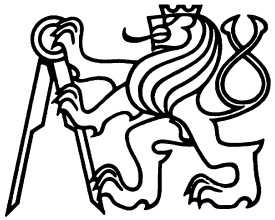
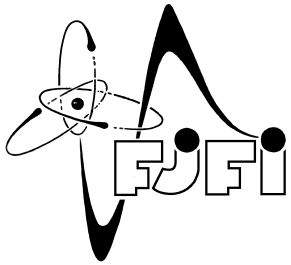


CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Nuclear Sciences and Physical Engineering



Hybrid Discriminative-Generative Training for Set Data

Jakub Bureš

Department of Mathematics

June 23, 2022

Content

Motivation

In machine learning, one can encounter two standard techniques:

- discriminative modeling,
- generative modeling.

Hybrid combination of these two approaches can improve a performance of a model. We focus on two hybrid variants:

1. HDGM [Abeel et. al 2020],
2. Semi-supervised VAE [Kingma et. al 2014],

which we apply to more complex cases, specifically, set data.

Discriminative modeling

- Data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where the input variable is $\boldsymbol{x} \in \mathbb{R}^D$ and the output variable $y \in \mathcal{C}$.
- *Classification problems* - the goal is to classify a new sample $\boldsymbol{x}$ into some category y from the finite set $\mathcal{C}$, so we are looking for distribution $p(y|\boldsymbol{x})$.
- We train the model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) : \mathbb{R}^D \to \mathcal{C}$, where $\boldsymbol{\theta}$ denotes the parameters of this model.
- In practice, *one–hot encoding* is often used, thus the model is in the form of $f_{\boldsymbol{\theta}}(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^C$.

Discriminative modeling

- Let the symbol $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]$ denotes y^{th} element of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$.
- For distribution modeling is typically used Softmax

$$q_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \frac{\exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)}{\sum_{y \in \mathcal{C}} \exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)}, \tag{1}$$

  which is the basis for the definition of *cross–entropy*.
- Optimization of the model is performed by minimizing total cross–entropy (corresponding to the MLE), therefore

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p_{\text{data}}(\boldsymbol{x},y)}\left[\log q_{\boldsymbol{\theta}}(y|\boldsymbol{x})\right]. \tag{2}$$

Generative modeling - Contrastive learning

- Generative models capture the joint probability $p(\boldsymbol{x}, y)$, or just $p(\boldsymbol{x})$ if there are no labels.
- Contrastive learning [Abeel et. al 2020] is a ML method typically utilized in the image classification.
- Here, we usually optimize

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})} \left[\log \frac{\exp\left(m_{\boldsymbol{\theta}}(\boldsymbol{x}) \cdot m_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)}{\sum_{i=1}^{M} \exp\left(m_{\boldsymbol{\theta}}(\boldsymbol{x}) \cdot m_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right)} \right] \tag{3}$$

- We define a function $m_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^H$ that maps each sample to the representation space of dimension H.
- The sample x' is called augmented view.

Generative modeling - Contrastive learning

Modification to our case:

- there is no need for augmented views $\boldsymbol{x}'$
- instead of mapping m we simply use $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]$ with labels y, yielding generative term

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}|y) \approx \frac{\exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)}{\sum_{j=1}^{M} \exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_j)[y]\right)}, \tag{4}$$

where $M < N$ is a number of normalization samples.

Hybrid Discriminative–Generative Modeling

- At this point, we have defined discriminative component $q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}\right)$ and generative component $q_{\boldsymbol{\theta}}\left(\boldsymbol{x}|y\right)$.
- We can finally minimize the hybrid, convex combination of these two components

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p_{\text{data}}(\boldsymbol{x},y)}\left[\alpha \log q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}\right) + (1-\alpha) \log q_{\boldsymbol{\theta}}\left(\boldsymbol{x}|y\right)\right]. \tag{5}$$

- Parameter $\alpha \in [0,1]$ weighs generative and discriminative counterparts.

Variational Autoencoder

- A generative modeling method, the goal is to find $p(\boldsymbol{x})$ using the latent variable $\boldsymbol{z}$, specifically, encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$.
- It leads to the ELBO optimization

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \,\|p_{\boldsymbol{\theta}}(\boldsymbol{z})\right) \tag{6}$$

- No labels!
- However, there is no combination of discriminative and generative models -> **semi–supervised VAE** [Kingma et al. 2014].

VAE, semi–supervised version

It is necessary to consider two cases.

- First, consider observation $\boldsymbol{x}$ that has its class label y:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x},y)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z},y)+\log p_{\boldsymbol{\theta}}(y)\right]-D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x},y\right)\|p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) \quad (7)$$

- Secondly, observation $\boldsymbol{x}$ is lacking it class label y:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(y,\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z},y)+\log p_{\boldsymbol{\theta}}(y)\right]-D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(y,\boldsymbol{z}|\boldsymbol{x}\right)\|p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) \quad (8)$$

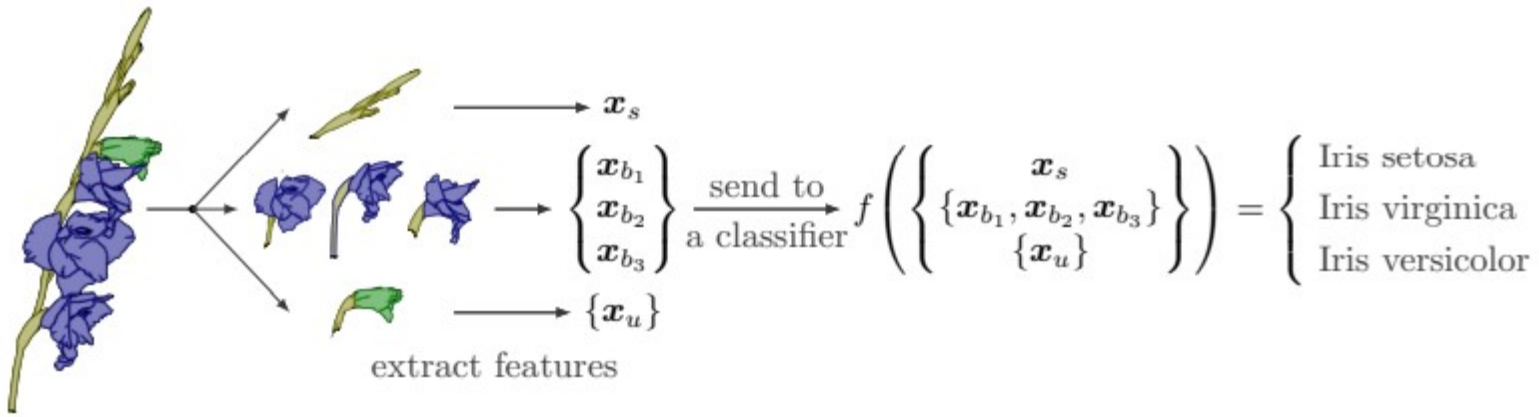
- We need a modification for a fully supervised dataset. We obtain

$$\tilde{J}^{\nu}_{HM}\left(\boldsymbol{\theta},\boldsymbol{\phi}\right)=\tilde{J}_{HM}\left(\boldsymbol{\theta},\boldsymbol{\phi}\right)+\nu\cdot\mathbb{E}_{\tilde{p}_l(\boldsymbol{x},y)}\left[-\log q_{\boldsymbol{\phi}}\left(y|\boldsymbol{x}\right)\right], \quad (9)$$

  where parameter ν is a weight.

Multiple Instance Learning, MIL

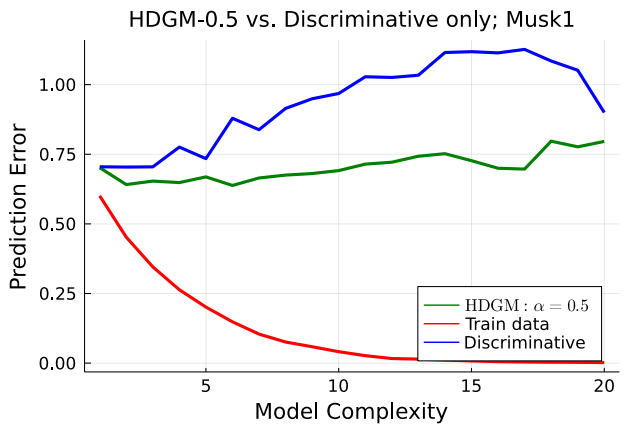
- In MIL, one sample is a set of vectors, these vectors are called instances and the sets are called bags.
- If the space of all instances is $\mathcal{X}$ and the space of all bags is $\mathcal{B}(\mathcal{X})$, then in MIL the model is defined as $f_{\boldsymbol{\theta}} : \mathcal{B}(\mathcal{X}) \to \mathcal{C}$.
- However, to properly define such a model, we need an embedded space and aggregation functions.
- We use the HMill framework and the Mill.jl package.



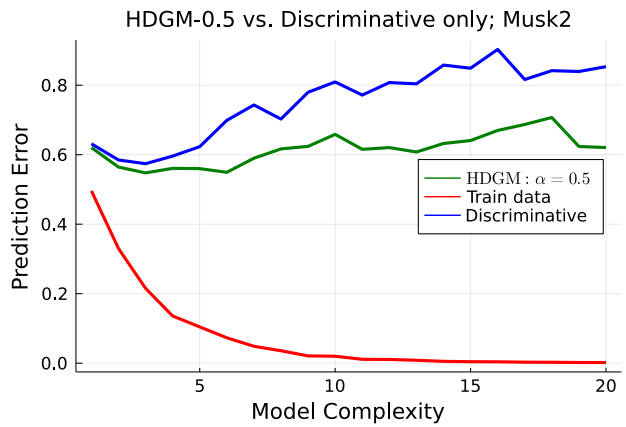
Obrázek: A general representation of set data. Credit [2].

Results - HDGM

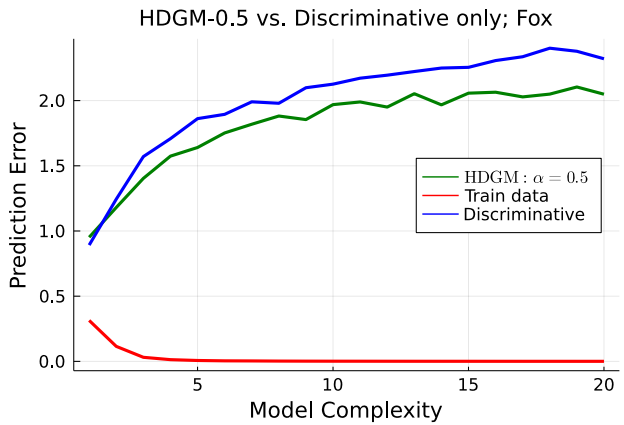
Cross–Validation of HDGM:



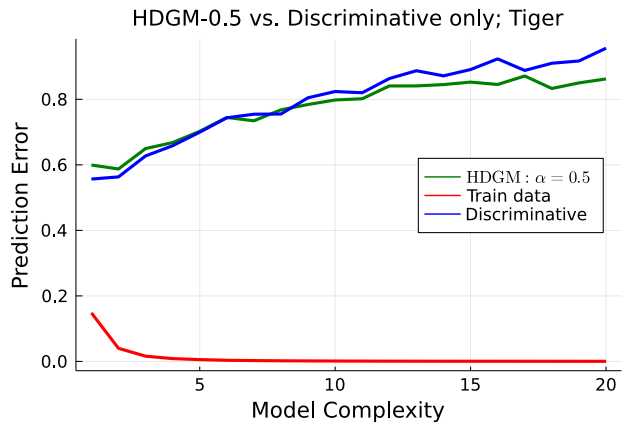
(a) Musk1.



(b) Musk2.



(c) Fox.



(d) Tiger.

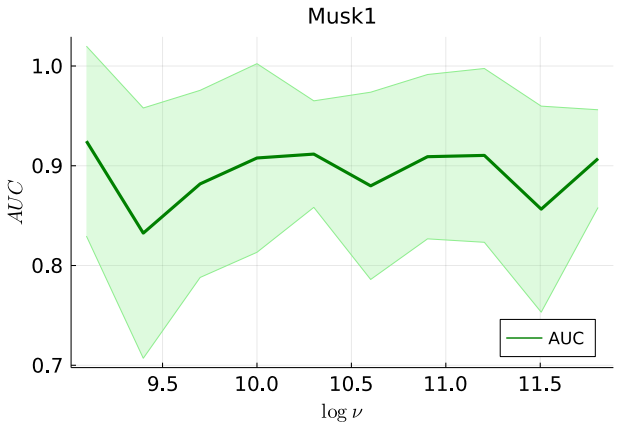
Results - HDGM

Average AUCs and st. deviations of individual data sets for both approaches:

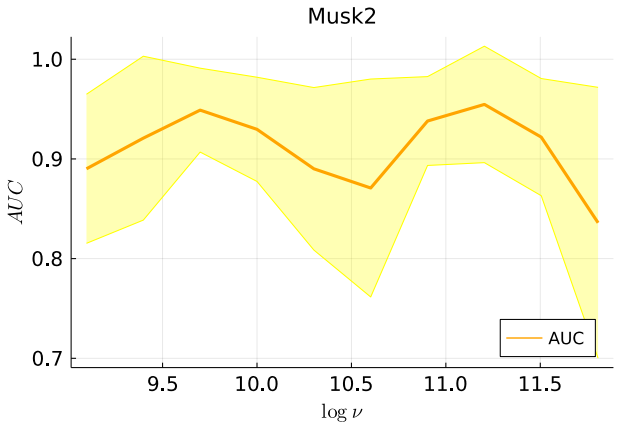
	D	**HDGM** $\alpha = 0.5$
Musk1	81.40$\pm$12.11 %	**83.89** $\pm$11.13%
Musk2	79.77$\pm$8.96%	**82.78** $\pm$7.38%
Tiger	**91.22** $\pm$2.34%	90.25$\pm$2.80%
Fox	54.43 $\pm$3.77%	**56.22**$\pm$6.18%

Results - Hybrid VAE

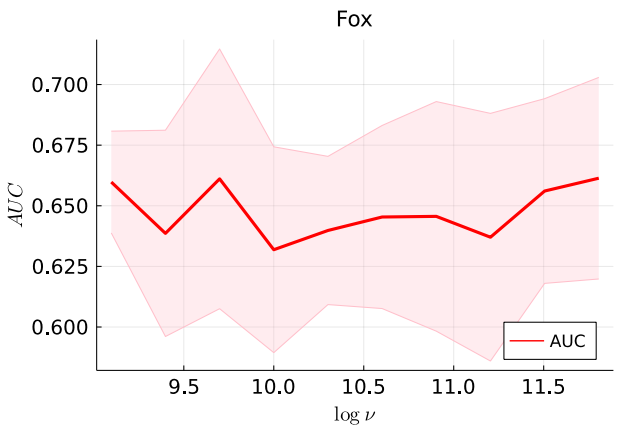
- Dependence of the average AUC on the logarithm of the parameter ν



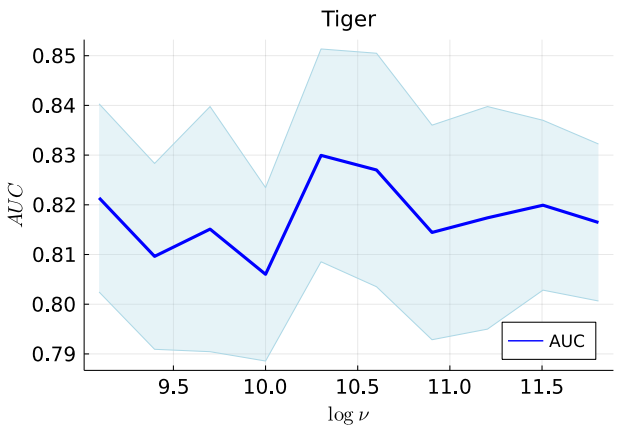
(e) Musk1.



(f) Musk2.



(g) Fox.



(h) Tiger.

Conclusion

- Slight reduction of prediction error for test data.
- Overall increase of AUC.
- High variance.

Thank you for your attention.


T. Pevny, S. Mandlik, *Mill.jl framework: a flexible library for (hierarchical) multi-instance learning* [on–line] Available from: `https://github.com/CTUAvastLab/Mill.jl`. Accessed 30 April 2020.


S.Mandlik.: *Mapping the Internet: Modelling Entity Interactions in Complex Heterogeneous Networks*. arXiv preprint arXiv:2104.09650.


D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes.* arXiv preprint arXiv:1312.6114, 2013.