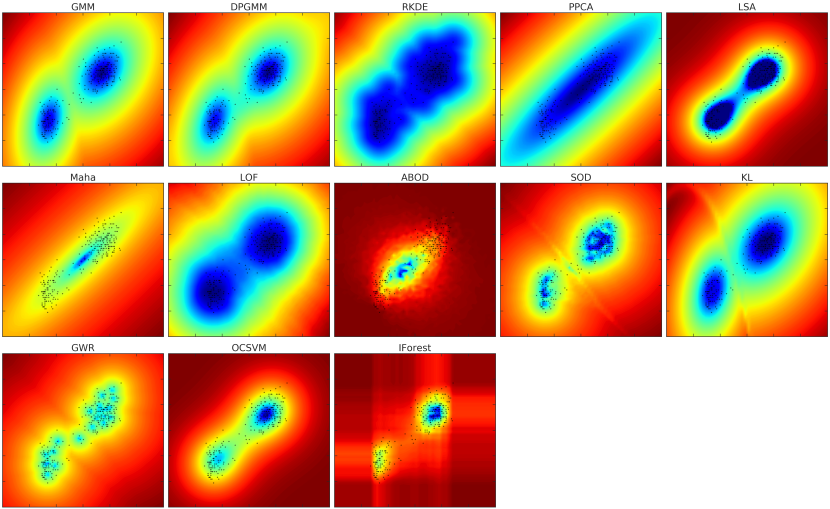
- **The ROC AUC downplays the impact of false positives when negative observations are over-represented**

Scalability

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

	Training/prediction time		Mem. usage		Robustness		
Algorithm	↗ Samples	↗ Features	↗ Samples	↗ Features	↗ Noise	High dim.	Stability
GMM	Low/Low	Medium/Medium	Low	Medium	High	Medium	Medium
BGM	Low/Low	Medium/Medium	Low	Medium	High	Medium	High
DPGMM	Medium/Low	High/High	Low	High	High	High	High
RKDE	High/High	High/High	High	Low	High	High	High
PPCA	Low/Low	High/Low	Low	Low	High	Medium	Medium
LSA	Low/Medium	Low/Low	Medium	Low	Low	Low	Medium
MAHA	Low/Medium	Medium/Low	Low	Medium	Medium	Low	High
LOF	High/High	Low/Low	High	Low	Medium	High	High
ABOD	Low/High	Low/Medium	Low	Low	Medium	Low	Medium
SOD	High/High	Low/Medium	High	Low	Low	High	Medium
KL	Low/Medium	Low/Medium	Low	Medium	High	Medium	High
GWR	Medium/Medium	Medium/Low	Low	Low	Low	High	Medium
OCSVM	High/High	Low/Low	Low	Low	Low	High	High
IFOREST	Low/Medium	Low/Low	Medium	Low	High	High	Medium

Contours - Old Faithful dataset



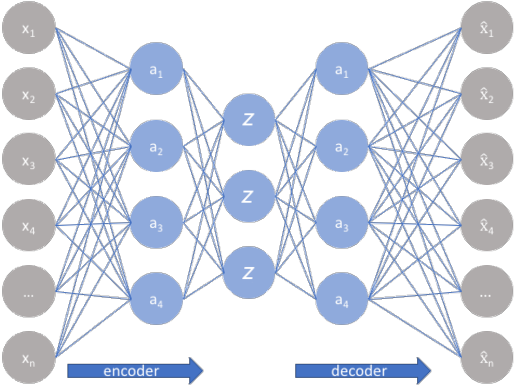
Deep Gaussian Process autoencoder

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

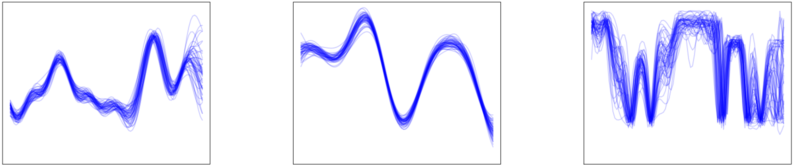


Deep Gaussian Process autoencoders

- Deep probabilistic models
- Composition of functions

$$\mathbf{f}(\mathbf{x}) = \left(\mathbf{h}^{(N_\mathrm{h}-1)}\left(\theta^{(N_\mathrm{h}-1)}\right) \circ \ldots \circ \mathbf{h}^{(0)}\left(\theta^{(0)}\right)\right)(\mathbf{x})$$

$\mathbf{h}^{(0)}(\mathbf{x})$ $\quad$ $\mathbf{h}^{(1)}(\mathbf{x})$ $\quad$ $\mathbf{h}^{(1)}\left(\mathbf{h}^{(0)}(\mathbf{x})\right)$



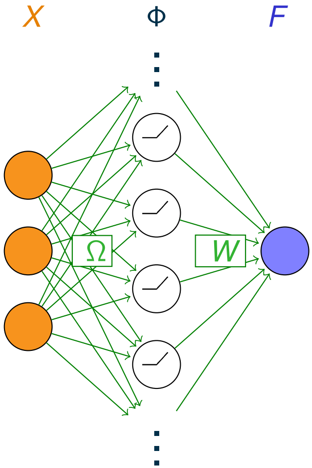
Marginal likelihood

- Inference requires calculating the marginal likelihood:

$$
\begin{aligned}
p(X|\boldsymbol{\theta}) \quad = \quad & \int p\left(X|F^{(N_{\mathrm{L}})}, \boldsymbol{\theta}^{(N_{\mathrm{L}})}\right) \times \\
& p\left(F^{(N_{\mathrm{L}})}|F^{(N_{\mathrm{L}}-1)}, \boldsymbol{\theta}^{(N_{\mathrm{L}}-1)}\right) \times \ldots \times \\
& p\left(F^{(1)}|F^{(N_0)}, \boldsymbol{\theta}^{(0)}\right) dF^{(N_{\mathrm{L}})} \ldots dF^{(1)}
\end{aligned}
$$

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

$$p\left(W_{\cdot i}\right) = \mathcal{N}\left(\mathbf{0}, \mathrm{I}\right)$$

Random Feature Expansion of Kernels

- Low-rank approximation of GP covariance functions
- The **RBF kernel** can be approximated using **trigonometric functions**

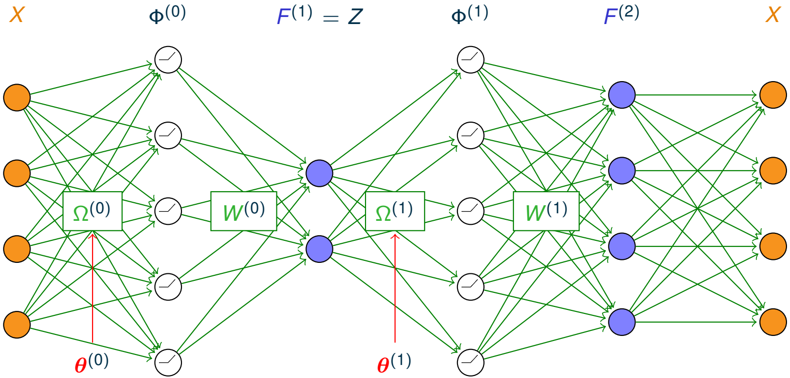
$$\Phi_{\text{RBF}} = \sqrt{\frac{\sigma^2}{N_{\text{RF}}}} \left[\cos\left(F\Omega\right), \sin\left(F\Omega\right)\right] \quad \text{with} \quad p\left(\Omega_{\cdot j} \middle| \boldsymbol{\theta}\right) = \mathcal{N}\left(\mathbf{0}, \Lambda^{-1}\right)$$

- The first order **Arc-Cosine kernel** can be approximated using **Rectified Linear Units (ReLU)**

$$\Phi_{\text{ARC}} = \sqrt{\frac{2\sigma^2}{N_{\text{RF}}}} \max\left(\mathbf{0}, F\Omega\right) \quad \text{with} \quad p\left(\Omega_{\cdot j} \middle| \boldsymbol{\theta}\right) = \mathcal{N}\left(\mathbf{0}, \Lambda^{-1}\right)$$

- Approximated multivariate GPs are **Bayesian linear models**

DGP-AEs with RFs (2 layers)



DGP-AE with RFs - Stochastic Variational Inference

- Define $\Psi = \left(\Omega^{(0)}, \ldots, \Omega^{(L)}, W^{(0)}, \ldots, W^{(L)}\right)$
- Lower bound on the marginal likelihood:

$$\log\left[p(X|\boldsymbol{\theta})\right] \geq \mathbb{E}_{q(\Psi)}\left(\log\left[p\left(X|\Psi, \boldsymbol{\theta}\right)\right]\right) - \mathrm{D}_{\mathrm{KL}}\left[q(\Psi) \| p\left(\Psi\right)\right]$$

  where $q(\Psi)$ approximates $p(\Psi|X, \boldsymbol{\theta})$
- 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**

$$\mathbb{E}_{q(\Psi)}\left(\log\left[p(X|\Psi,\boldsymbol{\theta})\right]\right) \approx \frac{n}{m}\sum_{\mathrm{k}\in\mathcal{I}_m}\mathbb{E}_{q(\Psi)}\left(\log[p(\mathbf{x}_{\mathrm{k}}|\Psi,\boldsymbol{\theta})]\right)$$

  - **Monte Carlo sampling**

$$\mathbb{E}_{q(\Psi)}\left(\log\left[p(x_{\mathrm{k}}|\Psi,\boldsymbol{\theta})\right]\right) \approx \frac{1}{N_{\mathrm{MC}}}\sum_{r=1}^{N_{\mathrm{MC}}}\log[p(x_{\mathrm{k}}|\tilde{\Psi}_r,\boldsymbol{\theta})]$$

  with $\tilde{\Psi}_r \sim q(\Psi)$

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

DGPs with RFs - Stochastic variational inference

- Reparameterization trick

$$\left(\tilde{W}_r^{(l)}\right)_{ij} = s_{ij}^{(l)} \epsilon_{rij}^{(l)} + m_{ij}^{(l)}$$

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

Predictions

- Predictive distribution

$$p(\mathbf{x}_*|X,\boldsymbol{\theta}) = \int p(\mathbf{x}_*|\Psi,\boldsymbol{\theta})p(\Psi|X,\boldsymbol{\theta})d\Psi$$

- Approximation

$$\begin{aligned} p(\mathbf{x}_*|X,\boldsymbol{\theta}) &\approx \int p(\mathbf{x}_*|\Psi,\boldsymbol{\theta})q(\Psi)d\Psi \\ &\approx \frac{1}{N_{\mathrm{MC}}}\sum_{r=1}^{N_{\mathrm{MC}}} p(\mathbf{x}_*|\tilde{\Psi}_r,\boldsymbol{\theta}) \end{aligned}$$

Likelihood functions

- Model inference for mixed-type features
  - Normal: $p(x_{[G]}|\mathbf{f}^{(N_{\mathrm{L}})}) = \mathcal{N}(x_{[G]}|f_{[G]}^{(N_{\mathrm{L}})}, \sigma_{[G]}^2)$
  - Softmax: $p((\mathbf{x}_{[C]})_j|\mathbf{f}^{(N_{\mathrm{L}})}) = \frac{\exp[(f_{[C]}^{(N_{\mathrm{L}})})_j]}{\sum_i \exp[(f_{[C]}^{(N_{\mathrm{L}})})_i]}$
  - Combined likelihood: $p(\mathbf{x}|\mathbf{f}^{(N_{\mathrm{L}})}) = \prod_k p(\mathbf{x}_{[k]}|\mathbf{f}^{(N_{\mathrm{L}})})$

DGP-AE Evaluation

Algorithms

- **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)

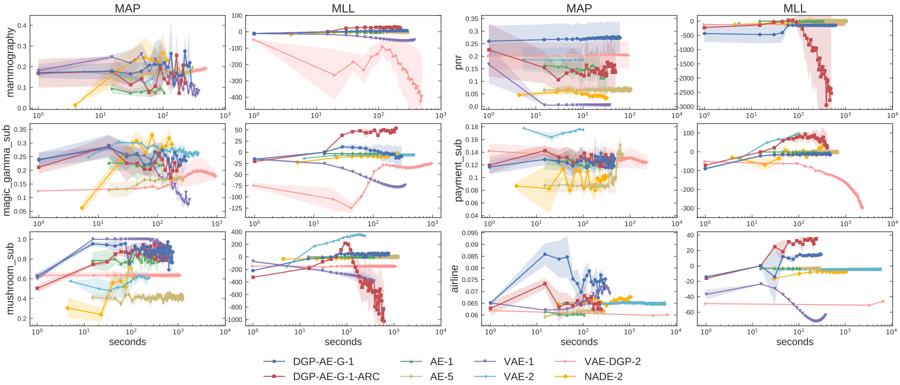
Method comparison

- 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

	DGP-AE G-1	DGP-AE G-2	DGP-AE GS-1	DGP-AE GS-2	VAE-DGP-2	AE-1	AE-5	VAE-1	VAE-2	NADE-2	RKDE	IFOREST
MAMMOGRAPHY	**0.222**	0.183	**0.222**	0.183	**0.221**	0.118	0.075	0.119	0.148	0.193	**0.231**	**0.244**
MAGIC-GAMMA-SUB	0.260	0.340	0.260	0.340	0.235	0.253	0.125	0.230	0.305	**0.398**	**0.402**	0.290
WINE-QUALITY	**0.224**	**0.203**	**0.224**	**0.203**	0.075	0.106	0.042	0.064	0.124	0.102	0.051	0.059
MUSHROOM-SUB	0.811	0.677	**0.940**	0.892	0.636	0.725	0.331	0.758	0.479	0.596	0.839	0.546
CAR	0.050	0.061	0.043	0.067	0.045	0.044	0.032	**0.071**	0.050	0.030	0.034	0.041
GERMAN-SUB	0.066	0.077	**0.106**	0.098	**0.113**	0.065	**0.103**	**0.104**	0.062	**0.118**	**0.109**	0.079
PNR	**0.190**	0.172	**0.190**	0.172	**0.201**	0.059	0.107	0.100	0.106	0.006	0.146	0.124
TRANSACTIONS	0.756	0.752	**0.810**	**0.835**	0.509	0.563	0.510	0.532	0.760	0.373	0.585	0.564
SHARED-ACCESS	0.692	0.738	0.692	0.738	0.668	0.546	**0.766**	0.471	0.527	0.239	**0.783**	0.746
PAYMENT-SUB	**0.173**	**0.173**	0.168	0.168	0.137	0.157	0.129	**0.175**	0.143	0.101	**0.180**	0.142
AIRLINE	**0.081**	**0.079**	**0.081**	**0.079**	0.060	0.063	0.059	0.068	0.074	0.064	-	0.069
AVERAGE	**0.344**	**0.338**	**0.366**	**0.370**	0.284	0.264	0.222	0.262	0.270	0.216	0.336	0.284

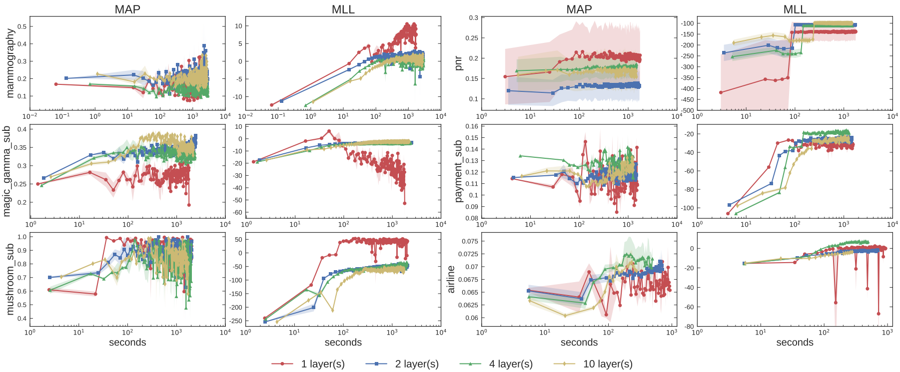
Convergence monitoring - Networks

- 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

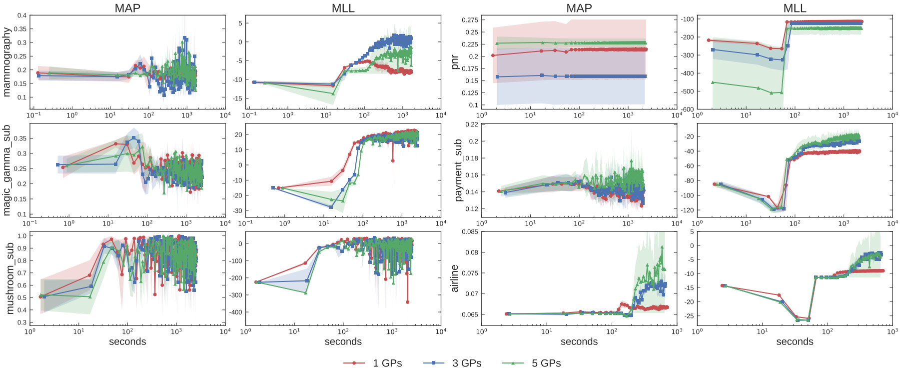
Convergence monitoring - Depth



- Correlation between a higher test likelihood and a higher MAP
- Moderately deep networks capture the complexity of data without an important convergence overhead

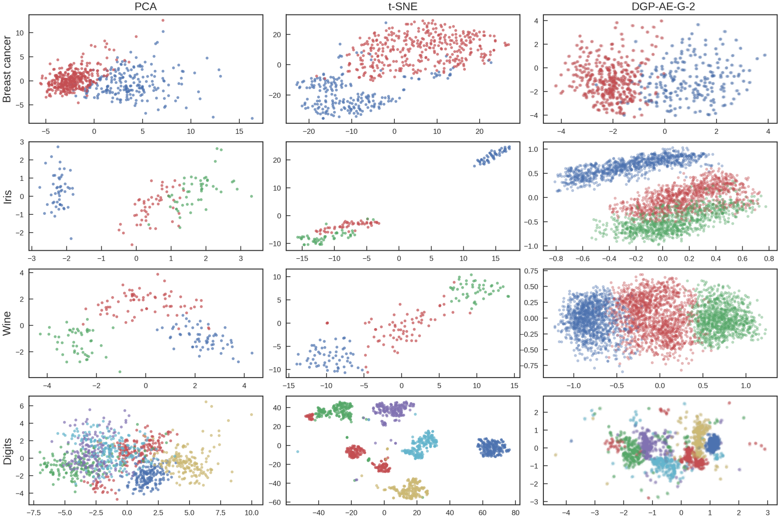
Convergence monitoring - GPs

- 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**

  R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, A comparative evaluation of outlier detection algorithms: experiments and analyses, ***Pattern Recognition***, vol. 74, pp. 406–421, 2018

  R. Domingues, P. Michiardi, J. Barlet, and M. Filippone, A comparative evaluation of novelty detection algorithms for discrete sequences, ***Artificial Intelligence Review***, vol. 52, no. 1, 2019
  Under review

- **Journal & conference**

  R. Domingues, P. Michiardi, J. Zouaoui, and M. Filippone, Deep gaussian process autoencoders for novelty detection, ***Machine Learning***, 2018
  Presented at ***ECML-PKDD***, 2018

- **Workshop**

  R. Domingues, F. Buonora, R. Senesi, and O. Thonnard, An application of unsupervised fraud detection to passenger name records, in *2016 46th Annual IEEE/IFIP International Conference on* ***Dependable Systems and Networks*** *Workshop (DSN-W)*, 2016, pp. 54–59

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
  - $h(\boldsymbol{x})$ function
  - η^* natural parameter
  - $T(\boldsymbol{x})$ sufficient statistics
  - $a(\eta^*)$ normalization factor

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

- Conjugate prior, based on the previous likelihood

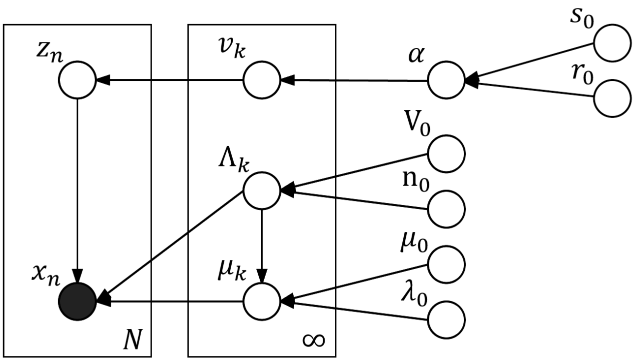
$$p(\eta^*|\boldsymbol{\lambda}) = h_p(\eta^*) \exp\left(\boldsymbol{\lambda}_1^T \eta^* + \lambda_2(-a_l(\eta^*)) - a_p(\boldsymbol{\lambda})\right), \tag{3}$$

  - Same dimensionality for $\boldsymbol{\lambda}$ and η^*, λ_2 is a scalar

- Posterior

$$p(\eta^*|\boldsymbol{\tau}) = h_p(\eta^*) \exp\left(\boldsymbol{\tau}_1^T \eta^* + \tau_2(-a_l(\eta^*)) - a_p(\boldsymbol{\tau})\right). \tag{4}$$

Dirichlet Process Mixture Model



Dirichlet Process Mixture Model

- Mean-field variational inference
  - The optimal solution q_j^* for each of the factors q_j is:

$$\ln q_j^*(\boldsymbol{W}_j|\boldsymbol{X}) = \mathbb{E}_{i\neq j}[\ln p(\boldsymbol{X}, \boldsymbol{W})] + const \tag{5}$$

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