Machine Learning in Communications
Lecture 5a: Theory-Guided Machine Learning

Harpreet S. Dhillon

Wireless@VT, Bradley Department of Electrical & Computer Engineering
Virginia Tech, Blacksburg, VA

https://www.dhillon.ece.vt.edu
hdhillon@vt.edu

JTG/IEEE Information Theory Society Summer School
IIT Kanpur

Acknolwedgements and References

- Acknowledgements: Keerthana Bhogi, Virginia Tech.

- Many of the foundational ideas presented in this lecture are from the following references.

[1] A. Karpatne, *et al.*, "Theory-guided data science: A new paradigm for scientific discovery from data," IEEE Trans. on Knowledge and Data Engineering, vol. 29, no. 10, pp. 2318-2331, 2017.

[2] L. Von Rueden, *et al.*, "Informed machine learning-a taxonomy and survey of integrating knowledge into learning systems," IEEE Trans. on Knowledge and Data Engineering, to appear. arXiv:1903.12394, 2019.

Lots and Lots of Training Data

- The "mainstream" machine learning applications such as, natural language process and computer vision, have seen a tremendous success.
- Along the way, it has transformed commercial industries, such as retail and advertising.
- Enabler: Large volumes of training data (think of all the images on the Internet).
- It is now easy to imagine why it has not had a similar impact on other scientific disciplines.
  - Unlike object detection or image processing, the cost of data acquisition for some applications can be prohibitive.
  - Example: Equipment needed for propagation measurements in new bands is expensive!
  - The notion of ML being a "black-box solution" does not help.

Shortcomings of Machine Learning

Insufficient Amount of Data in Most Disciplines

- Insufficient training data leads to overfitting.
- As we discussed already, overfitting learns spurious relationships in the training data because of which the learning models will not generalize well on the unseen data.

Inconsistency of Models

- The trained models may not be interpretable.
- Even worst, they may not comply with the basic physical laws governing the system being studied.
- Example: Remember our statistical model: $Y = f(X) + \epsilon$. What if we know that function $f(\cdot)$ is monotonic with respect to some feature but our trained model does not comply with that?
  - Solution: Do not ignore the domain knowledge. Even more important in the presence of limited data.

Theory-guided Machine Learning (TGML)

- Integrate scientific knowledge to further enhance the unique capabilities of ML approaches.
- This general idea is known by multiple names:
  - Physics-guided neural networks
  - Theory-guided deep learning
  - Theory-guided data science
- This paradigm could potentially overcome the shortcomings of both approaches: theory-based (theory-based models are consistent but simple) and data-based (trained models are complex but may not remain consistent).
- Objective: Accuracy (model performance), Regularization (model simplicity), Physical Consistency.

TGML: Theory Meets Data

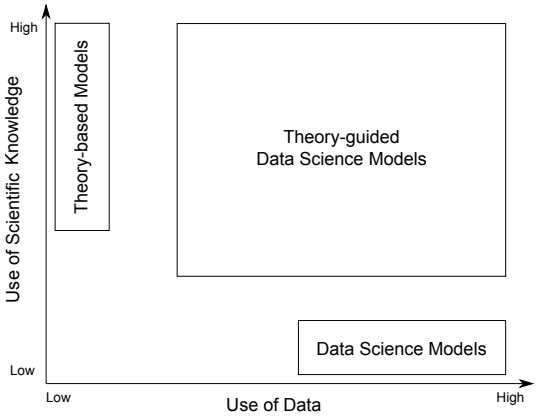

Figure: TGML makes better use of limited data by being cognizant of the underlying structures of the problems (scientific knowledge). Picture from [1].

TGML Workflow

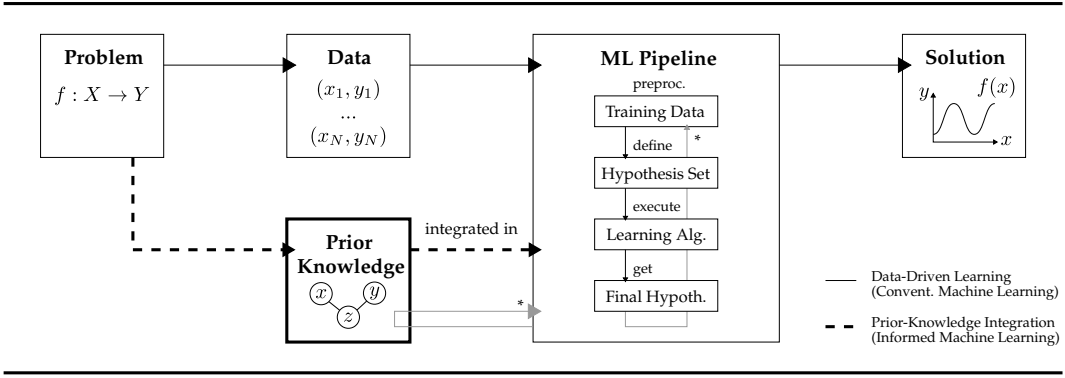

Figure: Information flow in a TGML model. Picture from [2].

Knowledge Representation and Integration

- Prior knowledge could come in various forms (often highly context specific). Common representations include:
  - Algebraic equations
  - Logic equations
  - Differential equations
  - Probabilistic relations
  - Invariances (today's case study on Grassmann clustering)
- Prior knowledge can also be integrated in the learning process in numerous ways:
  - Training data
  - Hypothesis class (today's case study on Grassmann clustering)
  - Learning algorithm (today's case study on Grassmann clustering)

Knowledge Integration: Training Data

- Theory knowledge can be used for augmenting training data obtained from measurements.
- Can you think of such an example from communications? Let's discuss in the live session.

Knowledge Integration: Hypothesis Set

- Recall that a smaller hypothesis set is desirable to avoid overfitting.
- Prior knowledge can help with that through the choice of appropriate loss function or model architecture.
- Model architecture: Chosen based on the problem setup. Example: N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in IEEE SPAWC, Jul. 2017.
- Loss function: Theory-guided regularization. For instance, put a large penalty in the loss function if some physical constraint/law is violated.
- In today's case study (Lecture 6), we will see how the invariance property of our solution helped in restricting our hypothesis set, which further allowed us to select a simple learning algorithm.

Knowledge Integration: Learning Algorithms

- Choice of learning algorithms: Do I need NN or is k-means enough? Today's case study.
- Initialization: Prior knowledge can help in avoiding bad initializations. Example: models could be pre-trained on physically-consistent (could be simulated) data and then fine-tined using real measurements.
- Theory-guided priors: Physically consistent priors could also be used in the Bayesian setup to develop better learning models.
- Constrained optimization: Constraints restrict the space of feasible model parameters.

Final Note: Creating Hybrid Models

- One can also think of hybrid combinations of theory-based and data-based models, where some parts of the problem are handled by theory-based models while the remaining are modeled using data-based models.
- Our *Determinantal Learning* case study from Day 1 falls in this category. The idea of using a general probabilistic model came from theory and its parameters were learnt from data.
  - Of course, this general approach of selecting a model and then learning its parameters is extremely popular but when the model comes from theory, it also comes under the purview of TGML.

Summary

- Vast prior knowledge of a given area must not be ignored.
- TGML is context specific: Need to come up with innovative techniques to integrate prior knowledge for a given problem.
- One can either enforce physical consistency or problem constraints strictly (such as through designing model architecture or specifying theory-based constraints) or softly (such as through priors or regularization terms).