Few-shot learning with KRR

Prudencio Tossou

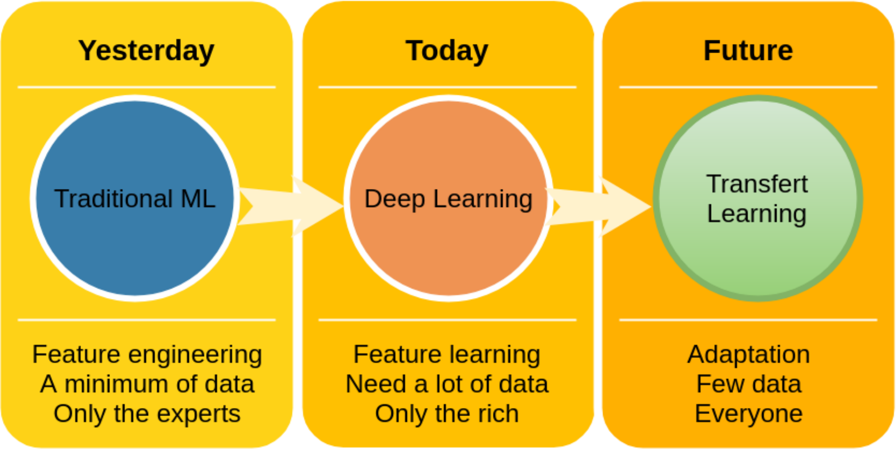
Groupe de Recherche en Apprentissage Automatique
Départment d'informatique et de génie logiciel
Université Laval

April 6, 2018

Plan

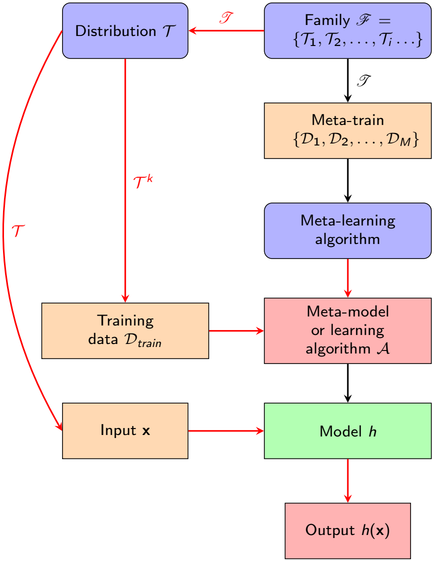
1. Motivation
2. Problem statement
3. Our approach: MetaKRR
4. Experiments
5. Future works and conclusion

The future of ML



What few learning is trying to do?

- By leveraging past learning experiences
- Through META-LEARNING
- The past gives a strong prior knowledge
- If one use it, things can be done more efficiently in the present



Few-shot regression

- Recent works focus on classification and reinforcement learning
- Not much experiments with regression datasets
- Versus classification: harder to generalize from few examples
- Applications:
  $\hookrightarrow$ drug discovery –> Drugs at lower costs
  $\hookrightarrow$ recommender systems –> Deal with products with few ratings

Plan

The objective

The optimal meta-model

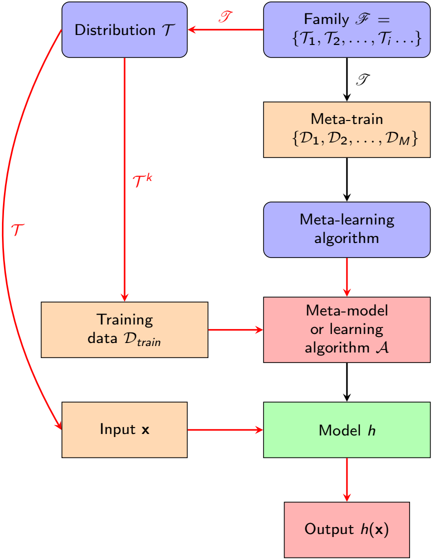
$$\mathcal{A}^* = \operatorname*{argmin}_{\mathcal{A}} \mathop{\mathbf{E}}_{\mathcal{T}\sim\mathscr{T}} \mathop{\mathbf{E}}_{\mathcal{D}_{train}\sim\mathcal{T}^k} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \mathcal{L}(\mathcal{A}(\mathcal{D}_{train})(\mathbf{x}), y)$$

Few-shot regression

- Quadratic loss

$$\mathcal{L}(y, y') = (y - y')^2$$

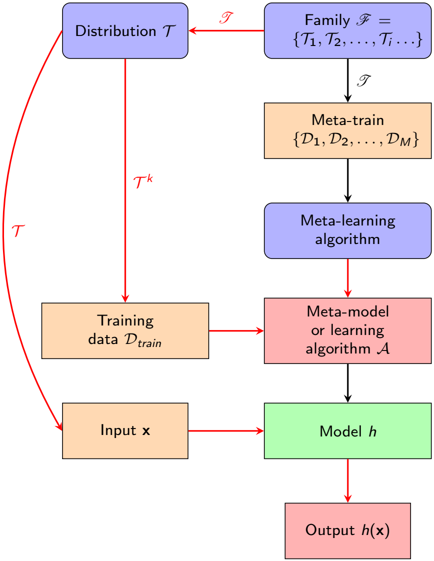
- $|\mathcal{D}_{train}| \leq 20$



In practice (1)

The meta-datasets

- Sample a distribution $\mathcal{T}_i$ from $\mathscr{F}$
- For each $\mathcal{T}_i$ sample a $\mathcal{D}_i$
- Split the resulting collection of datasets in 3 partitions:
  $\hookrightarrow$ $\mathscr{D}_{meta-train}$ for training
  $\hookrightarrow$ $\mathscr{D}_{meta-valid}$ for hyper-parameter selection
  $\hookrightarrow$ $\mathscr{D}_{meta-test}$ for unbiased evaluation of the meta-model



In practice (2)

Episodic training

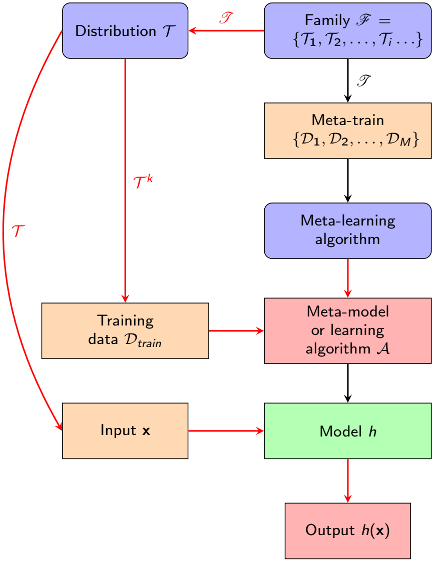
Initialize Θ

Loop

- Sample an $\mathcal{D}_i$ from $\mathscr{D}_{meta-train}$
- Sample $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ of k examples each from $\mathcal{D}_i$
- Compute $h := \mathcal{A}(\mathcal{D}_{train}, \Theta)$
- Estimate the loss of h on $\mathcal{D}_{test}$
- Update Θ

An episode

A pair of $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ from a $\mathcal{D}_i$



Plan

The meta-model

Tandem combination

- Feature extractor $\phi : \mathcal{X} \rightarrow \mathcal{K}$ shared by all tasks
- Regression Algorithm –> $h(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}), \quad \mathbf{w} \in \mathcal{K}$

Feature extractor

Could be anything

- CNN for images
- LSTM for sequences
- FC for vectors, etc

Parameters to be found during the episodic training

The model (1)

The regression algorithm should aim for generalization (SRM) [4]

Given $\mathcal{D}_{train}$,

$$\mathbf{w}^*_{\mathcal{D}_{train}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{(\mathbf{x},y)\in\mathcal{D}_{train}} (\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}) - y)^2 + \lambda \|\mathbf{w}\|_2^2,$$

The optimal solution is given by KRR[3]

$$\mathbf{w}^* = \sum_{i=1}^{k} \alpha_i \boldsymbol{\phi}(\mathbf{x}_i), \quad \text{with} \quad \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)^T = (K + \lambda I)^{-1}\mathbf{y},$$

$$K_{ij} = \boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j), \quad \text{where} \quad i = 1\ldots k, j = 1\ldots k$$

The model (2)

Advantages of KRR

- Closed form
- Few-shot –> solving the dual system is highly advantageous over the primal

Drawbacks

- Fine tuning regularizer and kernel hyper-parameters
- Cross-validation and validation set: costly and need more data

Selection of regression hyper-parameters

Option 1: Episode dependant

$\hookrightarrow$ FC network g to predict the right values
$\hookrightarrow$ inputs = sufficient statistics of the training examples of $\mathcal{D}_{train}$
$\hookrightarrow$ statistics = mean, std, max, min of $\{y_1, y_2, \ldots, y_k\}$ and $\{\phi(\mathrm{x}_1), \phi(\mathrm{x}_2), \ldots, \phi(\mathrm{x}_k)\}$
$\hookrightarrow$ For example, for a given $\mathcal{D}_{train}$, the KRR regularizer is given by:

$$\exp\left(HardTanh_{a,b}(g(\mathcal{D}_{train}))\right), \text{ with } HardTanh_{a,b}(x) = \begin{cases} a & \text{if } x < a \\ x & \text{if } a \leq x \leq b \\ b & \text{if } x > b \end{cases}$$

Option 2: Same for all episodes

- Associate a parameter to each Hp
- Find the right value during back propagation

MetaKRR: the training with option 1

Pseudo-code

Initialize Θ of ϕ and Λ of g

Loop

- Sample an $\mathcal{D}_i$ from $\mathscr{D}_{meta-train}$
- Sample a $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ from $\mathcal{D}_i$
- Transform all inputs with ϕ
- Compute λ_{train} with g
- Solve KRR to find $\mathbf{w}^*$, thus h^*
- Compute the quadratic loss of h on $\mathcal{D}_{test}$
- Back-propagate the loss and update Θ and Λ

Other details

- Train for 20K episodes
- Use $\mathscr{D}_{meta-valid}$ to select the best model

Plan

Datasets

MHC class II peptides

- Task: predict the binding energy of a peptide to a protein (MHC II complex).
- Collection of 14 datasets, one per protein
- Each dataset has from 500 to 5K examples
- Input = peptide (string)
- Output=energy to a MHC protein
- 14 few-shot regression tasks
- CNN feature extractor 256×3

Binding molecules

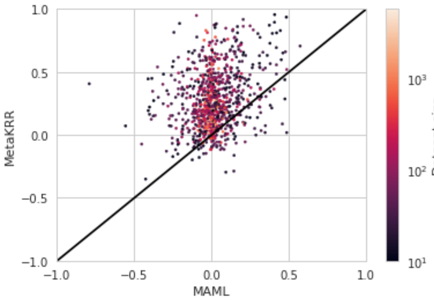
- Task: predict the binding affinity of small molecules to a protein
- Collection of 3741 regression task each related to a protein and an organism
- Input = molecule SMILES (string)
- Output = binding affinity
- Meta-train, meta-valid and meta-test contain 2104, 702 and 935 tasks
- CNN feature extractor 512×4

Results on MHC class II peptides

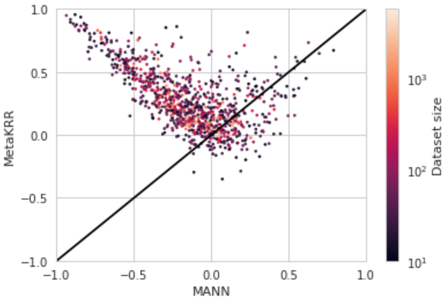
Test complex	MetaKRR-g	MetaKRR-u	MAML[1]	MANN[2]	pretrain
DRB1*0101	0.435	0.475	0.469	0.530	0.176
DRB1*0301	0.512	0.501	0.405	0.522	-0.177
DRB1*0401	0.547	0.555	0.457	0.484	0.165
DRB1*0404	0.573	0.608	0.470	0.617	0.105
DRB1*0405	0.643	0.652	0.531	0.676	0.156
DRB1*0701	0.694	0.694	0.613	0.673	0.199
DRB1*0802	0.404	0.388	0.407	0.426	0.125
DRB1*0901	0.509	0.535	0.389	0.565	0.159
DRB1*1101	0.641	0.626	0.567	0.537	-0.169
DRB1*1302	0.471	0.477	0.401	0.465	0.116
DRB1*1501	0.640	0.629	0.623	0.644	0.180
DRB3*0101	0.318	0.356	0.294	0.313	0.071
DRB4*0101	0.574	0.602	0.548	0.596	0.203
DRB5*0101	0.660	0.624	0.559	0.669	0.221
Average	0.544	0.552	0.481	0.551	0.109

Results on BindingDB

MetaKRR versus MAML

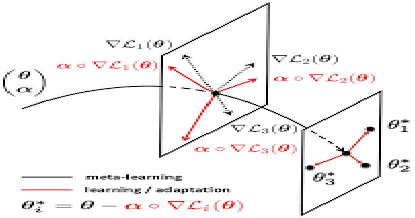


MetaKRR versus MANN



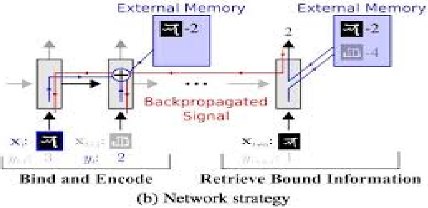
Discussion

MAML



- Hard to find the best initialization point.

MANN



- Easy to classify with but harder for regression

Plan

Future works

- Select the right values with the network g within a grid of hyper-parameters
- Include in our experiments a recommender system dataset : Netflix challenge

Conclusion

- We have introduced MetaKRR, a few-shot regression algorithm
- State of the art performances
- Three key ideas:
    - Leverage past experiences to find the most appropriate mapping function
    - Use the structural risk minimization to enforce generalization
    - Leverage past experiences to choose adequately the trade-off inside the SRM
- Not new ideas but they graciously combine together to give the MetaKRR

References

[1] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[2] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[3] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. 1998.

[4] V. Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

Thanks for your attention