Revisiting Active Sets for Gaussian Process Decoders

Pablo Moreno-Muñoz*, Cilie W. Feldager* and Søren Hauberg

NEURAL INFORMATION PROCESSING SYSTEMS
New Orleans (USA), 2022

DTU

Highlights

▷ We revisit active sets for scaling GP decoders, a *sparse* approximation predominantly used before the seminal work of Snelson and Ghahramani (2006).

▷ New links between active sets and cross-validation based on the recent result from Fong and Holmes (2020).

▷ Formulation of the stochastic active sets (SAS) approach for both deterministic and Bayesian versions of GP decoders.

Historical Remarks

"Traditionally, sparse models have very often been built upon a carefully chosen subset of the training inputs. [...] In sparse Gaussian processes it has also been suggested to select the inducing inputs $\mathbf{X_u}$ from among the training inputs. Since this involves a prohibitive combinatorial optimization, greedy optimization approaches have been suggested [...]. Recently, Snelson and Ghahramani (2006) have proposed to relax the constraint that the inducing variables must be a subset of training/test cases, turning the discrete selection problem into one of continuous optimization."

Quiñonero-Candela and Rasmussen (2005) on the main difficulties behind the optimal selection of subsets.

Gaussian Process Decoders

The Gaussian process latent variable model (GP-LVM) (Lawrence, 2005) defines a decoder which is a non-linear mapping $\boldsymbol{x} = f(\boldsymbol{z})$ from the latent space $\mathcal{Z} \in \mathbb{R}^Q$ to observation space $\mathcal{X} \in \mathbb{R}^D$. The prior on this map is a Gaussian process (GP) so it is drawn like $f \sim \mathcal{GP}(0, k_\theta(\cdot,\cdot))$, where k_θ is the covariance function or *kernel*. $\boldsymbol{K}_{NN}$ denotes the evaluated kernel function so the i, jth element of $\boldsymbol{K}_{NN}$ equals $k_\theta(\boldsymbol{z}_i, \boldsymbol{z}_j)$.

$$p(\boldsymbol{x}|f,\boldsymbol{z}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n|f(\boldsymbol{z}_n), \sigma^2) \qquad p(f|\boldsymbol{z}) = \mathcal{N}(f(\boldsymbol{z})|0, \boldsymbol{K}_{NN})$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \int p(\boldsymbol{x}|f,\boldsymbol{z})p(f|\boldsymbol{z})\mathrm{d}f = \mathcal{N}(\boldsymbol{x}|0, \boldsymbol{K}_{NN} + \sigma^2\mathbb{I}).$$

$$\boxed{\mathcal{L} = -\frac{DN}{2}\log 2\pi - \frac{D}{2}\log|\boldsymbol{K}_{NN} + \sigma^2\mathbb{I}| - \frac{1}{2}\mathrm{tr}\big((\boldsymbol{K}_{NN} + \sigma^2\mathbb{I})^{-1}\boldsymbol{x}\boldsymbol{x}^\top\big)}$$

Whiteboard

Acknowledgments

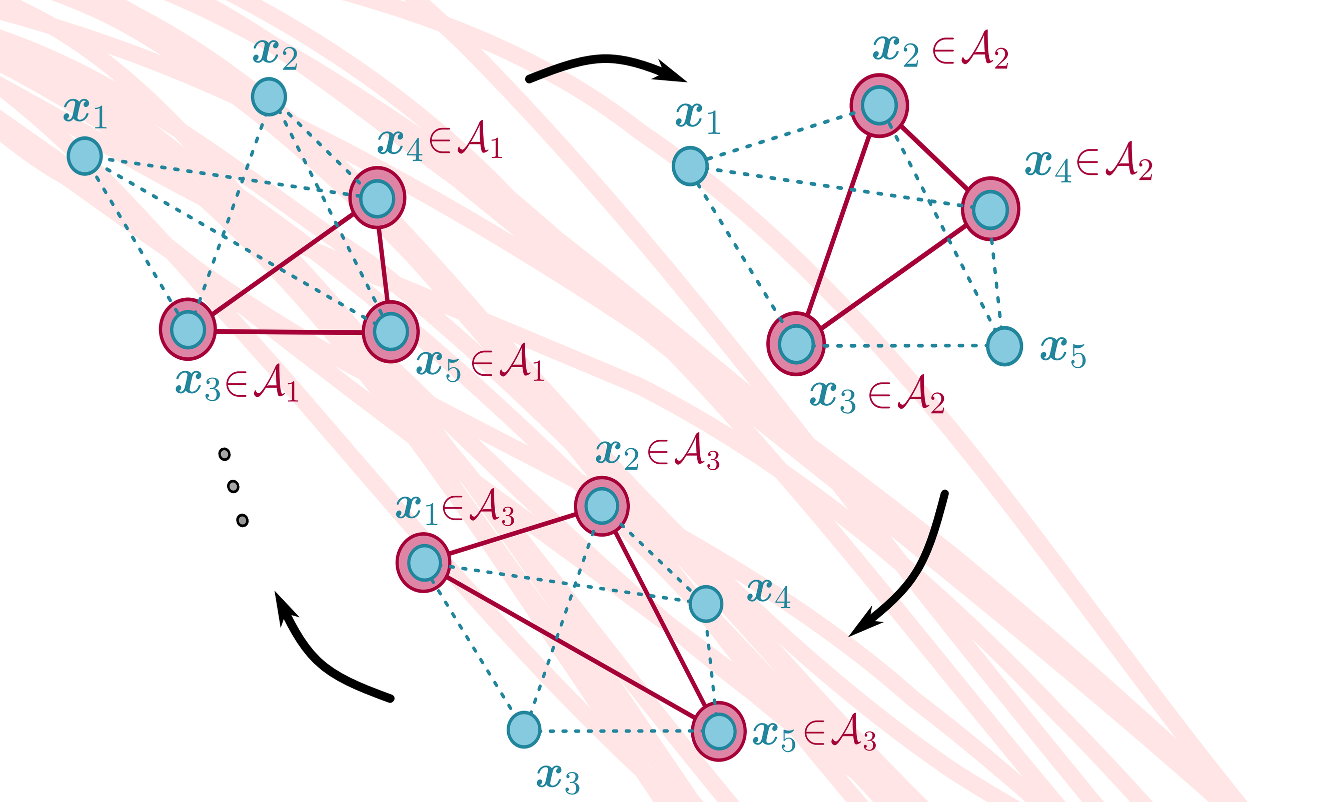
This work was supported by research grants (15334, 42062) from VILLUM FONDEN. This project has also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 757360). This work was funded in part by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606). This work was further supported by the Pioneer Centre for AI, DNRF grant number P1.

Stochastic Active Sets

Recently, Fong and Holmes (2020) linked *cross validation* (CV) with the log-marginal likelihood, effectively showing that the latter is equivalent to the average over exhaustive leave-R-out CV scores. In particular, the average is w.r.t. the size of the hold-out set.

$$\mathcal{S}_{\mathrm{CV}}(\boldsymbol{x}|R) = \frac{1}{\mathcal{C}}\sum_{p=1}^{\mathcal{C}}\frac{1}{R}\sum_{n\in\mathcal{R}_p}\log p(\boldsymbol{x}_n|\boldsymbol{x}_{\mathcal{A}_p},\boldsymbol{z}) = \frac{1}{R}\mathbb{E}_{\mathcal{A}}\left[\sum_{n\in\mathcal{R}_p}\log p(\boldsymbol{x}_n|\boldsymbol{x}_{\mathcal{A}},\boldsymbol{z})\right],$$

$$\log p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{r=1}^{N}\mathcal{S}_{\mathrm{CV}}(\boldsymbol{x}|r) = \mathcal{S}_{\mathrm{CCV}}(\boldsymbol{x}|R) + \mathcal{S}_{\mathrm{PCV}}(\boldsymbol{x}|R). \qquad \begin{aligned}\mathcal{S}_{\mathrm{CCV}}(\boldsymbol{x}|R) &= \textstyle\sum_{r=1}^{R}\mathcal{S}_{\mathrm{CV}}(\boldsymbol{x}|r)\\ \mathcal{S}_{\mathrm{PCV}}(\boldsymbol{x}|R) &= \mathbb{E}_{\mathcal{A}}[\log p(\boldsymbol{x}_{\mathcal{A}}|\boldsymbol{z}_{\mathcal{A}})]\end{aligned}$$



Stochastic Approximation

$$p(\boldsymbol{x}_{\mathcal{A}}|\boldsymbol{z}_{\mathcal{A}}) = \mathcal{N}(\boldsymbol{x}_{\mathcal{A}}|0, \boldsymbol{K}_{\mathcal{A}\mathcal{A}} + \sigma_n^2\mathbb{I}), \qquad p(\boldsymbol{x}_n|\boldsymbol{x}_{\mathcal{A}},\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{m}_{n|\mathcal{A}}, \boldsymbol{c}_{n|\mathcal{A}}),$$

$$\boxed{\log p(\boldsymbol{x}|\boldsymbol{z}) \approx \sum_{n\in\mathcal{R}}\log p(\boldsymbol{x}_n|\boldsymbol{x}_{\mathcal{A}},\boldsymbol{z}) + \log p(\boldsymbol{x}_{\mathcal{A}}|\boldsymbol{z}_{\mathcal{A}})}$$

Training algorithms

Algorithm 1 SAS for GP decoders

```
Input: Observed data x
Parameters: Initialize θ, ϕ            // θ, z if NA
for e in epochs do
    for b in batches do
        Sample x_batch ~ x
        x_R, x_A ← random_split(x_batch)
        if amortized then
            Get {z_R, z_A} ← g(x_R, x_A | ϕ)
        end if
        Compute K_AA^{-1}              // via Cholesky
        Evaluate log p(x_A | z_A)
        Evaluate log p(x_n | x_A, z), ∀x_n ∈ x_R
        Evaluate Eq. 6
        do Adam(θ, ϕ) step for L
    end for
end for
```

NA: Non-amortized.

Algorithm 2 SAS for Bayesian GP decoders

```
Input: Observed data x
Parameters: Initialize θ, ϕ   // θ, μ, σ if NA
for e in epochs do
    for b in batches do
        Sample x_batch ~ x
        x_R, x_A ← random_split(x_batch)
        if amortized then
            Get μ_z ← g_μ(x_R, x_A | ϕ)
            Get σ_z ← g_σ(x_R, x_A | ϕ)
        end if
        Sample {z_R, z_A} ~ q(μ_z, σ_z)   // RT
        Compute K_AA^{-1}                 // via Cholesky
        Evaluate L in Eq. 8
        do Adam(θ, ϕ) step for L_ELBO
    end for
end for
```

NA: Non-amortized, RT: Reparametrization trick.

Experiments & Results

Table 1: Comparative metrics for SAS and Bayesian SAS on MNIST, FMNIST and CIFAR-10.

MODEL	SAS			BAYESIAN SAS		
ACTIVE SET SIZE	$A = 100$	$A = 200$	$A = 400$	$A = 100$	$A = 200$	$A = 400$
MNIST / RMSE ↓	2.55 ± 0.98	2.47 ± 0.98	2.41 ± 0.93	2.16 ± 0.02	2.08 ± 0.02	1.99 ± 0.02
MNIST / MAE ↓	1.61 ± 0.97	1.55 ± 0.99	1.51 ± 0.96	1.11 ± 0.02	1.04 ± 0.02	0.96 ± 0.01
MNIST / NLPD ↓	2.99 ± 1.41	2.92 ± 1.38	2.84 ± 1.31	2.33 ± 0.03	2.26 ± 0.02	2.17 ± 0.02
FMNIST / RMSE ↓	2.37 ± 0.95	2.31 ± 0.94	2.25 ± 0.90	1.99 ± 0.17	1.88 ± 0.20	1.85 ± 0.13
FMNIST / MAE ↓	1.48 ± 0.91	1.42 ± 0.91	1.39 ± 0.89	1.11 ± 0.02	1.02 ± 0.03	0.98 ± 0.02
FMNIST / NLPD ↓	2.76 ± 1.33	2.71 ± 1.31	2.65 ± 1.23	2.16 ± 0.18	2.07 ± 0.19	2.04 ± 0.12
CIFAR10 / RMSE ↓	2.66 ± 1.08	2.55 ± 1.06	2.55 ± 1.03	2.74 ± 1.07	2.64 ± 1.08	2.57 ± 1.02
CIFAR10 / MAE ↓	1.77 ± 1.06	1.69 ± 1.06	1.69 ± 1.02	1.84 ± 1.03	1.76 ± 1.05	1.71 ± 1.03
CIFAR10 / NLPD ↓	3.20 ± 1.55	3.07 ± 1.44	3.32 ± 1.89	3.24 ± 1.53	3.14 ± 1.53	3.06 ± 1.45

All metrics are ($\times 10^{-1}$).

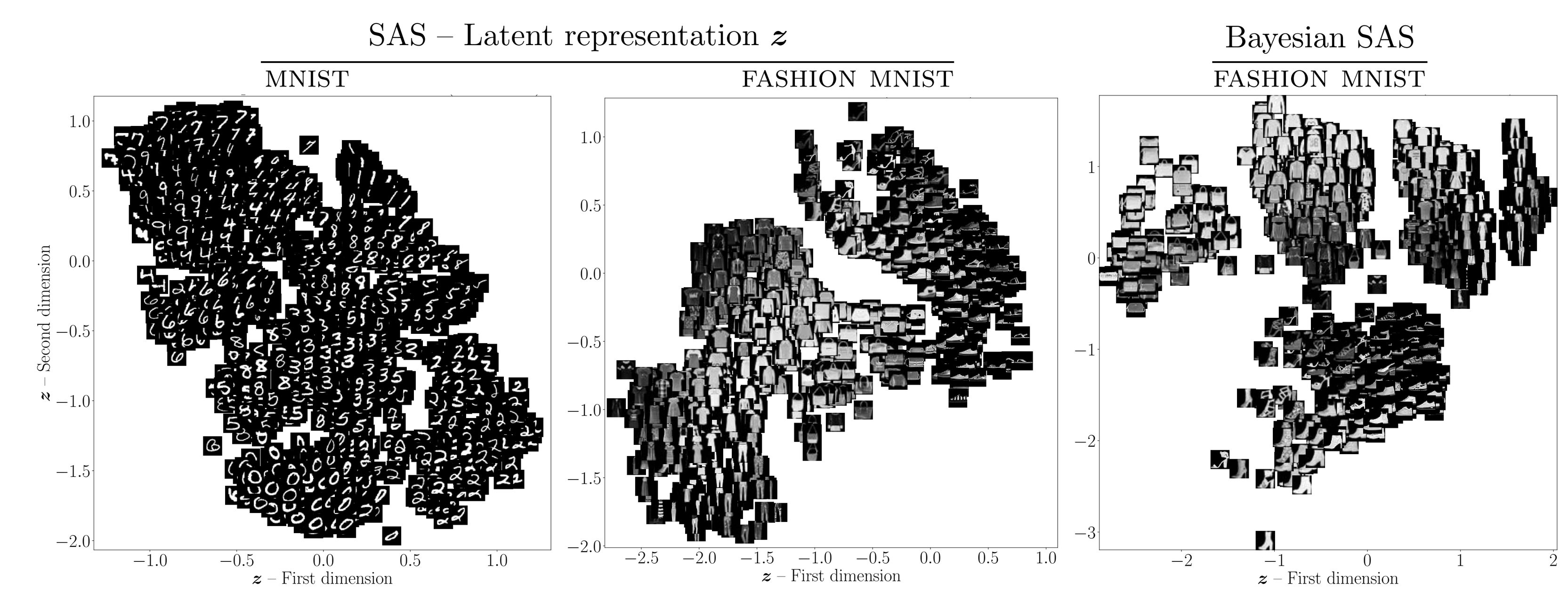
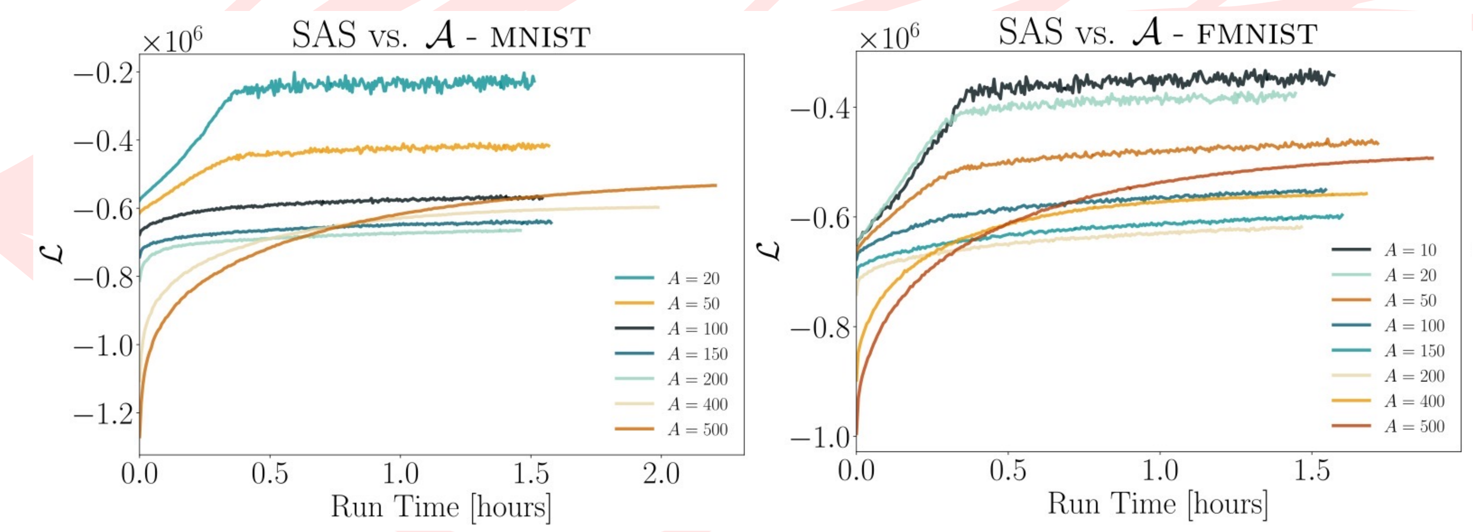


Table 2: Classification accuracy (↑) on 2-dim. latent space $\mathcal{Z}$.

MODEL	MNIST	FMNIST
BAYESIAN SAS-GP DEC. **(ours)**	0.63 ± 0.022	0.63 ± 0.020
BAYESIAN GP-LVM	0.18 ± 0.033	0.24 ± 0.043
VAE	0.54 ± 0.026	0.58 ± 0.008



References

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research (JMLR)*, 6(Dec):1939–1959, 2005.

E. Fong and C. C. Holmes. On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496, 2020.

N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research (JMLR), 6(11), 2005.*

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. *In Advances in Neural Information Processing Systems (NIPS)*, 2006.

Section for Cognitive Systems, DTU, Denmark
{pabmo,cife,sohau}@dtu.dk

https://github.com/pmorenoz/SASGP

PIONEER CENTRE FOR ARTIFICIAL INTELLIGENCE