Scalable Gaussian Processes with Billions of Inducing Inputs via Tensor Train Decomposition

Pavel Izmailov\(^1\) \hspace{1cm} Alexander Novikov\(^2,3\) \hspace{1cm} Dmitry Kropotov\(^4\)

\(^1\)Cornell University
\(^2\)National Research University Higher School of Economics
\(^3\)Institute of Numerical Mathematics RAS
\(^4\)Lomonosov Moscow State University

April 10, 2018
Tensor Train Decomposition [Oseledets 2011]

- Generalizes low rank approximation

**Low-Rank**

\[
A_{3,4} = u_3^T v_4
\]

**Tensor Train**

\[
B_{2,3,1} = u_2^T v_3 w_1
\]

- Doesn’t suffer from curse of dimensionality
- Allows fast implementation of linear algebra operations
ML Applications of TT

- TensorNet: DNN compression
  - Feed Forward [Novikov et al. 2015]
  - Convolutional [Garipov et al. 2016]
  - Recurrent [Yu et al. 2018]

- Markov Random Fields [Novikov et al. 2014]

- Theoretical analysis of RNN expressive power [Khrulkov et al. 2018]

- Discrete VAE [coming soon]
ML Applications of TT

- TensorNet: DNN compression
  - Feed Forward [Novikov et al. 2015]
  - Convolutional [Garipov et al. 2016]
  - Recurrent [Yu et al. 2018]

- Markov Random Fields [Novikov et al. 2014]

- Theoretical analysis of RNN expressive power [Khrulkov et al. 2018]

- Discrete VAE [coming soon]

- TT-GP – Scalable GP framework
Gaussian Processes

Definition
Gaussian process is a collection of random variables, any finite number of which have joint Gaussian distribution.

Posterior distribution of a one-dimensional Gaussian process
Gaussian Processes

Definition
Gaussian process is a collection of random variables, any finite number of which have joint Gaussian distribution.

Posterior distribution of a one-dimensional Gaussian process

In Machine Learning GPs
- Allow automatic tuning of model complexity (non-parametric model)
- Provide principled uncertainty estimates
- Can discover complex non-linear patterns in data
Gaussian Processes

Definition
Gaussian process is a collection of random variables, any finite number of which have joint Gaussian distribution.

In Machine Learning GPs

- Allow automatic tuning of model complexity (non-parametric model)
- Provide principled uncertainty estimates
- Can discover complex non-linear patterns in data
- Exact inference is $O(n^3)$
Inducing Inputs

Approximate posterior distribution based on inducing inputs

- Auxiliary observations that approximate the data
- Allow fast approximate inference
Previous Methods

- Classical methods [e.g. Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013] require $O(nm^2 + m^3)$ computations, $m$ is the number of inducing points
  - Applicable for large $n$ (e.g. $10^6$)
  - Infeasible for large $m \gg 10^3$
Previous Methods

- Classical methods [e.g. Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013] require $O(nm^2 + m^3)$ computations, $m$ is the number of inducing points
  - Applicable for large $n$ (e.g. $10^6$)
  - Infeasible for large $m \gg 10^3$

- KISS-GP [Wilson and Nickisch 2015] leverages the structure in the covariance matrices; requires $O(n + m \log m)$ computations, $m = m_0^D$ and $D$ is the number of features
  - Applicable for large $n$ (e.g. $10^6$) and $m$ (e.g. $10^4$)
  - Infeasible for large $D \gg 4$
Previous Methods

- Classical methods [e.g. Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013] require $O(nm^2 + m^3)$ computations, $m$ is the number of inducing points
  - Applicable for large $n$ (e.g. $10^6$)
  - Infeasible for large $m \gg 10^3$

- KISS-GP [Wilson and Nickisch 2015] leverages the structure in the covariance matrices; requires $O(n + m \log m)$ computations, $m = m_0^D$ and $D$ is the number of features
  - Applicable for large $n$ (e.g. $10^6$) and $m$ (e.g. $10^4$)
  - Infeasible for large $D \gg 4$

- *Tensor Train GP (TT-GP)* extends KISS-GP to high-dimensional problems
  - Applicable for large $n$ (e.g. $10^6$) and $m$ (e.g. $10^8$)
  - Applicable for larger $D$ (e.g. 10)
**ELBO [Hensman et al. 2013]**

Evidence Lower Bound (ELBO) for GP regression:

\[
\log p(y) \geq \sum_{i=1}^{n} \left( \log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma^2) - \frac{1}{2\sigma^2} (\tilde{K}_{ii} + \text{tr}(k_i^T K_{mm}^{-1} \Sigma K_{mm}^{-1} k_i)) \right) - \frac{1}{2} \left( \log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right) \rightarrow \max_{\mu, \Sigma, \theta, \sigma}
\]

where

- \( K_{mm} \in \mathbb{R}^{m \times m} \) is the covariance matrix computed at the inducing points
- \( k_i \in \mathbb{R}^m \) is the vector of covariances between the \( i \)-th training object and the inducing points
- \( \sigma^2 \) is the noise variance
- \( \mu \in \mathbb{R}^m, \Sigma \in \mathbb{R}^{m \times m} \) — variational parameters
- \( \tilde{K}_{ii} = \delta^2 - k_i^T K_{mm}^{-1} k_i \), where \( \delta^2 \) is the prior variance of the process at any point
- \( \theta \) represents kernel hyper-parameters
ELBO

Assume $m$ is very large (e.g. $10^{10}$)

$$
\log p(y) \geq \sum_{i=1}^{n} \left( \log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma^2) - \frac{1}{2\sigma^2} \left( \tilde{K}_{ii} + \text{tr}(k_i^T K_{mm}^{-1} \Sigma K_{mm}^{-1} k_i) \right) \right) - \\
\frac{1}{2} \left( \log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)
$$
Assume $m$ is very large (e.g. $10^{10}$)

$$\log p(y) \geq \sum_{i=1}^{n} \left( \log \mathcal{N}(y_i|w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} (\tilde{K}_{ii} + \text{tr}(w_i^T \Sigma w_i)) \right)$$

$$- \frac{1}{2} \left( \log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

- Set inducing points on a grid
- Assume product kernel
- $K_{mm}$ is in Kronecker product format
- $k_i \approx K_{mm} w_i$, $w_i$ in Kronecker product format
TT-GP (Our Method)

\[
\log p(y) \geq \sum_{i=1}^{n} \left( \log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} (\tilde{K}_{ii} + \text{tr}(w_i^T \Sigma w_i)) \right)
\]

\[
-\frac{1}{2} \left( \log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)
\]

Restrict the format of variational parameters:
TT-GP (Our Method)

\[
\begin{align*}
\log p(y) & \geq \sum_{i=1}^{n} \left( \log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} (\tilde{K}_{ii} + \text{tr}(w_i^T \Sigma w_i)) \right) \\
& - \frac{1}{2} \left( \log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)
\end{align*}
\]

Restrict the format of variational parameters:

- \( \Sigma \) in Kronecker product format

\[
\Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \ldots \otimes \Sigma^D
\]
TT-GP (Our Method)

\[
\log p(y) \geq \sum_{i=1}^{n} \left( \log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} (\tilde{K}_{ii} + \text{tr}(w_i^T \Sigma w_i)) \right) \\
- \frac{1}{2} \left( \log \frac{|K_{mm}|}{|\Sigma|} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)
\]

Restrict the format of variational parameters:

- \( \Sigma \) in Kronecker product format
  \[
  \Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \ldots \otimes \Sigma^D
  \]
- \( \mu \) in TT format
  - \( \mu \) naturally reshapes to a tensor
Tensor Train format [Oseledets 2011]

Tensor $\mu$ is said to be represented in TT format if:

$$
\mu(i_1, \ldots, i_D) = G_1[i_1] \cdot G_2[i_2] \cdots G_D[i_D], \quad i_k \in \{1, \ldots, m_0\}
$$

$G_k$ — TT-cores, $r$ — TT-rank

- TT-format uses $\mathcal{O}(Dm_0r^2)$ memory to approximate a tensor with $m_0^D$ elements
- Allows efficient implementation of linear algebra operations
- Generalizes Kronecker product format ($r = 1$)
TT-GP method

- Set inducing points $\mathcal{Z}$ on a grid in the feature space.
- $\Sigma$ in Kronecker product format, $\mu$ in TT format
- Maximize the ELBO wrt to
  - TT-cores of $\mu$
  - Kronecker factors of $\Sigma$
  - kernel hyper-parameters
Properties of TT-GP

- Computational complexity

\[ O(nDm^{1/D}r^2 + Dm^{1/D}r^3 + Dm^{3/D}); \]

\[ m = m_0^D, \text{TT-ranks are on the scale of } r \approx 10; \]

- In the experiments we use up to \( n \approx 10^6, m \approx 10^{10} \)

- Computationally tractable for large \( D \)
  - For \( D >> 10 \) more practical to train embedding
Deep Kernel Embedding [Wilson et al. 2016]

Given base kernel \(k\), e.g. RBF

\[ k(x, x') = \alpha^2 \cdot \exp(-\|x - x'\|^2 / \beta^2), \]

define deep kernel as

\[ k_{\text{net}}(x, x') = k(\text{net}(x), \text{net}(x')), \]

where \(k\) is the base kernel, \(\text{net}\) is a mapping performed by a DNN.

- DNN weights → kernel hyperparameters
- Train as before
### Experiments: RBF kernel

| Dataset     | Name          | $n$  | $D$ | acc. | $m$ | $t$ (s) | acc. | $m$ | $d$ | $t$ (s) |
|-------------|---------------|------|-----|------|-----|--------|------|-----|-----|--------|
| SVI-GP / KLSP-GP | Powerplant    | 7654 | 4   | 0.94 | 200 | 10     | 0.95 | 354 | -   | 5      |
|             | Protein       | 36584| 9   | 0.50 | 200 | 45     | 0.56 | 309 | -   | 40     |
|             | YearPred      | 463K | 90  | 0.30 | 1000| 597    | 0.32 | 106 | 6   | 105    |
|             | Airline       | 6M   | 8   | 0.665*| -   | -      | 0.694 | 208 | -   | 5200   |
|             | svmguide1     | 3089 | 4   | 0.967| 200 | 4      | 0.969 | 204 | -   | 1      |
|             | EEG           | 11984| 14  | 0.915| 1000| 18     | 0.908 | 1210| 10  | 10     |
|             | covtype bin   | 465K | 54  | 0.817| 1000| 320    | 0.852 | 106 | 6   | 172    |

- SVI-GP – [Hensman et al. 2013]
- KLSP-GP – [Hensman et al. 2015]
Experiments: Deep Kernel Embedding

Learned representation for the Digits dataset, $n = 1797$, $D = 64$
## Experiments: Deep kernels

| Dataset  | SV-DKL | DNN | TT-GP |
|----------|--------|-----|-------|
|          | Name   | n   | acc.  | acc. | t (s) | acc. | d | t (s) |
| Airline  | 6M     | 0.781 | 0.780 | 1055 |       | 0.788±0.002 | 2 | 1375 |
| CIFAR-10 | 50K    | —   | 0.915 | 166  |       | 0.908±0.003 | 9 | 220  |
| MNIST    | 60K    | —   | 0.993 | 23   |       | 0.9936±0.0004 | 10 | 64   |

- SV-DKL — [Wilson et al. 2016]
Discussion

TT-GP

- Uses Tensor Train decomposition and Kronecker format for variational parameters
- Scales to large $n$, $m$, $D$
- Naturally allows training deep kernels
Discussion

TT-GP

- Uses Tensor Train decomposition and Kronecker format for variational parameters
- Scales to large $n$, $m$, $D$
- Naturally allows training deep kernels
- Tends to overestimate uncertainties