UC San Diego

MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

Carnegie Mellon University School of Computer Science

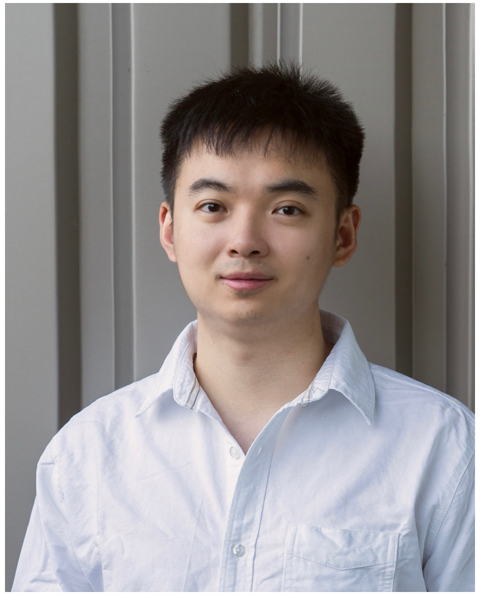
Petuum

A "Standard Model" for Machine Learning

Zhiting Hu, Hao Zhang, Eric Xing

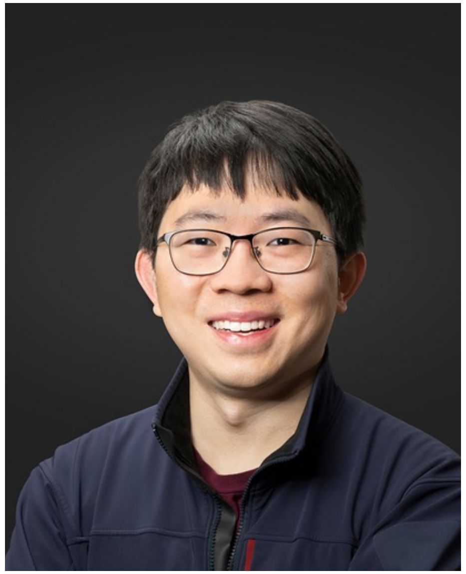
DeepLearn 2023 Winter

Presenters



Zhiting Hu

Assistant Professor
@UCSD



Hao Zhang

Assistant Professor
@UCSD

Eric P. Xing

Professor @ CMU
President @ MBZUAI
Co-founder @ Petuum

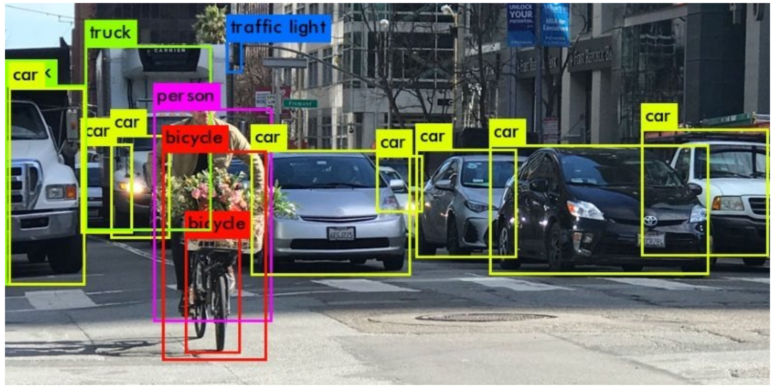
UC San Diego

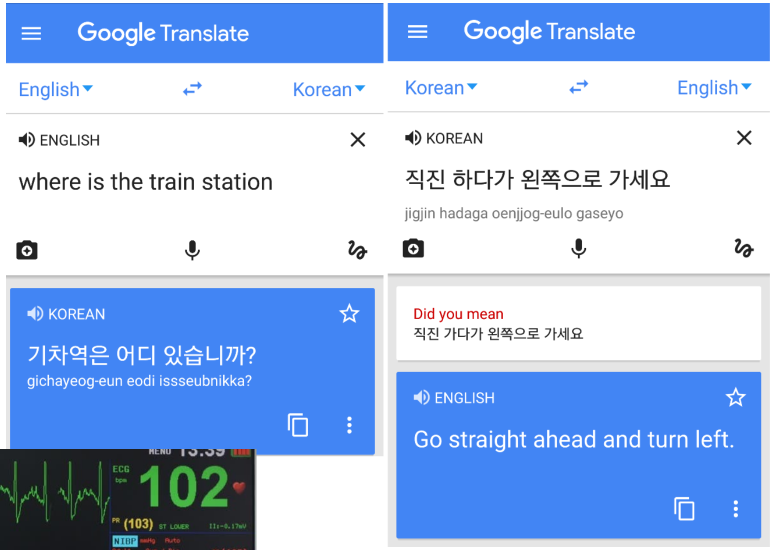
MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

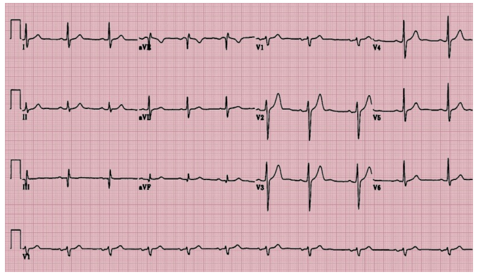
Carnegie Mellon University
School of Computer Science

Petuum

Real-world Machine Learning Problems





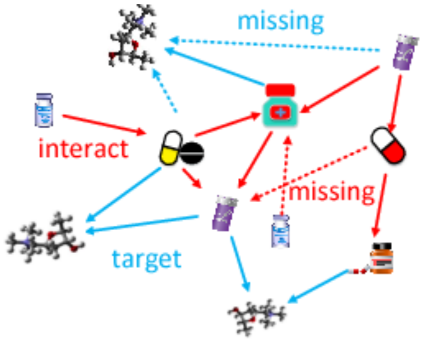


Data and experience of all kinds

Data examples

Type-2 diabetes is 90% more common than type-1

Rules/Constraints

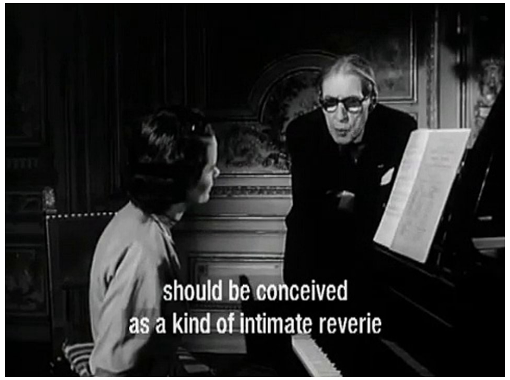


Knowledge graphs

Rewards

Auxiliary agents

Adversaries



Master classes

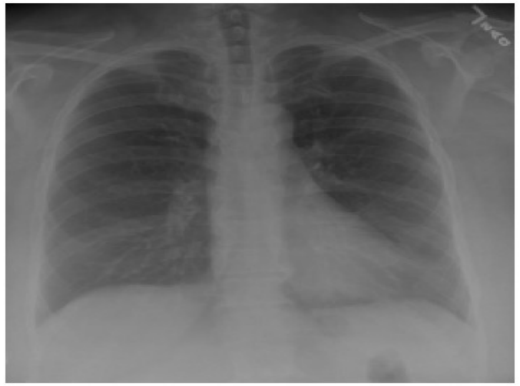
…

- *And all combinations of such*
- *Interpolations between such*
- *…*

An Example: ML for Healthcare

A ready-to-use real AI solution is extremely complex, given all these experiences to train on

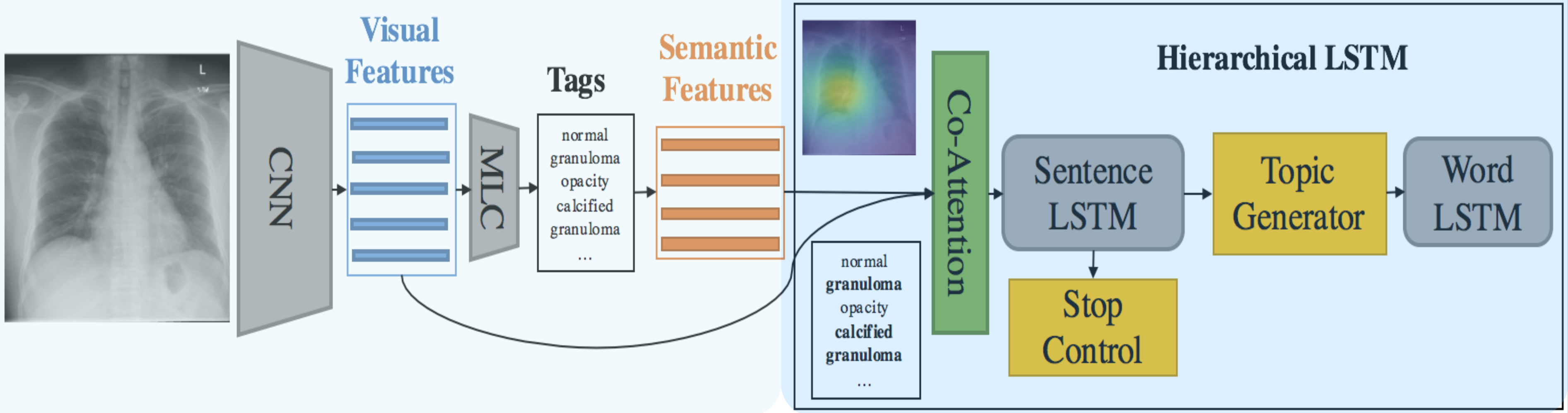
Use Case: Automatic Medical (or other) Report Generation

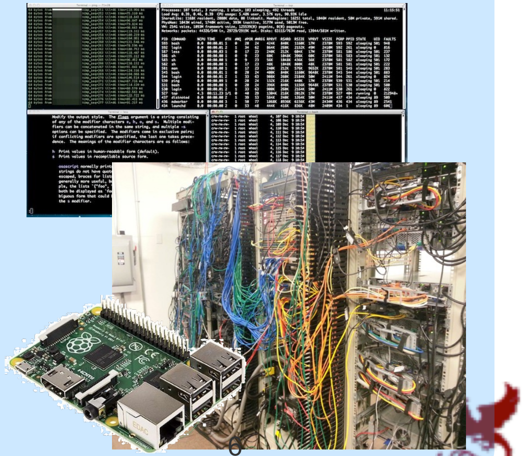


Findings:
There are no focal areas of consolidation.
No suspicious pulmonary opacities.
Heart size within normal limits.
No pleural effusions.
There is no evidence of pneumothorax.
Degenerative changes of the thoracic spine.
Impression:
No acute cardiopulmonary abnormality.

- Abnormal regions in medical images are difficult to identify.
- How to localize the image regions and tags that are relevant to a sentence?
- How to distribute topics across sentences
- How to make report readable to humans?

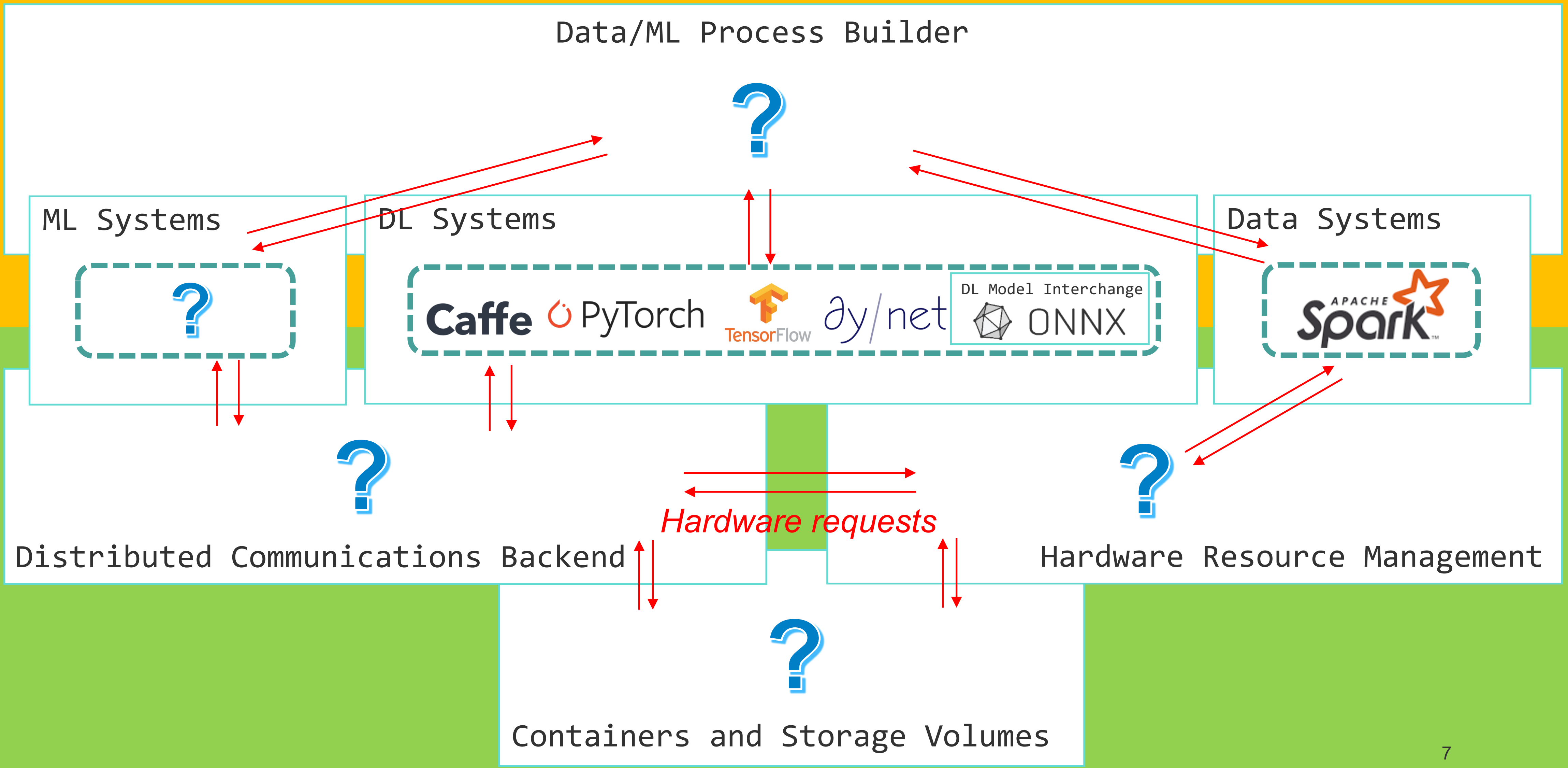
Raw Data Enrichment | Model/Algorithm | System/Infra





Pe

Inter-operability between diverse systems?



An AI solution

Data wrangling
Feature engineering
Model compiling
Algorithm designing
Distributed training
Debugging
Resource provisioning
Hardware management
Fault recovery
…etc

Build versus Craft

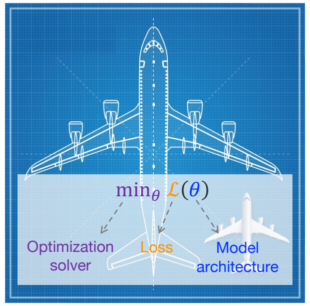
- Modules, Building-blocks
- Nuts and Bolts
- Interoperability
- Process
- Soundness

Schedule

- **Lecture#1:** Theory: The Standard Model of ML

  A blueprint of ML paradigms for ALL experience

  (Jan 19 Thursday, 4:45pm-6:15pm UK Time)



- **Lecture#2:** Tooling: Operationalizing The Standard Model

  Compose your ML solutions like playing Lego

  (Jan 20 Thursday, 1:00pm-2:30pm)

- **Lecture#3:** Computing: Modern infrastructure for productive ML

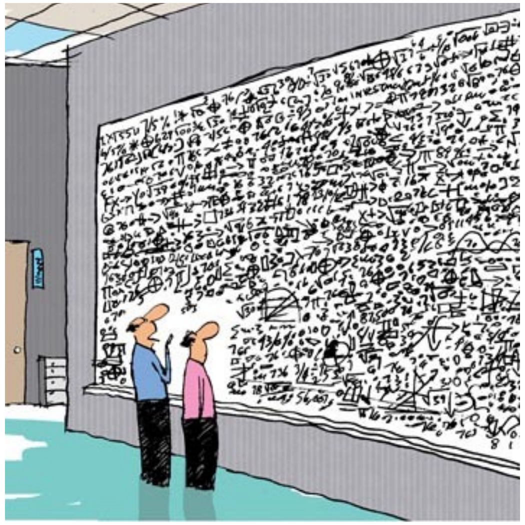
  Automatic tuning, distributing, and scheduling

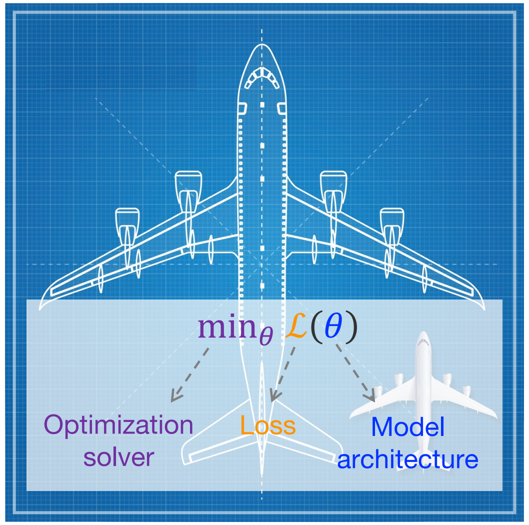
  (Jan 20 Thursday, 4:45pm-6:15pm)

UC San Diego · Mohamed bin Zayed University of Artificial Intelligence · Carnegie Mellon University School of Computer Science · Petuum

Theory:

The Standard Model – A Blueprint for ML



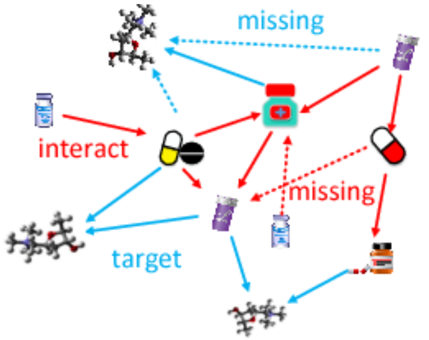


Experience of all kinds

Data examples

Type-2 diabetes is 90% more common than type-1

Rules/Constraints

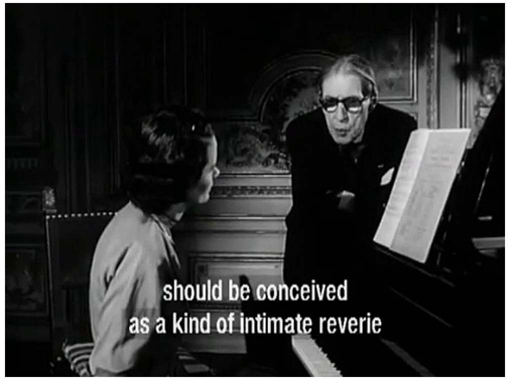


Knowledge graphs

Rewards

Auxiliary agents

Adversaries

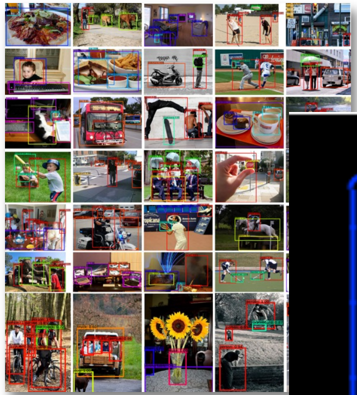


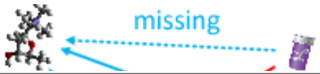
Master classes

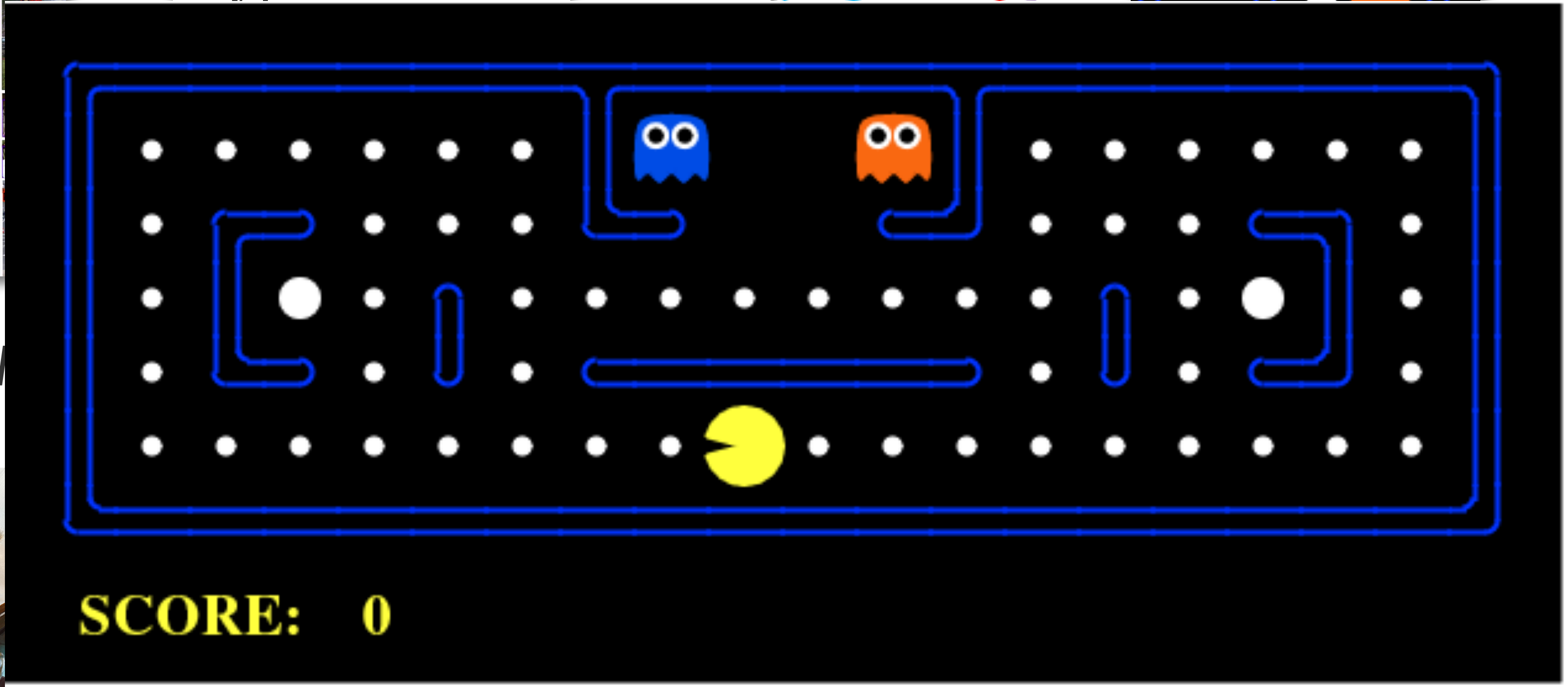
…

- *And all combinations of such*
- *Interpolations between such*
- *…*

Experience of all kinds



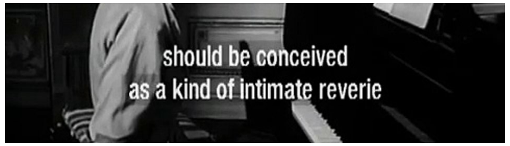



Data exampl

Auxiliary agents

ations thereof

Adversaries

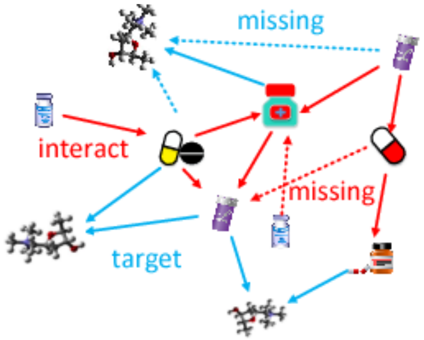
Master classes

Experience of all kinds

Data examples

Type-2 diabetes is 90% more common than type-1

Rules/Constraints

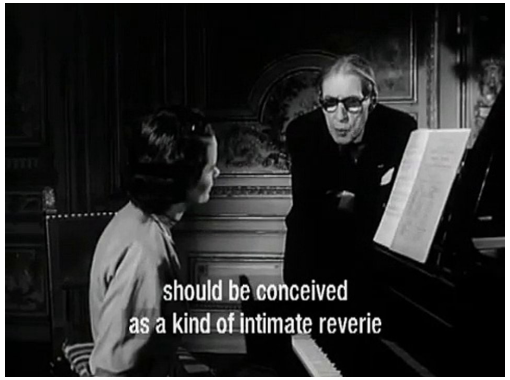


Knowledge graphs

Rewards

Auxiliary agents

Adversaries



Master classes

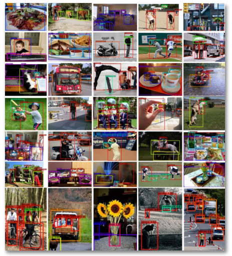
...

- *And all combinations of such*
- *Interpolations between such*
- *...*

UC San Diego

MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

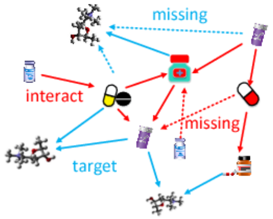
Human learning vs machine learning



Data examples

Type-2 diabetes is 90% more common than type-1

Rules/Constraints



Knowledge graphs

Rewards

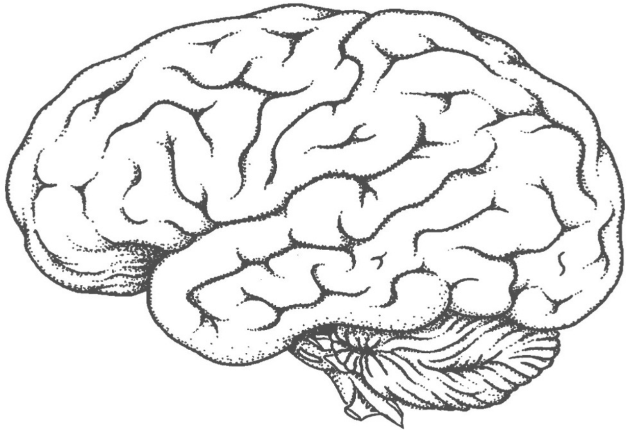
Auxiliary agents

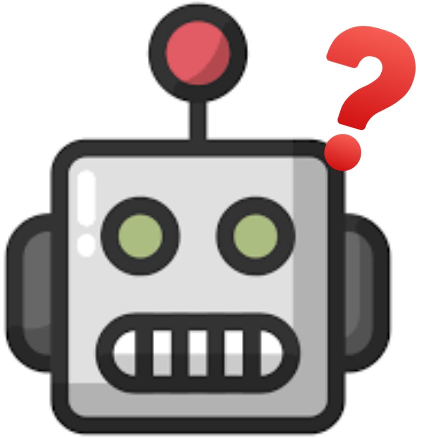
Adversaries

Master classes

…

- *And all combinations of such*
- *Interpolations between such*
- …





Petuum

The zoo of ML/AI models

- Neural networks
  - Convolutional networks
  - AlexNet, GoogleNet, ResNet
  - Recurrent networks, LSTM
  - Transformers
  - BERT, GPTs
- Graphical models
  - Bayesian networks
  - Markov Random fields
  - Topic models, LDA
  - HMM, CRF
- Kernel machines
  - Radial Basis Function Networks
  - Gaussian processes
  - Deep kernel learning
  - Maximum margin
  - SVMs
- Decision trees
- PCA, Probabilistic PCA, Kernel PCA, ICA
- Boosting

The zoo of ML/AI algorithms

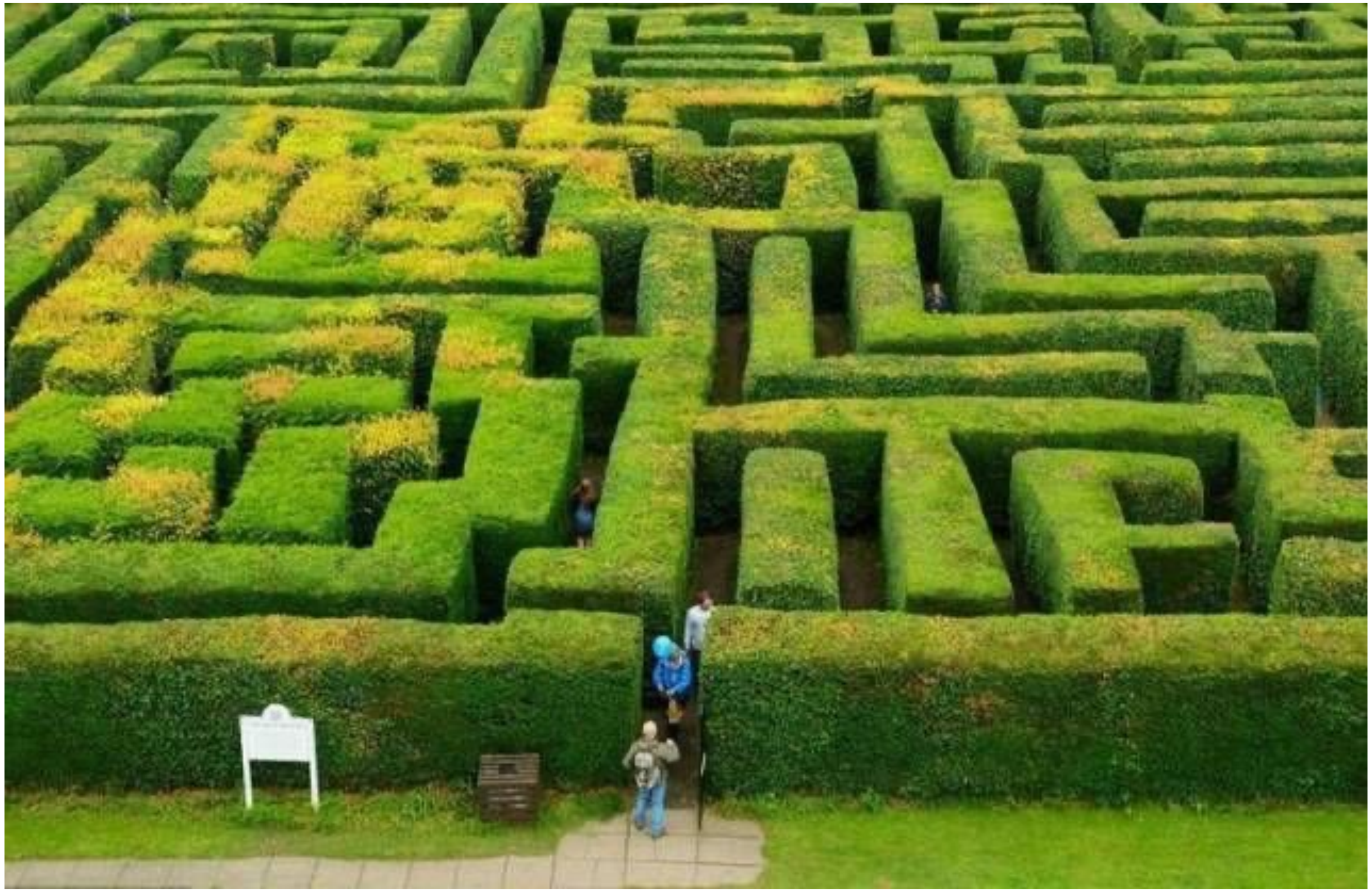
The zoo of ML/AI algorithms

maximum likelihood estimation
reinforcement learning as inference
data re-weighting
inverse RL
policy optimization
active learning
data augmentation
actor-critic
reward-augmented maximum likelihood
label smoothing
imitation learning
softmax policy gradient
adversarial domain adaptation
posterior regularization
GANs
constraint-driven learning
knowledge distillation
intrinsic reward
generalized expectation
prediction minimization
regularized Bayes
learning from measurements
energy-based GANs
weak/distant supervision

UC San Diego

MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

Really hard to navigate, and to realize

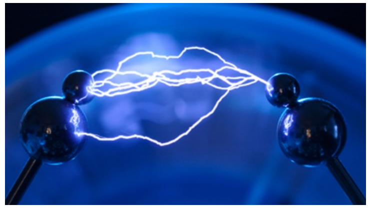


- Depending on individual's expertise and creativity
- Bespoke, delicate pieces of art
- Like an airport with different runways for every different types of aircrafts

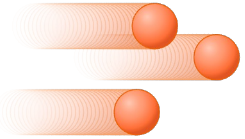
Petuum

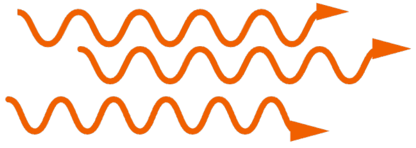
Physics in the 1800's

- Electricity & magnetism:
  - Coulomb's law, Ampère, Faraday, ...



- Theory of light beams:
  - Particle theory: Isaac Newton, Laplace, Plank
  - Wave theory: Grimaldi, Chris Huygens, Thomas Young, Maxwell





- Law of gravity
  - Aristotle, Galileo, Newton, …

UC San Diego

MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

Standard Model in Physics

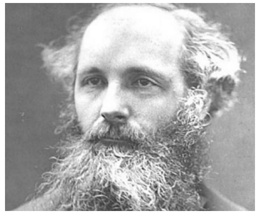
Diverse electro-magnetic theories

Maxwell's Eqns: original form

$e+\frac{df}{dx}+\frac{dg}{dy}+\frac{dh}{dz}=0$	(1)	Gauss' Law
$\mu\alpha=\frac{dH}{dy}-\frac{dG}{dz}$ $\mu\beta=\frac{dF}{dz}-\frac{dH}{dx}$ $\mu\gamma=\frac{dG}{dx}-\frac{dF}{dy}$	(2)	Equivalent to Gauss' Law for magnetism
$P=\mu\left(\gamma\frac{dy}{dt}-\beta\frac{dz}{dt}\right)-\frac{dF}{dt}-\frac{d\Psi}{dx}$ $Q=\mu\left(\alpha\frac{dz}{dt}-\gamma\frac{dx}{dt}\right)-\frac{dG}{dt}-\frac{d\Psi}{dy}$ $R=\mu\left(\beta\frac{dx}{dt}-\alpha\frac{dy}{dt}\right)-\frac{dH}{dt}-\frac{d\Psi}{dz}$	(3)	Faraday's Law (with the Lorentz Force and Poisson's Law)
$\frac{d\gamma}{dy}-\frac{d\beta}{dz}=4\pi p'$ $\quad p'=p+\frac{df}{dt}$ $\frac{d\alpha}{dz}-\frac{d\gamma}{dx}=4\pi q'$ $\quad q'=q+\frac{dg}{dt}$ $\frac{d\beta}{dx}-\frac{d\alpha}{dy}=4\pi r'$ $\quad r'=r+\frac{dh}{dt}$	(4)	Ampère-Maxwell Law
$P=-\zeta p \quad Q=-\zeta q \quad R=-\zeta r$		Ohm's Law
$P=kf \quad Q=kg \quad R=kh$		The electric elasticity equation ($\mathbf{E}=\mathbf{D}/\varepsilon$)
$\frac{de}{dt}+\frac{dp}{dx}+\frac{dq}{dy}+\frac{dr}{dz}=0$		Continuity of charge

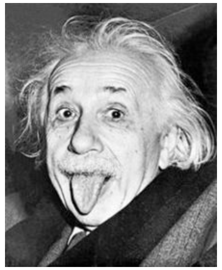
Simplified w/ rotational symmetry

$$\nabla\cdot\mathbf{D}=\rho_V$$
$$\nabla\cdot\mathbf{B}=0$$
$$\nabla\times\mathbf{E}=-\frac{\partial\mathbf{B}}{\partial t}$$
$$\nabla\times\mathbf{H}=\frac{\partial\mathbf{D}}{\partial t}+\mathbf{J}$$



Further simplified w/ symmetry of special relativity

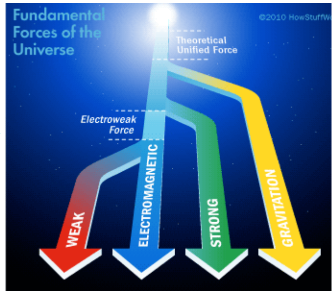
$$\varepsilon^{uvk\lambda}\partial_v F_{k\lambda}=0$$
$$\partial_v F^{uV}=\frac{4\pi}{c}j^u$$



Standard Model w/ Yang-Mills theory and US(3) symmetry

$$\mathcal{L}_{gf}=-\frac{1}{2}\mathrm{Tr}(F^2)=-\frac{1}{4}F^{a\mu\nu}F^a_{\mu\nu}$$

Unification of fundamental forces?



1861 1910s 1970s ?

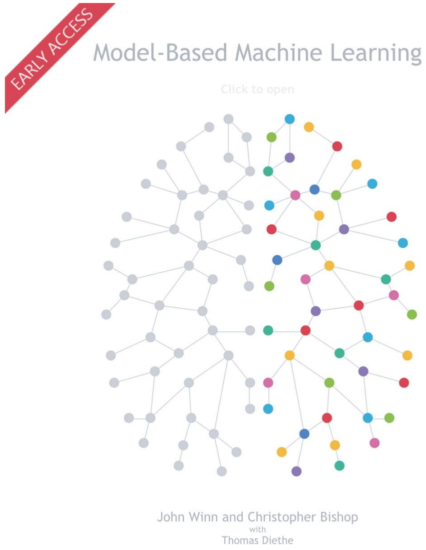
Quest for more standardized, unified ML principles

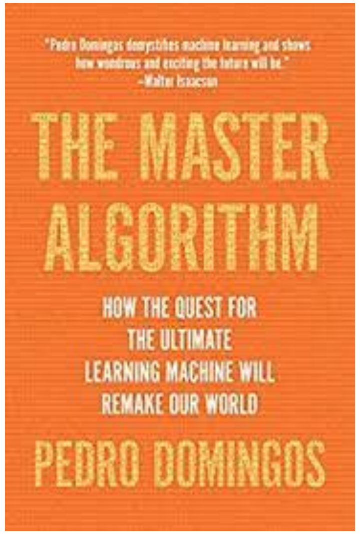
Machine Learning 3: 253–259, 1989

EDITORIAL

Toward a Unified Science of Machine Learning

[P. Langley, 1989]





REVIEW Communicated by Steven Nowlan

A Unifying Review of Linear Gaussian Models

Sam Roweis*
Computation and Neural Systems, California Institute of Technology, Pasadena, CA 91125, U.S.A.

Zoubin Ghahramani*
Department of Computer Science, University of Toronto, Toronto, Canada

Toward A “Standard Model” of ML

- Loss
- Experience
- Optimization solver
- Model architecture

$$\min_\theta \mathcal{L}(\theta, \mathcal{E})$$

Optimization solver — Loss — Model architecture — Experience

Toward A “Standard Model” of ML

- Loss
- Experience
- Optimization solver
- Model architecture

$$\min_{q,\theta} \; -\mathbb{E} + \mathbb{D} - \mathbb{H}$$

Experience Divergence Uncertainty

Toward A "Standard Model" of ML

Toward a 'Standard Model' of Machine Learning

Zhiting Hu[†,*], Eric P. Xing[‡,◇,♮,**]
[†] Halıcıoğlu Data Science Institute, University of California San Diego, San Diego, USA
[‡] Machine Learning Department, Carnegie Mellon University, Pittsburgh, USA
[♮] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[◇] Petuum Inc., Pittsburgh, USA

[Hu & Xing, Harvard Data Science Review, 2022]: https://arxiv.org/abs/2108.07783

$$\min_{q,\theta} \; -\mathbb{E} + \mathbb{D} - \mathbb{H}$$

Experience Divergence Uncertainty

Maximum likelihood estimation (MLE) at a close look:

- The most classical learning algorithm

- Supervised:
  - Observe data $\mathcal{D} = \{(\boldsymbol{x}^*, \boldsymbol{y}^*)\}$
  - Solve with SGD

$$\min_\theta - \mathbb{E}_{(\boldsymbol{x}^*,\boldsymbol{y}^*)\sim\mathcal{D}}\left[\log p_\theta(\boldsymbol{y}^*|\boldsymbol{x}^*)\right]$$

- Unsupervised:
  - Observe $\mathcal{D} = \{(\boldsymbol{x}^*)\}$, $\boldsymbol{y}$ is latent variable
  - Posterior $p_\theta(\boldsymbol{y}|\boldsymbol{x})$
  - Solve with EM:
    - E-step imputes latent variable $\boldsymbol{y}$ through expectation on complete likelihood
    - M-step: supervised MLE

$$\min_\theta - \mathbb{E}_{\boldsymbol{x}^*\sim\mathcal{D}}\left[\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y})\right]$$

MLE as Entropy Maximization

- Duality between supervised MLE and maximum entropy, when p is exponential family

Shannon entropy H

$$\min_{p(\boldsymbol{x},\boldsymbol{y})} \; H(p)$$

features $T(\boldsymbol{x},\boldsymbol{y})$

$$s.t. \;\; \mathbb{E}_p[T(\boldsymbol{x},\boldsymbol{y})] = \mathbb{E}_{(\boldsymbol{x}^*,\boldsymbol{y}^*)\sim\mathcal{D}}[T(\boldsymbol{x},\boldsymbol{y})]$$

data as constraints

Solve w/ Lagrangian method $\Downarrow$

$$p(\boldsymbol{x},\boldsymbol{y}) = \exp\{\boldsymbol{\theta}\cdot T(\boldsymbol{x})\} \;/\; Z(\boldsymbol{\theta})$$

Lagrangian multiplier $\boldsymbol{\theta}$

$$\min_{\theta} -\mathbb{E}_{(\boldsymbol{x}^*,\boldsymbol{y}^*)\sim\mathcal{D}}[\boldsymbol{\theta}\cdot T(\boldsymbol{x},\boldsymbol{y})] + \log Z(\boldsymbol{\theta})$$

--> *Negative log-likelihood*

How to estimate θ – Close form? SGD?

MLE as Entropy Maximization

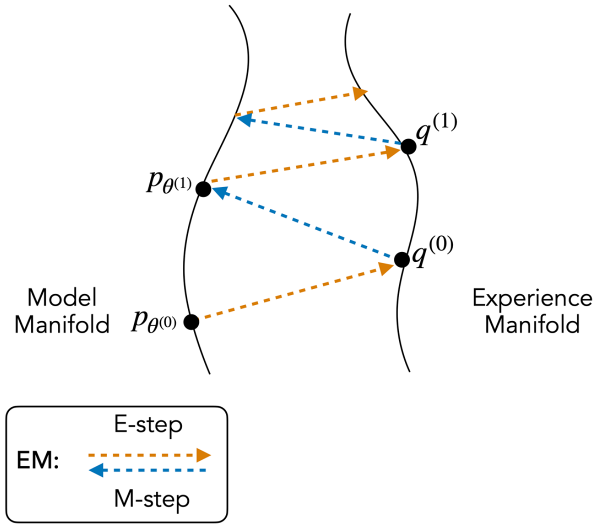
- Unsupervised MLE can be achieved by maximizing the negative free energy:
  - Introduce an auxiliary variational distribution $q(\boldsymbol{y}|\boldsymbol{x})$ (and then play with its entropy and cross entropy, etc.)

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) = \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}\left[\log \frac{p_\theta(\boldsymbol{x}^*, \boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x}^*)}\right] + \mathrm{KL}\big(q(\boldsymbol{y}|\boldsymbol{x}^*) \,||\, p_\theta(\boldsymbol{y}|\boldsymbol{x}^*)\big)$$

$$\geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$$

Algorithms for Unsupervised MLE

Alternating projection



$$\min_{\theta} - \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D}} \left[\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) \right]$$

1) Solve with EM

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) = \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)} \left[\log \frac{p_\theta(\boldsymbol{x}^*, \boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x}^*)} \right] + \mathrm{KL}\big(q(\boldsymbol{y}|\boldsymbol{x}^*) \,||\, p_\theta(\boldsymbol{y}|\boldsymbol{x}^*)\big)$$
$$\geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$$

- E-step: Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t q, equivalent to minimizing KL by setting

  $$q(\boldsymbol{y}|\boldsymbol{x}^*) = p_{\theta^{old}}(\boldsymbol{y}|\boldsymbol{x}^*)$$

- M-step: Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$: $\max_{\theta} \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$

Algorithms for Unsupervised MLE (cont'd)

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) = \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}\left[\log \frac{p_\theta(\boldsymbol{x}^*, \boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x}^*)}\right] + \mathrm{KL}\big(q(\boldsymbol{y}|\boldsymbol{x}^*) \,||\, p_\theta(\boldsymbol{y}|\boldsymbol{x}^*)\big)$$

$$\geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$$

2) When model $p_{\boldsymbol{\theta}}$ is complex, directly working with the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}^*)$ is intractable ⇒ Variational EM
   - Consider a sufficiently restricted family Q of $q(\boldsymbol{y}|\boldsymbol{x})$ so that minimizing the KL is tractable
     - E.g., parametric distributions, factorized distributions

   - E-step: Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t $q \in Q$, equivalent to minimizing KL
   - M-step: Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$: $\max_\theta \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$

Algorithms for Unsupervised MLE (cont'd)

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) = \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}\left[\log \frac{p_\theta(\boldsymbol{x}^*, \boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x}^*)}\right] + \mathrm{KL}\big(q(\boldsymbol{y}|\boldsymbol{x}^*) \,||\, p_\theta(\boldsymbol{y}|\boldsymbol{x}^*)\big)$$

$$\geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$$

3) When q is complex, e.g., deep NNs, optimizing q in E-step is difficult (e.g., high variance) ⇒ Wake-Sleep algorithm [Hinton et al., 1995]

- Sleep-phase (E-step): $\min_\phi \mathrm{KL}(p_\theta(\boldsymbol{y}|\boldsymbol{x}^*) || q_\phi(\boldsymbol{y}|\boldsymbol{x}^*))$ -----> *Reverse KL*

- Wake-phase (M-step): Maximize $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$: $\max_\theta \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$

Other tricks: reparameterization in VAE ('2014), control variates in NVIL ('2014)

Quick summary of MLE

- Supervised:
  - Duality with MaxEnt
  - Solve with SGD
- Unsupervised:
  - Lower bounded by negative free energy
  - Solve with EM, VEM, Wake-Sleep, …

- Close connections to MaxEnt
- With MaxEnt, algorithms (e.g., EM) arises naturally

Posterior Regularization (PR)

- Make use of constraints in Bayesian learning
  - An auxiliary posterior distribution $q(\theta)$
  - Slack variable ξ, constant weight $\alpha = \beta > 0$

$$\min_{q,\,\xi} \; -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] + \xi$$

$$s.t. \; -\mathbb{E}_q [f_\theta(\boldsymbol{x}, \boldsymbol{y})] \leq \xi$$

[Ganchev et al., 2010]

  - E.g., max-margin constraint for linear regression [Jaakkola et al., 1999] and general models (e.g., LDA, NNs) [Zhu et al., 2014]

- Solution for q

$$q(\boldsymbol{\theta}) = \exp\left\{ \frac{\beta \log p_\theta(\boldsymbol{x}, \boldsymbol{y}) + f_\theta(\boldsymbol{x}, \boldsymbol{y})}{\alpha} \right\} / Z$$

More general learning leveraging PR

- No need to limit to Bayesian learning
- E.g., Complex rule constraints on general models [Hu et al., 2016], where
  - q can be over arbitrary variables, e.g., $q(\boldsymbol{x}, \boldsymbol{y})$
  - $p_\theta(\boldsymbol{x}, \boldsymbol{y})$ is NNs of arbitrary architectures with parameters $\boldsymbol{\theta}$

$$\min_{q,\theta,\xi} -\alpha H(q) - \beta \mathbb{E}_q \Big[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \Big] + \xi$$

$$s.t.\ \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \Big[1 - r(\boldsymbol{x}, \boldsymbol{y}) \Big] \leq \xi$$

E.g., $r(\boldsymbol{x}, \boldsymbol{y})$ is a 1st-order logical rule:

If sentence $\boldsymbol{x}$ contains word ``but''
⇒ its sentiment $\boldsymbol{y}$ is the same as the sentiment after “but”

EM for the general PR

- Rewrite without slack variable:

$$\min_{q,\theta} -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

  - Solve with EM
    - E-step: $q(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{ \frac{\beta \log p_\theta(\boldsymbol{x}, \boldsymbol{y}) + f(\boldsymbol{x}, \boldsymbol{y})}{\alpha} \right\} / Z$
    - M-step: $\min_{\theta} \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right]$

Reformulating unsupervised MLE with PR

$$\log \int_{\boldsymbol{y}} p_\theta(\boldsymbol{x}^*, \boldsymbol{y}) \geq H\big(q(\boldsymbol{y}|\boldsymbol{x}^*)\big) + \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x}^*)}[\log p_\theta(\boldsymbol{x}^*, \boldsymbol{y})]$$

- Introduce arbitrary $q(\boldsymbol{y}|\boldsymbol{x})$

$$\min_{q,\theta,\xi} \; -\alpha H(q) - \beta \mathbb{E}_q\Big[\log p_\theta(\boldsymbol{x}, \boldsymbol{y})\Big] + \xi$$

$$s.t. \; -\mathbb{E}_q\Big[f(\boldsymbol{x}\,;\,\mathcal{D})\Big] < \xi$$

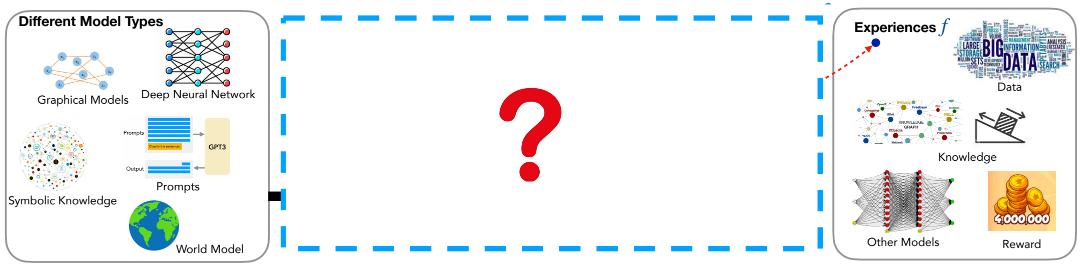
Data as constraint.
Given $\boldsymbol{x} \sim \mathcal{D}$, this constraint doesn't influence the solution of q and $\boldsymbol{\theta}$

  - $f(\boldsymbol{x}\,;\mathcal{D}) := \log \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D}}[\,\mathbb{1}_{\boldsymbol{x}^*}(\boldsymbol{x})\,]$
    - A constraint saying $\boldsymbol{x}$ must equal to one of the true data points
    - Or alternatively, the (log) expected similarity of $\boldsymbol{x}$ to dataset $\mathcal{D}$, with $\mathbb{1}(\cdot)$ as the similarity measure (we'll come back to this later)
  - $\alpha = \beta = 1$

A “Standard Model” of Machine Learning

Petuum

The Standard Equation (SE)



$$\min_{q,\theta,\xi\geq 0} \beta \mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \alpha \mathbb{H}(q) + \xi$$

$$s.t. \; -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] < \xi$$

Equivalently:

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \beta \mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \alpha \mathbb{H}(q)$$

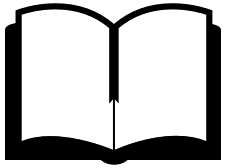
3 terms:

Experience
(exogenous regularizations)
e.g., data examples, rules

Divergence
(fitness)
e.g., Cross Entropy

Uncertainty
(self-regularization)
e.g., Shannon entropy

Textbook
$f(\boldsymbol{x},\boldsymbol{y} | \cdot)$



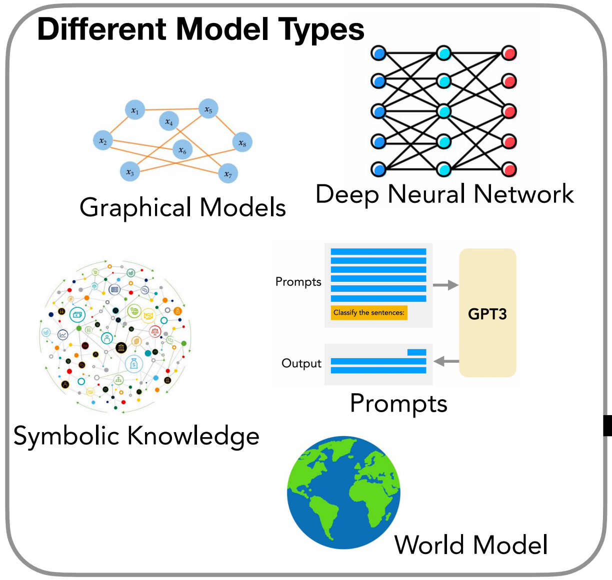
Teacher
$q(\boldsymbol{x},\boldsymbol{y})$

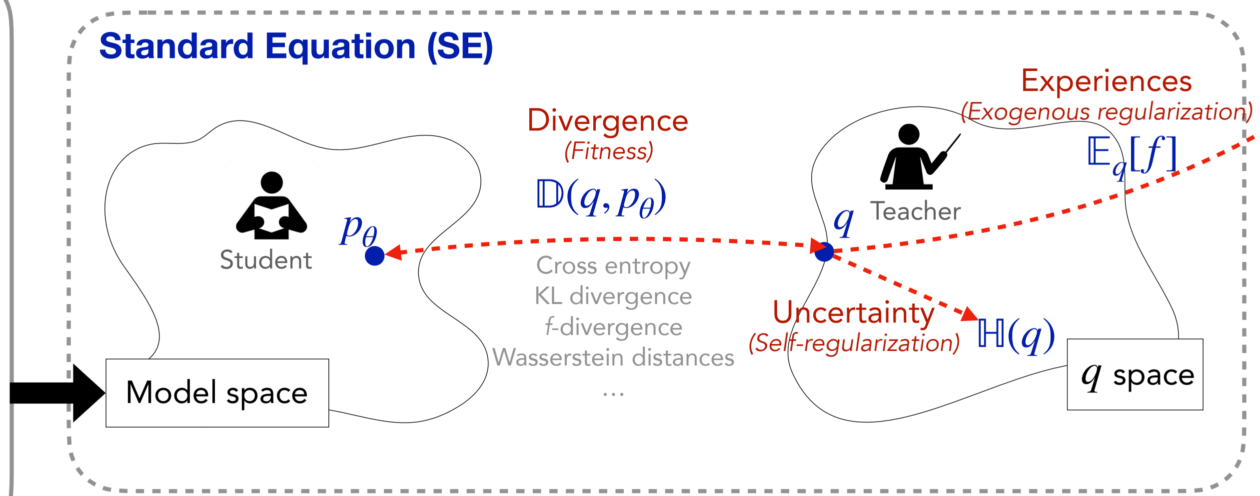
Student
$p_\theta(\boldsymbol{x},\boldsymbol{y})$

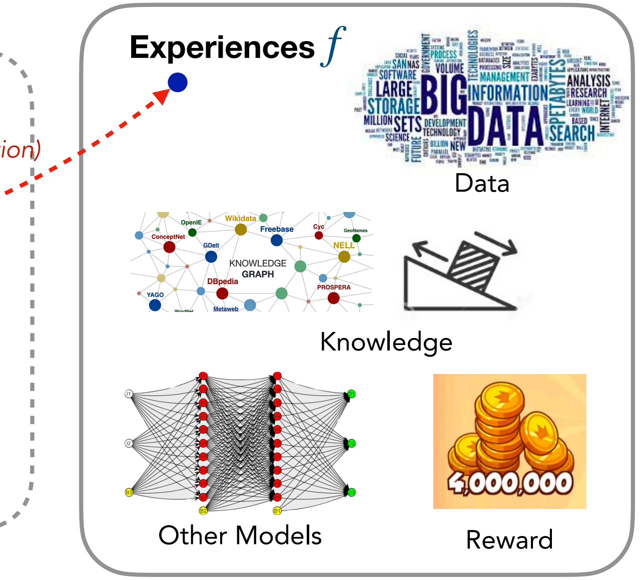
Uncertainty

The Standard Equation (SE)

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[f(\boldsymbol{x},\boldsymbol{y})\Big] + \beta\mathbb{D}\Big(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\Big) - \alpha\mathbb{H}(q)$$







[Note: in SE, experience function f can also depends on θ. See the paper for mor details]

Overview: well-known algorithms/paradigms recovered by SE

Experience type	Experience function f	Divergence $\mathbb{D}$	α	β	Algorithm
Data instances	$f_{\text{data}}(\boldsymbol{x}; \mathcal{D})$	CE	1	1	Unsupervised MLE
	$f_{\text{data}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Supervised MLE
	$f_{\text{data-self}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Self-supervised MLE
	$f_{\text{data-w}}(\boldsymbol{t}; \mathcal{D})$	CE	1	ϵ	Data Re-weighting
	$f_{\text{data-aug}}(\boldsymbol{t}; \mathcal{D})$	CE	1	ϵ	Data Augmentation
	$f_{\text{active}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Active Learning (Ertekin et al., 2007)
Knowledge	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	Posterior Regularization (Ganchev et al., 2010)
	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	CE	$\mathbb{R}$	1	Unified EM (Samdani et al., 2012)
Reward	$\log Q^{\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	Policy Gradient
	$\log Q^{\theta}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in,\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	+ Intrinsic Reward
	$Q^{\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	$\rho > 0$	$\rho > 0$	RL as Inference
Model	$f^{\text{mimicking}}_{\text{model}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Knowledge Distillation (G. Hinton et al., 2015)
Variational	binary classifier	JSD	0	1	Vanilla GAN (Goodfellow et al., 2014)
	discriminator	f-divergence	0	1	f-GAN (Nowozin et al., 2016)
	1-Lipschitz discriminator	W_1 distance	0	1	WGAN (Arjovsky et al., 2017)
	1-Lipschitz discriminator	KL	0	1	PPO-GAN (Y. Wu et al., 2020)
Online	$f_{\tau}(\boldsymbol{t})$	CE	$\rho > 0$	$\rho > 0$	Multiplicative Weights (Freund & Schapire, 1997)

SE Component: Experience Function f

Different choices of experience function f lead to different algorithms:

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \beta\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \alpha\mathbb{H}(q)$$

Experience
(exogenous regularizations)
e.g., data examples, rules

Set *Divergence* to *Cross Entropy*
$\mathbb{D}(q, p_\theta) = -\mathbb{E}_q[\log p_\theta]$

Set *Uncertainty* to *Shannon Entropy*
$\mathbb{H}(q) = H(q) := -\mathbb{E}_q[\log q]$

SE with Data Experience -- Supervised MLE

Observe data $\mathcal{D} = \{(\boldsymbol{x}^*, \boldsymbol{y}^*)\}$

$$\min_{q,\theta} -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_q \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

$$f := f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}) = \log \mathbb{E}_{(\boldsymbol{x}^*, \boldsymbol{y}^*) \sim \mathcal{D}} \left[\mathbb{1}_{(\boldsymbol{x}^*, \boldsymbol{y}^*)}(\boldsymbol{x}, \boldsymbol{y}) \right] \qquad \alpha = 1, \beta = \epsilon$$

Teacher step: $q(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{ \frac{\beta \log p_\theta(\boldsymbol{x}, \boldsymbol{y}) + f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})}{\alpha} \right\} / Z \approx \exp\{f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})\}/Z = \tilde{p}_d(\boldsymbol{x}, \boldsymbol{y})$

Student step: $\min_{\theta} -\mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right]$ → Negative data log-likelihood

SE with Data Experience -- Unsupervised MLE

Observe data $\mathcal{D} = \{(\boldsymbol{x}^*)\}$

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_q \left[\; f(\boldsymbol{x}, \boldsymbol{y}) \; \right]$$

$$f := f(\boldsymbol{x}\,;\,\mathcal{D}) = \log \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D}} [\, \mathbb{1}_{\boldsymbol{x}^*}(\boldsymbol{x}) \,] \qquad \alpha = \beta = 1$$

$$q = q(\boldsymbol{y} | \boldsymbol{x})$$

↓

$$\min_{q,\theta} \; - H(q) - \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right]$$

→ Negative variational lower bound

SE with “Oracle Data Experience”: Active Learning

- Have access to a vast pool of unlabeled data instances
- Can select instances (queries) to be labeled by an oracle (e.g., human)

- Experiences:
  - $u(\boldsymbol{x})$ measures *informativeness* of an instance $\boldsymbol{x}$
    - e.g., Uncertainty on $\boldsymbol{x}$, measured by Shannon entropy $H(\, p_\theta(\boldsymbol{y}|\boldsymbol{x})\,)$
  - Encode instances + oracle labels:

$$f(\boldsymbol{x}, \boldsymbol{y}\,;\ \text{Oracle}) = \log \mathbb{E}_{\boldsymbol{x}^* \sim \mathcal{D},\, \boldsymbol{y}^* \sim \text{Oracle}(\boldsymbol{x}^*)}\left[\, \mathbb{1}_{(\boldsymbol{x}^*, \boldsymbol{y}^*)}(\boldsymbol{x}, \boldsymbol{y})\,\right]$$

SE and Active Learning

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q\left[\log p_\theta(\boldsymbol{x},\boldsymbol{y})\right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[\; f(\boldsymbol{x},\boldsymbol{y}) \;\right]$$

$$f := f(\boldsymbol{x},\boldsymbol{y}\,;\,Oracle) + u(\boldsymbol{x}) \qquad\qquad \alpha = 1, \beta = \epsilon$$

- Teacher step: $q(\boldsymbol{x},\boldsymbol{y}) = \exp\left\{\frac{\beta \log p_\theta(\boldsymbol{x},\boldsymbol{y}) + f(\boldsymbol{x},\boldsymbol{y}\,;Oracle) + u(\boldsymbol{x})}{\alpha}\right\} / Z$

- Student step: $\min_\theta -\mathbb{E}_q\left[\log p_\theta(\boldsymbol{x},\boldsymbol{y})\right]$

Equivalent to [e.g., Ertekin et al., 07]:
- Randomly draw a subset $\mathcal{D}_{sub} = \{\boldsymbol{x}^*\}$
- Draw a query $\boldsymbol{x}^*$ from $\mathcal{D}_{sub}$ according to $\exp\{u(\boldsymbol{x})\}$
- Get label $\boldsymbol{y}^*$ for $\boldsymbol{x}^*$ from the oracle
- Maximize log likelihood on $(\boldsymbol{x}^*, \boldsymbol{y}^*)$

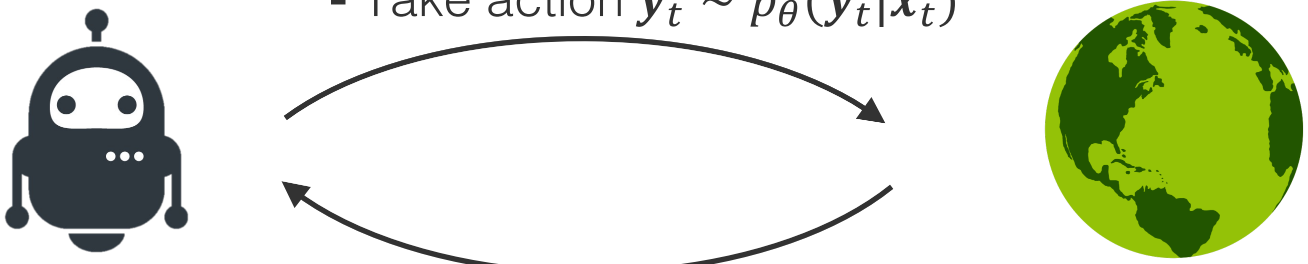
SE with Reward Experience

Markov Decision Process (MDP)

AGENT

- State $\boldsymbol{x}_t$
- Take action $\boldsymbol{y}_t \sim p_\theta(\boldsymbol{y}_t | \boldsymbol{x}_t)$

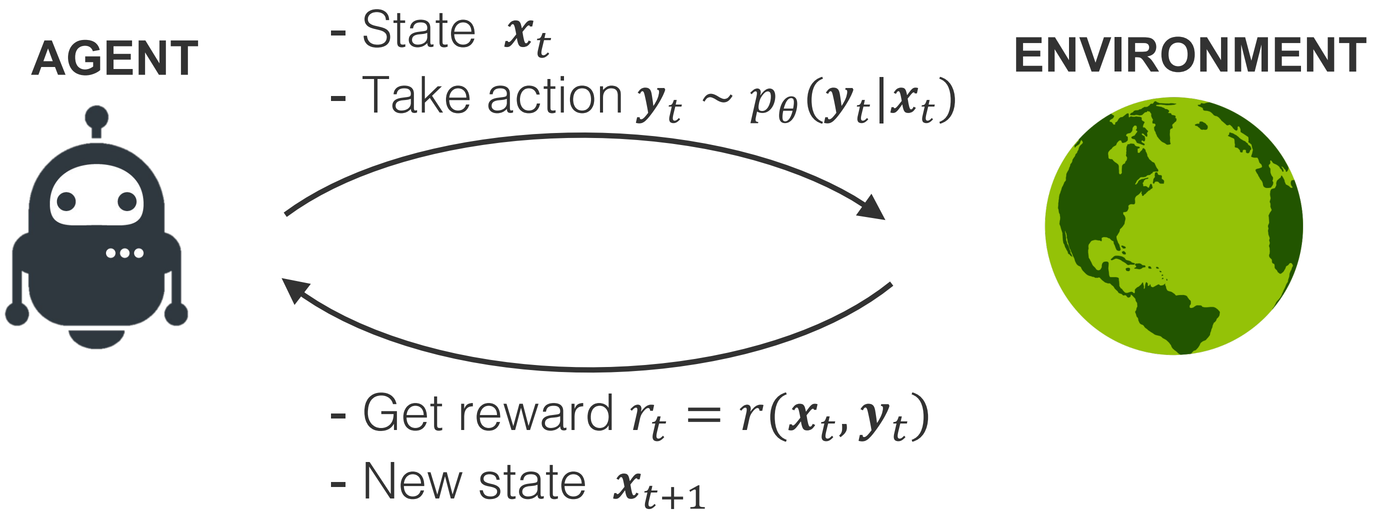
ENVIRONMENT



- Get reward $r_t = r(\boldsymbol{x}_t, \boldsymbol{y}_t)$
- New state $\boldsymbol{x}_{t+1}$

SE with Reward Experience: Reinforcement Learning

Markov Decision Process (MDP)



- $p_\theta(\boldsymbol{x}, \boldsymbol{y}) = p_\theta(\boldsymbol{y}|\boldsymbol{x})p_0(\boldsymbol{x})$, where $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ is the policy, $p_0(\boldsymbol{x})$ is the start state distribution
- $Q^\theta(\boldsymbol{x}, \boldsymbol{y})$ – expected future reward of taking action $\boldsymbol{y}$ in state $\boldsymbol{x}$ and continuing the current policy p_θ

$$Q^\theta(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{p_\theta}\left[\sum_{t=0}^{\infty} r_t \mid \boldsymbol{x}_0 = \boldsymbol{x}, \boldsymbol{y}_0 = \boldsymbol{y}\right]$$

- $\mu^\theta(\boldsymbol{x})$ – state distribution

$$\mu^\theta(\boldsymbol{x}) = \sum_{t=0}^{\infty} p(\boldsymbol{x}_t = \boldsymbol{x})$$

SE with Reward Experience I: Policy Gradient

$$\min_{q,\theta} \; -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_q \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

- Policy gradient

$$f^\theta(\boldsymbol{x}, \boldsymbol{y}) := \log Q^\theta(\boldsymbol{x}, \boldsymbol{y}) \qquad \alpha = \beta = 1$$

  - Teacher step: $q^{(n)}(\boldsymbol{x}, \boldsymbol{y}) = p_{\theta^{(n)}}(\boldsymbol{x}, \boldsymbol{y}) Q^{\theta^{(n)}}(\boldsymbol{x}, \boldsymbol{y}) \,/\, Z$
  - Student step:

$$\mathbb{E}_{q^{(n)}(\boldsymbol{x},\boldsymbol{y})} \left[\nabla_\theta \log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] + \mathbb{E}_{q^{(n)}(\boldsymbol{x},\boldsymbol{y})} \left[\nabla_\theta f^\theta_{\text{reward},1}(\boldsymbol{x}, \boldsymbol{y}) \right] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$$

$$= 1/Z \cdot \sum_{\boldsymbol{x}} p_0(\boldsymbol{x}) \nabla_\theta \sum_{\boldsymbol{y}} p_\theta(\boldsymbol{y}|\boldsymbol{x}) Q^\theta(\boldsymbol{x}, \boldsymbol{y}) \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$$

(log-derivative trick)

$$= 1/Z \cdot \sum_{\boldsymbol{x}} \mu^\theta(\boldsymbol{x}) \sum_{\boldsymbol{y}} Q^\theta(\boldsymbol{x}, \boldsymbol{y}) \nabla_\theta p_\theta(\boldsymbol{y}|\boldsymbol{x}) \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}}$$

(policy gradient theorem)

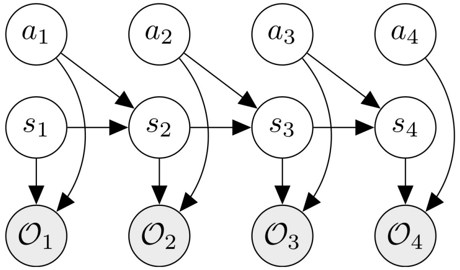
policy gradient

SE with Reward Experience II: RL as Inference

$$\min_{q,\theta} -\alpha H(q) - \beta \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_q \left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

- RL-as-inference [Dayan'97; Levine'18, …]

$$f^\theta(\boldsymbol{x}, \boldsymbol{y}) := Q^\theta(\boldsymbol{x}, \boldsymbol{y}) \qquad \alpha = \beta = \rho\ (> 0)$$



$$\min_{q,\theta} -\rho H(q) - \rho \mathbb{E}_q \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{y}) \right] - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})} \left[Q^\theta(\boldsymbol{x}, \boldsymbol{y}) \right]$$

$$\geq -\log \mathbb{E}_{p_\theta(\boldsymbol{x},\boldsymbol{y})} [\, p(o = 1 \mid \boldsymbol{x}, \boldsymbol{y}) \,]$$

Negative variational lower bound

Define random variable $o \in \{0,1\}$, $p(o = 1) \propto \exp\{ Q^{\theta^t}(\boldsymbol{x}, \boldsymbol{y}) / \rho \}$ (reward excitement fuc.)

SE with Other Experience

Experience type	Experience function f	Divergence $\mathbb{D}$	α	β	Algorithm
Data instances	$f_{\text{data}}(\boldsymbol{x}; \mathcal{D})$	CE	1	1	Unsupervised MLE
	$f_{\text{data}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Supervised MLE
	$f_{\text{data-self}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Self-supervised MLE
	$f_{\text{data-w}}(\boldsymbol{t}; \mathcal{D})$	CE	1	ϵ	Data Re-weighting
	$f_{\text{data-aug}}(\boldsymbol{t}; \mathcal{D})$	CE	1	ϵ	Data Augmentation
	$f_{\text{active}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Active Learning (Ertekin et al., 2007)
Knowledge	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	Posterior Regularization (Ganchev et al., 2010)
	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	CE	$\mathbb{R}$	1	Unified EM (Samdani et al., 2012)
Reward	$\log Q^{\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	Policy Gradient
	$\log Q^{\theta}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in,\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	+ Intrinsic Reward
	$Q^{\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	$\rho > 0$	$\rho > 0$	RL as Inference
Model	$f_{\text{model}}^{\text{mimicking}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Knowledge Distillation (G. Hinton et al., 2015)
Variational	binary classifier	JSD	0	1	Vanilla GAN (Goodfellow et al., 2014)
	discriminator	f-divergence	0	1	f-GAN (Nowozin et al., 2016)
	1-Lipschitz discriminator	W_1 distance	0	1	WGAN (Arjovsky et al., 2017)
	1-Lipschitz discriminator	KL	0	1	PPO-GAN (Y. Wu et al., 2020)
Online	$f_{\tau}(\boldsymbol{t})$	CE	$\rho > 0$	$\rho > 0$	Multiplicative Weights (Freund & Schapire, 1997)

See paper for more details

SE Component: Divergence Function $\mathbb{D}$

We now look at the choices of divergence $\mathbb{D}$:

$$\min_{q,\theta} - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y}) \right] + \beta \mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y}) \right) - \alpha \mathbb{H}(q)$$

Divergence
(fitness)
e.g., Cross Entropy

SE with Cross Entropy or KL Divergence

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \beta\mathbb{D}\Big(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\Big) - \alpha\mathbb{H}(q)$$

↓

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \beta\mathbb{E}_q\left[\log p_\theta(\boldsymbol{x},\boldsymbol{y})\right] - \alpha\mathbb{H}(q)$$

All the algorithms we've just seen

SE with Other Divergences

- For notation simplicity, we use $\boldsymbol{x}$ to replace $(\boldsymbol{x}, \boldsymbol{y})$

$$\min_{q,\theta} \; -\alpha \mathbb{H}(q) + \beta \mathbb{D}\Big(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[f(\boldsymbol{x}) \Big]$$

SE with Other Divergences

- For notation simplicity, we use $\boldsymbol{x}$ to replace $(\boldsymbol{x}, \boldsymbol{y})$

$$\min_{q,\theta} \; -\alpha \mathbb{H}(q) + \beta \mathbb{D}\Big(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[\; f(\boldsymbol{x}) \; \Big]$$

- Same as supervised MLE: $f \coloneqq f(\boldsymbol{x}\,; \mathcal{D}), \;\; \alpha = 1, \;\; \beta = \epsilon$
- Equivalent to $\min_\theta \mathbb{D}\Big(p_d(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big)$

SE with Other Divergences

- For notation simplicity, we use $\boldsymbol{x}$ to replace $(\boldsymbol{x}, \boldsymbol{y})$

$$\min_{q,\theta} \; -\alpha \mathbb{H}(q) + \beta \mathbb{D}\left(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \right) - \mathbb{E}_{q(\boldsymbol{x})}\left[f(\boldsymbol{x}) \right]$$

- Same as supervised MLE: $f \coloneqq f(\boldsymbol{x}\,;\mathcal{D}),\; \alpha = 1,\; \beta = \epsilon$
- Equivalent to $\min_\theta \mathbb{D}\left(p_d(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \right)$
- Solve with probability functional descent (PFD) [Chu et al., 2019]
  - $p_\theta(\boldsymbol{x})$ can be optimized by minimizing $\mathbb{E}_{p_\theta}[\Psi(\boldsymbol{x})]$, where $\Psi(\boldsymbol{x})$ is the influence function for $\mathbb{D}$ at p_{θ^t}
  - Ψ is obtained with convex duality

$$\Psi(\boldsymbol{x}) = \operatorname{argmax}_\psi \mathbb{E}_{p_\theta}[\psi(\boldsymbol{x})] - \mathbb{D}^*(\psi)$$

Convex conjugate of $\mathbb{D}$

  - So the whole optimization is

$$\min_\theta \max_\psi \mathbb{E}_{p_\theta}[\psi(\boldsymbol{x})] - \mathbb{D}^*(\psi)$$

UC San Diego

MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

SE with JS Divergence: Generative Adversarial Learning (GANs)

$$\min_{\theta} \mathbb{D}\left(p_d(\boldsymbol{x}), p_\theta(\boldsymbol{x})\right)$$

- Solve with probability functional descent (PFD) [Chu et al., 2019]
  - $p_\theta(\boldsymbol{x})$ can be optimized by minimizing $\mathbb{E}_{p_\theta}[\Psi(\boldsymbol{x})]$, where $\Psi(\boldsymbol{x})$ is the influence function for $\mathbb{D}$ at p_{θ^t}
  - Ψ is obtained with convex duality

$$\Psi(\boldsymbol{x}) = \mathrm{argmax}_\psi \, \mathbb{E}_{p_\theta}[\psi(\boldsymbol{x})] - \mathbb{D}^*(\psi)$$

  - So the whole optimization is

$$\min_\theta \max_\psi \mathbb{E}_{p_\theta}[\psi(\boldsymbol{x})] - \mathbb{D}^*(\psi)$$

Parameterize ψ with an NN C_ϕ. E.g., when $\mathbb{D}$ is JSD and

$$\psi_\phi(\boldsymbol{x}) := 0.5 \log\left(1 - C_\phi\right) - 0.5 \log 2$$

Plugging into the equation recovers vanilla GAN training

Jensen-Shannon Divergence: $\mathrm{JS}(q||p_\theta) = \frac{1}{2}\mathrm{KL}(q||h) + \frac{1}{2}\mathrm{KL}(p_\theta||h)$

where $h = \frac{1}{2}(q + p_\theta)$

Petuum

SE with Wasserstein Distance: W-GAN

$$\min_\theta \mathbb{D}\left(p_d(\boldsymbol{x}), p_\theta(\boldsymbol{x})\right)$$

- Based on the Kantorovich duality, the 1st-order Wasserstein distance between two distributions q and p is written as

$$W_1(q, p) = \sup_{||\psi||_L \leq 1} \mathbb{E}_q[\psi(\boldsymbol{x})] - \mathbb{E}_p(\psi(\boldsymbol{x}))$$

  - where $||\psi||_L \leq 1$ is the constraint of ψ: $\mathcal{X} \rightarrow \mathbb{R}$ being a 1-Lipschitz function

- Setting $\mathbb{D}$ to W_1 leads to the Wasserstein GAN algorithm [Arjovsky et al., 2017]

$$\min_\theta W_1(q, p) = \min_\theta \sup_{||\psi||_L \leq 1} \mathbb{E}_{p_d}[\psi(\boldsymbol{x})] - \mathbb{E}_{p_\theta}(\psi(\boldsymbol{x}))$$

Dynamic SE

- So far, we have seen SE as the ultimate learning objective
  - Fully defines the learning problem in an analytical form

$$\min_{q,\theta} - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y}) \right] + \beta \mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y}) \right) - \alpha \mathbb{H}(q)$$

- In a dynamic or online setting, the learning objective itself may be evolving over time
  - Data instances may follow changing distributions or come from evolving tasks (e.g., lifelong learning)
  - Experience in a strategic game context can involve complex interactions with the target model through co-training or adversarial dynamics

Dynamic SE

- So far, we have seen SE as the ultimate learning objective
  - Fully defines the learning problem in an analytical form

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \beta\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\, p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \alpha\mathbb{H}(q)$$

- In a dynamic or online setting, the learning objective itself may be evolving over time
- An extended view of the SE for learning in dynamic contexts
  - SE is a core part of an **outer loop**
  - E.g., consider dynamic experience f_τ indexed by time τ

$$\text{for } \tau = 1, 2, \ldots :$$

$$\text{Acquire experience } f_\tau,$$

$$\text{Solve SE: } \min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f_\tau(\boldsymbol{x},\boldsymbol{y})\right] + \beta\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\, p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \alpha\mathbb{H}(q)$$

Dynamic SE with Adversarial Experience: Variations of GAN

$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\Big(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[f(\boldsymbol{x}) \Big]$$

- Recall in MLE, f is a fixed function

$$f \coloneqq f(\boldsymbol{x}\,;\mathcal{D}) = \log \mathbb{E}_{\boldsymbol{x}^*\sim\mathcal{D}}\left[\mathbb{1}_{\boldsymbol{x}^*}(\boldsymbol{x}) \right]$$

- Intuitively, see f as a similarity metric that measures similarity of sample $\boldsymbol{x}$ against real data $\mathcal{D}$
- Instead of the manually fixed metric, can we learn a metric f_ϕ?

Dynamic SE with Adversarial Experience: Variations of GAN

- Augment the standard objective to account for ϕ:

$$\min_{\theta} \max_{\phi} \min_{q} -\alpha \mathbb{H}(q) + \beta \mathbb{D}\Big(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[f_\phi(\boldsymbol{x}) \Big] + \mathbb{E}_{p_d(\boldsymbol{x})}\Big[f_\phi(\boldsymbol{x}) \Big]$$

- Set $\alpha = 0, \beta = 1$. Under mild conditions, the objective recovers:
  - Vanilla GAN [Goodfellow et al., 2014], when $\mathbb{D}$ is JS Divergence and f_ϕ is a binary classifier
  - f-GAN [Nowozin et al., 2016], when $\mathbb{D}$ is f-divergence
  - W-GAN [Arjovsky et al., 2017], when $\mathbb{D}$ is Wasserstein distance and f_ϕ is a 1-Lipschitz function
  - PPO-GAN [Wu et al., 2020],when $\mathbb{D}$ is KL divergence

** Proofs adapted from Farnia & Tse 2018: "A Convex Duality Framework for GANs"*

Quick recap: well-known algorithms/paradigms recovered by SE

Experience type	Experience function f	Divergence $\mathbb{D}$	α	β	Algorithm
Data instances	$f_{\text{data}}(\boldsymbol{x}; \mathcal{D})$	CE	1	1	Unsupervised MLE
	$f_{\text{data}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Supervised MLE
	$f_{\text{data-self}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Self-supervised MLE
	$f_{\text{data-w}}(\boldsymbol{t}; \mathcal{D})$	CE	1	ϵ	Data Re-weighting
	$f_{\text{data-aug}}(\boldsymbol{t}; \mathcal{D})$	CE	1	ϵ	Data Augmentation
	$f_{\text{active}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Active Learning (Ertekin et al., 2007)
Knowledge	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	Posterior Regularization (Ganchev et al., 2010)
	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	CE	$\mathbb{R}$	1	Unified EM (Samdani et al., 2012)
Reward	$\log Q^{\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	Policy Gradient
	$\log Q^{\theta}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in,\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	1	1	+ Intrinsic Reward
	$Q^{\theta}(\boldsymbol{x}, \boldsymbol{y})$	CE	$\rho > 0$	$\rho > 0$	RL as Inference
Model	$f_{\text{model}}^{\text{mimicking}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	CE	1	ϵ	Knowledge Distillation (G. Hinton et al., 2015)
Variational	binary classifier	JSD	0	1	Vanilla GAN (Goodfellow et al., 2014)
	discriminator	f-divergence	0	1	f-GAN (Nowozin et al., 2016)
	1-Lipschitz discriminator	W_1 distance	0	1	WGAN (Arjovsky et al., 2017)
	1-Lipschitz discriminator	KL	0	1	PPO-GAN (Y. Wu et al., 2020)
Online	$f_{\tau}(\boldsymbol{t})$	CE	$\rho > 0$	$\rho > 0$	Multiplicative Weights (Freund & Schapire, 1997)

Paradigms not (yet) covered by SE:

- Meta learning
- Lifelong learning
- …

Interesting future work to study the connections

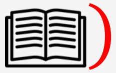
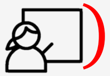
Why this is useful?

- Panoramic Learning: learning with ALL experience
  - Experience composition
  - Reuse specialized algorithms -- one runway for different aircrafts
- Complex interaction between experience
- Multi-agent game theoretic learning using all experience

Panoramic Learning: *experience composition*

- Distinct types of experience are all formulated with $f(\boldsymbol{x}, \boldsymbol{y})$
- Combine and plug different f functions into SE to drive learning

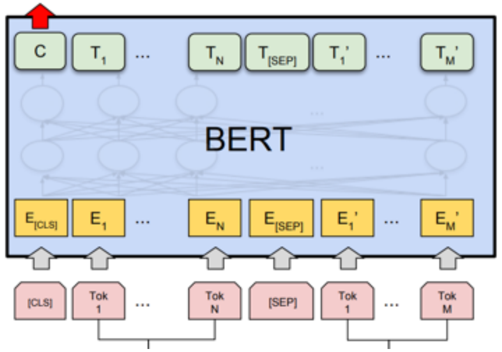
$$SE(f, \mathbb{D}, \alpha, \beta)$$

$f = w_1 \cdot f(\boldsymbol{x} \,|$  $) + w_2 \cdot f(\boldsymbol{x} \,|$  $) + w_3 \cdot f(\boldsymbol{x} \,|$  $) + w_4 \cdot f(\boldsymbol{x} \,|$  $) + \dots$

Focus on **what** to use, instead of worrying about **how** to use

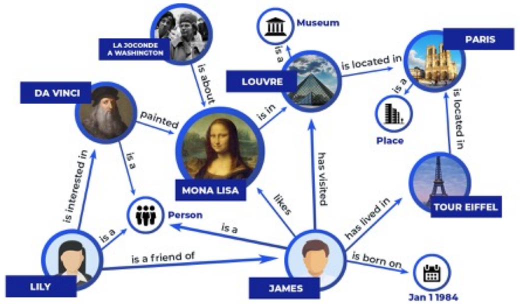
Panoramic Learning: *experience composition*

Ex.1: Using symbolic knowledge to learn neural networks

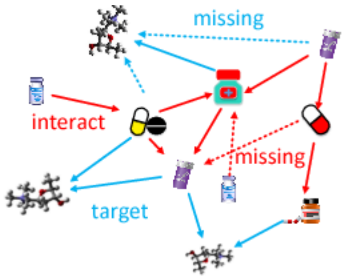


Neural model

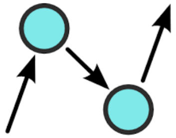
Taxt samples



Knowledge bases



Medical KG



ConceptNet (CN)

$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\ p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right]$$

Hu et al., ACL 2016, "Harnessing Deep Neural Networks with Logic Rules"
Hu et al., NeurIPS 2020, "Deep Generative Models with Learnable Knowledge Constraints"
Tan et al., EMNLP 2020, "Summarizing Text on Any Aspects: A Knowledge-Informed Weakly-Supervised Approach"

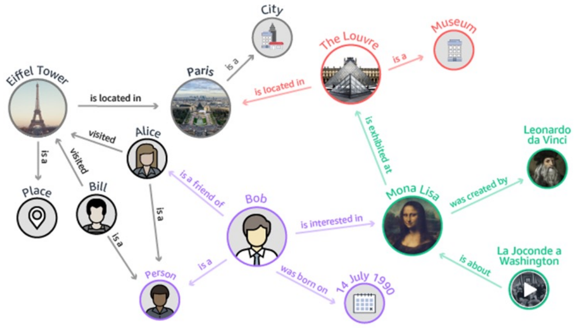
Panoramic Learning: *experience composition*

Ex.2: Using neural networks to "learn" symbolic knowledge

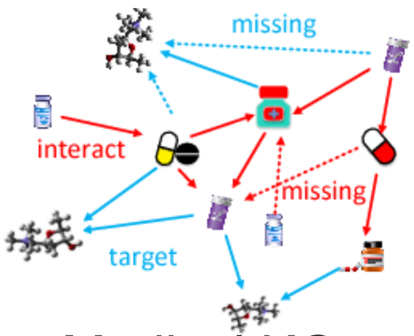
$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\Big(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\Big) - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\Big[f(\boldsymbol{x},\boldsymbol{y})\Big]$$

- θ: graph structure to be learned
- p_θ: a simulation model generating medical task samples $(\boldsymbol{x},\boldsymbol{y})$ based on the knowledge graph θ

Measuring likelihood of sample $(\boldsymbol{x},\boldsymbol{y})$ under a trained medical neural model



Commonsense graph



Medical KG

Hao, Tan et al., 2022, "BertNet: Harvesting Knowledge Graphs from Pretrained Language Models"

Panoramic Learning: *experience composition*

Ex.2: Using neural networks to "learn" symbolic knowledge

$$\min_{q,\theta} -\alpha \mathbb{H}(q) + \beta \mathbb{D}\left(q(\boldsymbol{x}, \boldsymbol{y}), p_\theta(\boldsymbol{x}, \boldsymbol{y})\right) - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x}, \boldsymbol{y}) \right]$$

Head entity	Relation	Tail entity	Head entity	Relation	Tail entity
exercise	*prevent*	obesity	students	*worth celebrating*	graduate
apple	*business*	Mac	newborn	*can but not good at*	sit
sleep	*prevent*	illness	social worker	*can help*	foster child
mall	*place for*	shopping	honey	*ingredient for*	honey cake
gym	*place for*	sweat	cabbage	*ingredient for*	cabbage salad
wheat	*source of*	flour	China	*separated by the ocean*	Japan
oil	*source of*	fuel	Africa	*separated by the ocean*	Europe

Figure 4: Examples of knowledge tuples harvested from ROBERTA-LARGE with MULTI-PROMPTS.

Hao, Tan et al., 2022, "BertNet: Harvesting Knowledge Graphs from Pretrained Language Models"

Panoramic Learning: *experience composition*

Ex.2: Using neural networks to “learn” symbolic knowledge

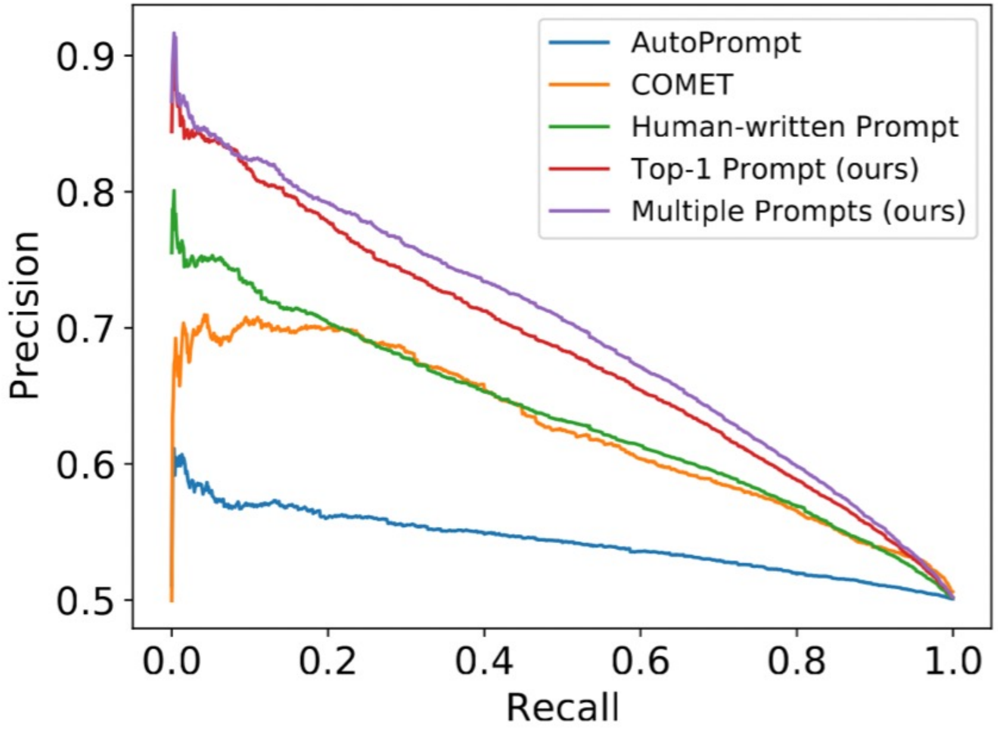


Figure 2: Precision-recall curve on ConceptNet relations.

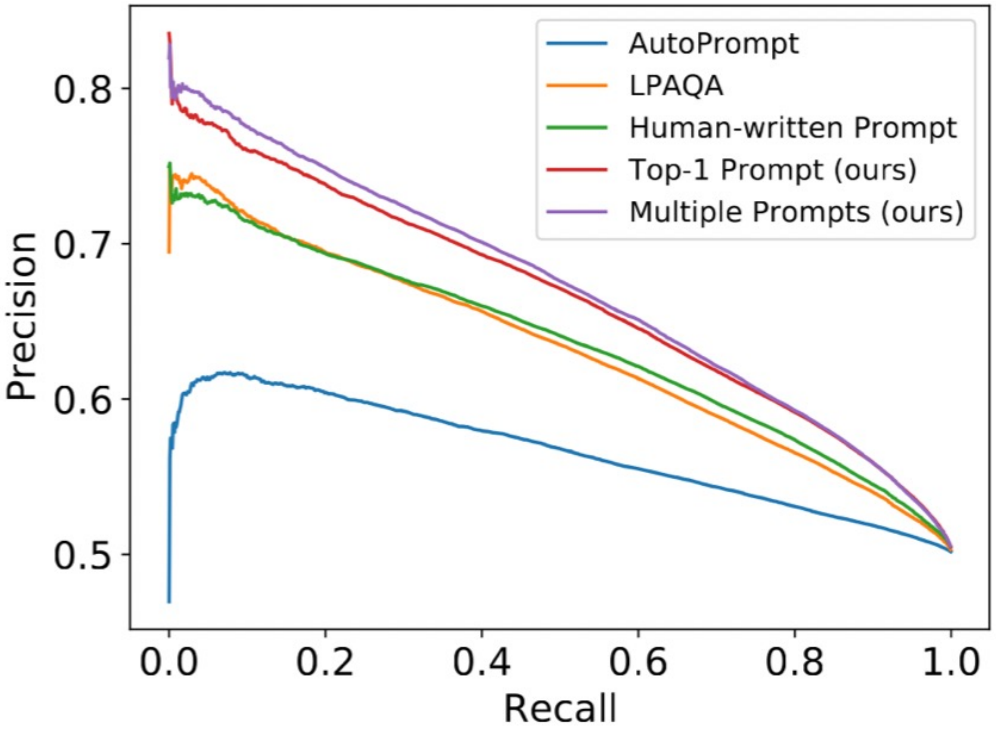


Figure 3: Precision-recall curve on LAMA relations.

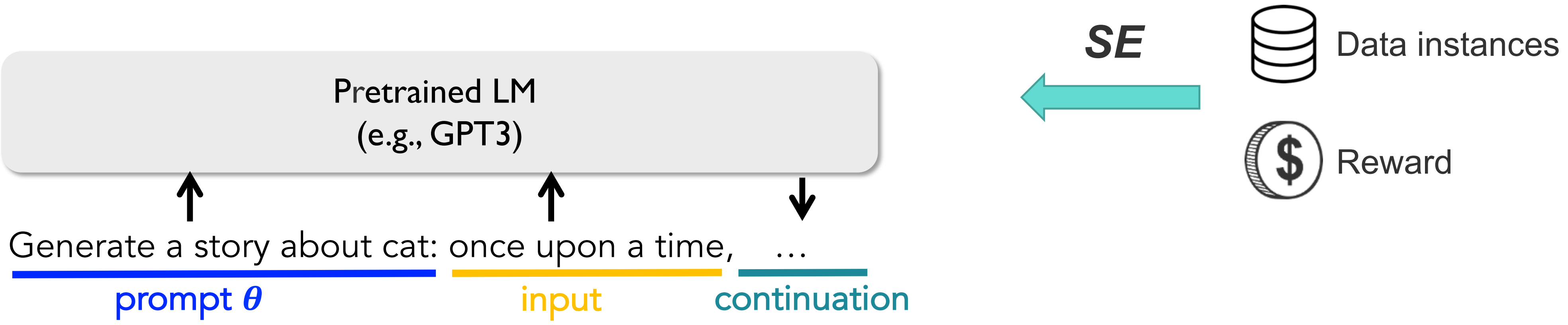
Hao, Tan et al., 2022, “BertNet: Harvesting Knowledge Graphs from Pretrained Language Models”

Panoramic Learning: *experience composition*

Ex.3: Learning prompts to control large pretrained models

$$\min_{q,\theta} - \alpha \mathbb{H}(q) + \beta \mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y}) \right]$$

Experiences *f*



Deng, Wang, Hsieh et al., EMNLP 2022, " RLPrompt: Optimizing Discrete Text Prompts With Reinforcement Learning"

Panoramic Learning: *experience composition*

Ex.4: Learning controllable text generation – more in Lecture#2

- Combine and plug different f functions into SE to drive learning

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \alpha\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \beta\mathbb{H}(q)$$

$$f = w_1 \cdot f_{data} + w_2 \cdot f_{rules} + w_3 \cdot f_{reward} + \cdots$$

- Enable applications for controllable content generation

Controllable text generation

f = sentiment classifier
+ linguistic rules
+ language model

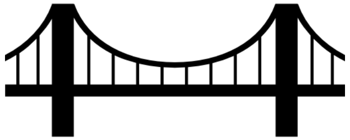
Controlling sentiment

Pos	*The film is full of imagination!*
Neg	*The film is strictly routine!*

[Hu et al., 2017; Yang et al., 2018]

Panoramic Learning: *reusing algorithms*

- Unifying perspective of diverse paradigms (each tailored for a specific type of experience) under SE



- Combining or integrating different experiences
- Re-use or repurpose originally specialized algorithms
    - Systematic idea transfer and solution exchange
    - Solving challenges in one paradigm by applying well-known solutions from another
    - Accelerate innovations across research areas

Panoramic Learning: *reusing algorithms – Ex.1*

- Rules in PR ⇔ Reward in RL
- Empower **reward learning** algo. to **learning rules** [Hu et al., 2018]

Algorithm	f	α	β	$\mathbb{D}$
Unsupervised MLE	$f(\boldsymbol{x}; \mathcal{D})$	1	1	CE
Supervised MLE	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	1	ϵ	CE
Active Learn.	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}) + u(\boldsymbol{x})$	temp., > 0	ϵ	CE
Reward-augment MLE	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	ϵ	CE
PG for Seq. Gen.	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	1	CE
Posterior Reg.	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, > 0	α	CE
Unified EM	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, $\in \mathbb{R}$	1	CE
Policy Gradient (PG)	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
+ Intrinsic Reward	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
RL as inference	$Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	temp., > 0	α	CE
Vanilla GAN	binary classifier	0	1	JSD
f-GAN	discriminator	0	1	f-divg.
WGAN	1-Lipschitz discriminator	0	1	W dist.

Panoramic Learning: *reusing algorithms – Ex.1*

- Rules in PR ⇔ Reward in RL
- Empower **reward learning** algo. to **learning rules** [Hu et al., 2018]

$$\min_{q,\theta} - \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y}) \right] + \beta \mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y}) \right) - \alpha \mathbb{H}(q)$$

MaxEnt inverse RL [Ziebart'08]:

- Parameterize reward $f_\phi(\boldsymbol{x},\boldsymbol{y})$ with ϕ
- Learn ϕ with the additional optimization step:

$$\min_{\phi} - \mathbb{E}_{(\boldsymbol{x}^*,\boldsymbol{y}^*)\sim\mathcal{D}}\left[\log q_\phi(\boldsymbol{x}^*,\boldsymbol{y}^*) \right]$$

Note: q is a function of f_ϕ, thus q depends on ϕ

Reuse to learn parameterized rules

PR with learnable rule constraints $f_\phi(\boldsymbol{x},\boldsymbol{y})$:

- E-step to get closed-form q_ϕ
- M-step to update p_θ
- Reused reward-learning step to update ϕ

Panoramic Learning: *reusing algorithms – Ex.2*

- Data in supervised MLE ⇔ Reward in RL
- Empower **reward learning** algo. to **learning data augmentation** [Hu et al., 2019]

Algorithm	f	α	β	$\mathbb{D}$
Unsupervised MLE	$f(\boldsymbol{x}; \mathcal{D})$	1	1	CE
Supervised MLE	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	1	ϵ	CE
Active Learn.	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}) + u(\boldsymbol{x})$	temp., > 0	ϵ	CE
Reward-augment MLE	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	ϵ	CE
PG for Seq. Gen.	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	1	CE
Posterior Reg.	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, > 0	α	CE
Unified EM	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, $\in \mathbb{R}$	1	CE
Policy Gradient (PG)	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
+ Intrinsic Reward	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
RL as inference	$Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	temp., > 0	α	CE
Vanilla GAN	binary classifier	0	1	JSD
f-GAN	discriminator	0	1	f-divg.
WGAN	1-Lipschitz discriminator	0	1	W dist.

Panoramic Learning: *reusing algorithms – Ex.2*

- Data in supervised MLE ⇔ Reward in RL
- Empower **reward learning** algo. to **learning data augmentation** [Hu et al., 2019]

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \alpha\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}), p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \beta\mathbb{H}(q)$$

Intrinsic reward learning [Zheng et al.,08]:

- Reward $f_\phi = f^{ex} + f_\phi^{in}$
- I.e., parameterize (intrinsic) reward f_ϕ^{in} with ϕ
- Learn ϕ with the additional optimization step:

$$\min_{\phi} \mathcal{L}_{SE}(\theta^t(\phi))$$

Same form as standard equation

Note: updates of θ depend on experience f_ϕ, thus the resulting θ^t is a function of ϕ

Reuse to learn parameterized data augmentation model

MLE with learnable data augmentation $f_\phi(\boldsymbol{x},\boldsymbol{y})$:

- E-step to get closed-form q_ϕ
- M-step to update p_θ
- Reused reward-learning step to update ϕ

Petuum

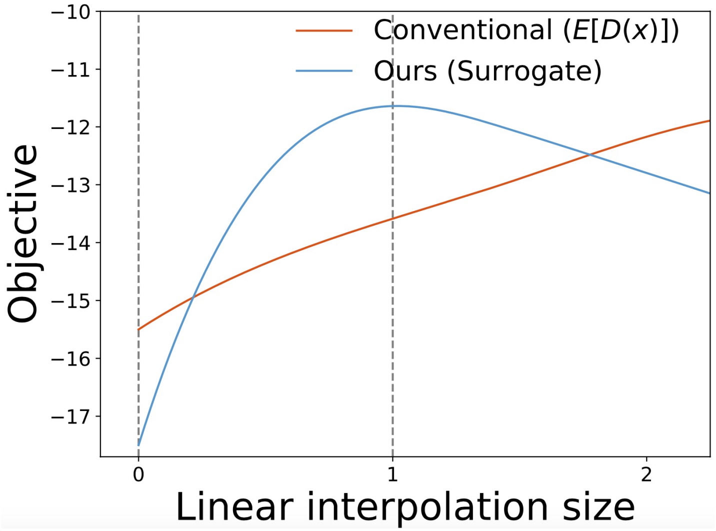
Panoramic Learning: *reusing algorithms – Ex.3*

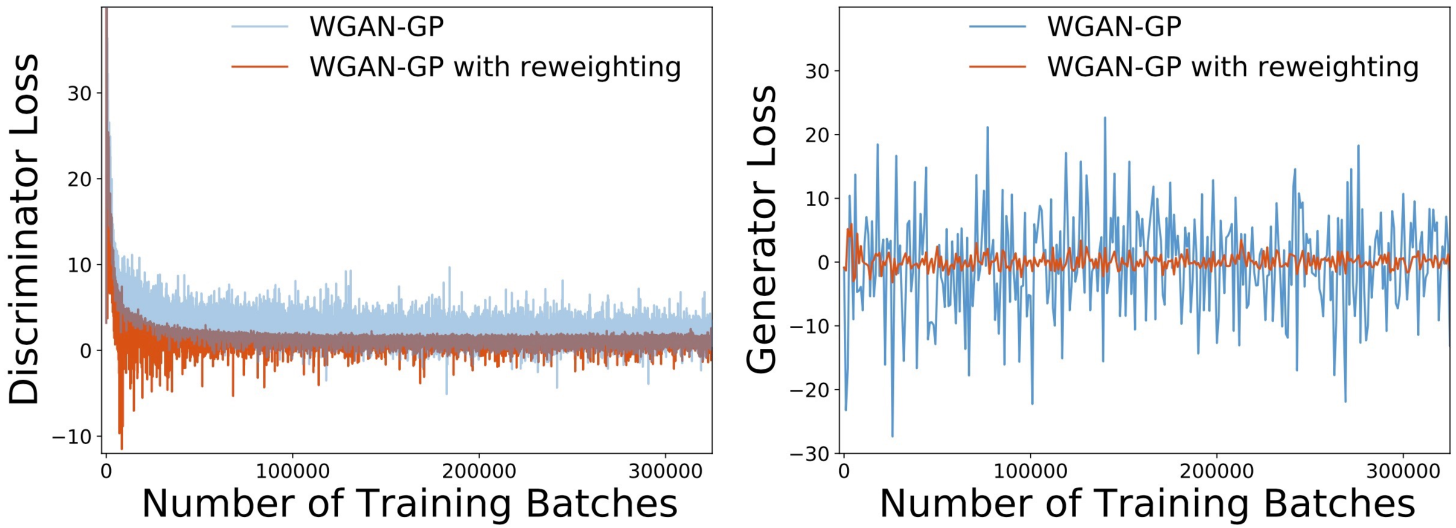
- GANs ⇔ RL ⇔ VI
- Empower **RL/VI** algo. (e.g., PPO) to **stabilize GAN training** [Wu et al., 2020]

Algorithm	f	α	β	$\mathbb{D}$
Unsupervised MLE	$f(\boldsymbol{x}; \mathcal{D})$	1	1	CE
Supervised MLE	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D})$	1	ϵ	CE
Active Learn.	$f(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}) + u(\boldsymbol{x})$	temp., > 0	ϵ	CE
Reward-augment MLE	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	ϵ	CE
PG for Seq. Gen.	$f_{\text{metric}}(\boldsymbol{x}, \boldsymbol{y}; \mathcal{D}, r)$	1	1	CE
Posterior Reg.	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, > 0	α	CE
Unified EM	$f_{rule}(\boldsymbol{x}, \boldsymbol{y})$	weight, $\in \mathbb{R}$	1	CE
Policy Gradient (PG)	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
+ Intrinsic Reward	$\log Q^{ex}(\boldsymbol{x}, \boldsymbol{y}) + Q^{in}(\boldsymbol{x}, \boldsymbol{y})$	1	1	CE
RL as inference	$Q^{ex}(\boldsymbol{x}, \boldsymbol{y})$	temp., > 0	α	CE
Vanilla GAN	binary classifier	0	1	JSD
f-GAN	discriminator	0	1	f-divg.
WGAN	1-Lipschitz discriminator	0	1	W dist.

Panoramic Learning: *reusing algorithms – Ex.3*

- GANs ⇔ RL ⇔ VI
- Empower **RL/VI** algo. (e.g., PPO) to **stabilize GAN training** [Wu et al., 2020]





(a) Re-use PPO objective for GAN training: discourage excessively large updates by "trapping" the update size around 1

(b) Re-use importance weighting in a VI perspective: greatly reduced variance in both generator and discriminator losses

Improved performance on a range of problems, including image generation, text generation, and text style transfer

Summary so far …

- The standard equation of objective

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right] + \beta\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\, p_\theta(\boldsymbol{x},\boldsymbol{y})\right) - \alpha\mathbb{H}(q)$$

- Experience function f can encode different types of experience
  - Data instances, constraints, informativeness, reward, adversary models, …
- Enable panoramic learning with ALL experience
  - Re-use or repurpose originally specialized algorithms to other contexts
  - Experience compositonality

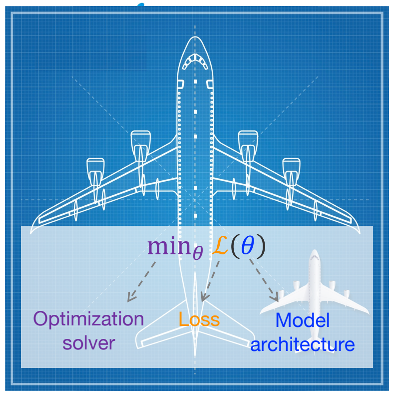
Toward A "Standard Model" of ML

- Loss
- Experience
- Optimization solver
- Model architecture

$$\min_{\theta} \mathcal{L}(\theta, \mathcal{E})$$

Optimization solver | Loss | Model architecture | Experience

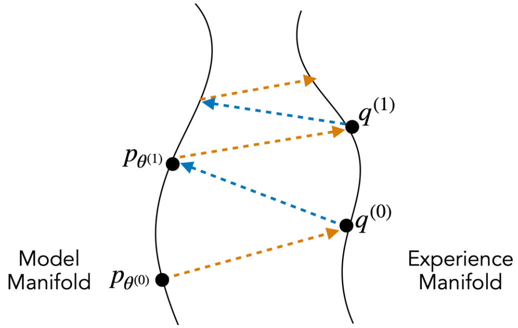
The zoo of optimization solvers



$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\left(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \right) - \mathbb{E}_{q(\boldsymbol{x})}\left[f(\boldsymbol{x}) \right]$$

Optimization of the loss, subject to $q \in \mathcal{P}_{\text{prob}}$.
Convex to q when $\alpha, \beta > 0$ and $\mathbb{D}$ is convex

- Like the Standard Equation as a *master objective* for many paradigms, is there a *master solver* for optimization of loss?
- No (yet) such a general algorithm
- Alternating Projection:
  - Most widely used
  - EM, Variational EM (Variational inference), Wake-Sleep, …



The Teacher-Student Mechanism

$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\left(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \right) - \mathbb{E}_{q(\boldsymbol{x})}\left[f(\boldsymbol{x}; .)\right]$$

when $\alpha, \beta > 0$ and $\mathbb{D} = \mathrm{CE}$

(1) Teacher step:

$$q(\boldsymbol{x}) = \exp\left\{ \frac{\beta \log p_\theta(\boldsymbol{x}) + f(\boldsymbol{x};\ .)}{\alpha} \right\} / Z$$

(2) Student step:

$$\min_{\theta} \mathbb{E}_{q(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}) \right]$$

Generalization of the classic Variational EM

- Generalized *E-step*
  Support all types of experience
- *M-step*

87

The Teacher-Student Mechanism

$$\min_{q,\theta} -\alpha \mathbb{H}(q) + \beta \mathbb{D}\left(q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \right) - \mathbb{E}_{q(\boldsymbol{x})}\left[f(\boldsymbol{x}; .) \right]$$

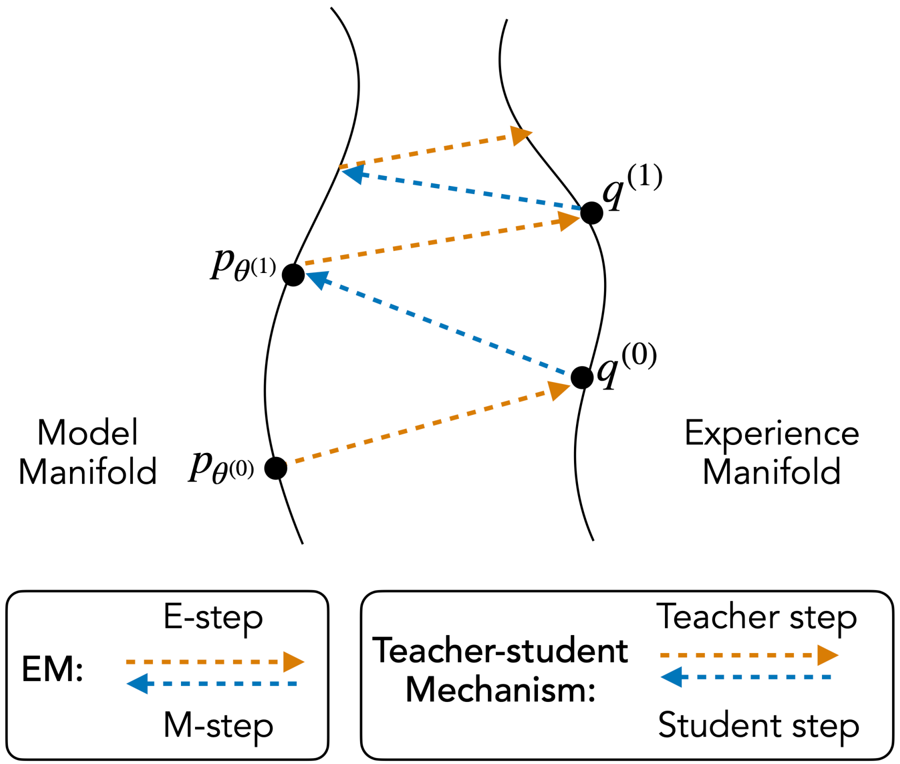
when $\alpha, \beta > 0$ and $\mathbb{D} = \mathrm{CE}$

(1) Teacher step:

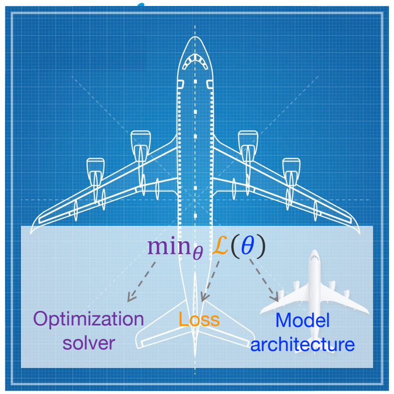
$$q(\boldsymbol{x}) = \exp\left\{ \frac{\beta \log p_\theta(\boldsymbol{x}) + f(\boldsymbol{x};\ .)}{\alpha} \right\} / Z$$

(2) Student step:

$$\min_{\theta} \mathbb{E}_{q(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}) \right]$$



Some "advanced" (specialized) techniques



$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\Big(\ q(\boldsymbol{x}), p_\theta(\boldsymbol{x})\ \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[\ f(\boldsymbol{x})\ \Big]$$

- Alternating Projection:
  - EM, Variational EM (Variational inference), Wake-Sleep, …
  - SGD, Back-propagation (BP)
- Convex duality, Lagrangian -- Kernel Tricks
- Integer linear programming (ILP)
- Probability functional descent (PFD) [Chu et al., 2019] -- Influence function, gives a neat formulation of GAN-like optimization and a few others

I: Duality

- Structured MaxEnt Discrimination (SMED) [Zhu and Xing, 2013]:

$$\min_{q,\,\xi\geq 0} \; -\alpha H(q) - \beta \mathbb{E}_q\left[\log p(\boldsymbol{\theta}) \right] + U(\boldsymbol{\xi})$$

$$s.t. \; -\mathbb{E}_q[\,\Delta F_i(\boldsymbol{y};\boldsymbol{\theta}) - \Delta \ell_i(\boldsymbol{y})\,] \leq \xi_i \quad \forall i$$

  - Solve the (primal) Lagrangian:

$$q(\boldsymbol{\theta}) = \exp\left\{ \frac{\beta \log p(\boldsymbol{\theta}) + \sum_{i,\boldsymbol{y}\neq \boldsymbol{y}_i^*} \lambda_i(\boldsymbol{y})(\Delta F_i(\boldsymbol{y};\boldsymbol{\theta}) - \Delta \ell_i(\boldsymbol{y}))}{\alpha} \right\} / Z(\boldsymbol{\lambda})$$

  - Solve Lagrangian multipliers $\boldsymbol{\lambda}$ from the dual problem (when $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|0,I)$; $U(\xi) = \sum \xi_i$)

$$\max_{\lambda\geq 0,\,\sum \lambda_i = 1} \sum_{i,\boldsymbol{y}\neq \boldsymbol{y}_i^*} \lambda_i(\boldsymbol{y}) \Delta \ell_i(\boldsymbol{y}) - \frac{1}{2}\left| \sum_{i,\boldsymbol{y}\neq \boldsymbol{y}_i^*} \lambda_i(\boldsymbol{y}) \Delta T_i(\boldsymbol{y}) \right|^2$$

Allows kernel trick for nonlinear interactions b/w experiences

II: Influence Function and Probability Functional Descent

- Gradient descent in the space of probability measures $\mathcal{P}(X)$

$$\min_{p \in \mathcal{P}(X)} \mathcal{J}(p)$$

$\mathcal{J}: \mathcal{P}(X) \to \mathbb{R}$: a probability functional

- Influence function $\Psi_p(x)$:

$$d\mathcal{J}_p(\chi) = \int_X \Psi_p(x)\chi(dx)$$
$$= \mathbb{E}_q[\Psi_p(x)] - \mathbb{E}_p[\Psi_p(x)]$$

Gateaux differential of $\mathcal{J}$ at p in the direction $\chi = q - p$

- With a linear approximation $\tilde{\mathcal{J}}(p)$ to $\mathcal{J}(p)$ around p_0:

$$\tilde{\mathcal{J}}(p) = \mathcal{J}(p_0) + d\mathcal{J}_{p_t}(p - p_0).$$
$$= \mathbb{E}_{x \sim p}\left[\, \Psi_{p_0}(x) \,\right] + const.$$

- Thus, once we obtain the influence function, we can optimize p by decreasing $\mathbb{E}_{x \sim p}\left[\, \Psi_{p_0}(x) \,\right]$

[Chu et al., 2019]

Adversarial learning using PFD

$$\mathcal{J}(p_\theta) = \mathbb{D}\left(\; p_d(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \; \right)$$

- Often no closed-form influence function, e.g., when $\mathbb{D}$ is JSD or W-distance
- Approximate with convex duality:
    - Convex conjugate $\mathcal{J}^*(\psi) = \sup_u \int_x \psi(\boldsymbol{x}) u(dx) - \mathcal{J}(u)$
    - Influence function is obtained via $\Psi_{p_\theta}(x) = \text{argmax}_\psi \, \mathbb{E}_{\boldsymbol{x} \sim p_\theta}[\psi(\boldsymbol{x})] - \mathcal{J}^*(\psi)$
    - Parameterize ψ as below to recover optimization of generator and discriminator

$$\psi_{\boldsymbol{\phi}}(\boldsymbol{x}) \coloneqq 0.5 \log\left(1 - C_\phi\right) - 0.5 \log 2$$

$$\Psi_{JS} = \text{argmax}_\phi \, \mathbb{E}_{p_{data}}\left[\log C_\phi\right] - \mathbb{E}_{p_\theta}\left[\log\left(1 - C_\phi\right)\right]$$

- The whole optimization of $\mathcal{J}(p)$ is thus

$$\min_\theta \max_\phi \mathbb{E}_{p_{data}}\left[\log C_\phi\right] - \mathbb{E}_{p_\theta}\left[\log\left(1 - C_\phi\right)\right]$$

[Chu et al., 2019]

Other popular algorithms in the PFD view

- PFD recovers optimization procedures in some popular algorithms

Algorithm	Type of derivative estimator
Generative adversarial networks	
Minimax GAN (Goodfellow et al., 2014)	Convex duality
Non-saturating GAN (Goodfellow et al., 2014)	Binary classification
Wasserstein GAN (Arjovsky et al., 2017)	Convex duality
Variational inference	
Black-box variational inference (Ranganath et al., 2014)	Exact
Adversarial variational Bayes (Mescheder et al., 2017)	Binary classification
Adversarial posterior distillation (Wang et al., 2018)	Convex duality
Reinforcement learning	
Policy iteration (Howard, 1960)	Exact
Policy gradient (Williams, 1992)	Monte Carlo
Actor-critic (Konda & Tsitsiklis, 2000; Sutton et al., 2000)	Least squares
Dual actor-critic (Chen & Wang, 2016; Dai et al., 2017b)	Convex duality

Estimation of the influence function

[Chu et al., 2019]

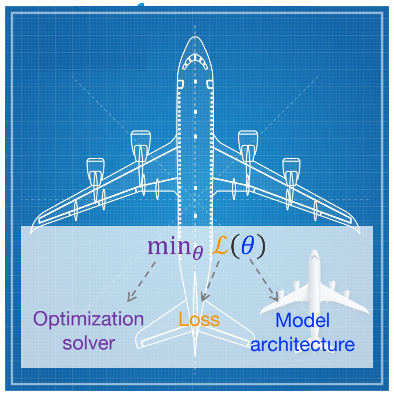
Toward A "Standard Model" of ML

- Loss
- Experience
- Optimization solver
- Model architecture

$$\min_{\theta} \mathcal{L}(\theta, \mathcal{E})$$

Optimization solver | Loss | Model architecture | Experience

Model architecture – *more in Lecture#2*



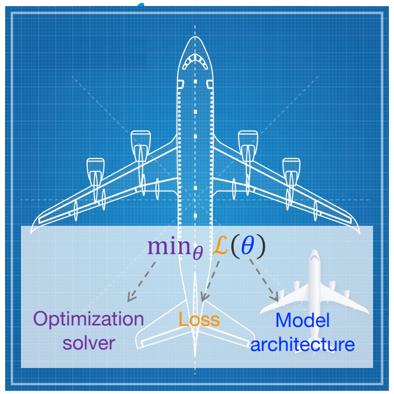
- Relatively well explored:
  - Neural network design
  - Graphical model design
  - Compositional architectures

$$\min_{q,\theta} -\alpha\mathbb{H}(q) + \beta\mathbb{D}\Big(\; q(\boldsymbol{x}), p_\theta(\boldsymbol{x}) \; \Big) - \mathbb{E}_{q(\boldsymbol{x})}\Big[\; f(\boldsymbol{x}) \; \Big]$$

Next lecture: a composable catalog of building blocks

UC San Diego

Summary: A "Standard Model" of ML



- Loss + experience
  - Standard Equation (SE)

$$\min_{q,\theta} -\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[f(\boldsymbol{x},\boldsymbol{y})\right]+\beta\mathbb{D}\left(q(\boldsymbol{x},\boldsymbol{y}),\, p_\theta(\boldsymbol{x},\boldsymbol{y})\right)-\alpha\mathbb{H}(q)$$

- Optimization solver
  - The extended EM algorithm gives a general primal solution in many cases
  - PFD gives a neat formulation for some cases (e.g., GANs)
- Model architecture: vast libraries of building blocks → compositionality

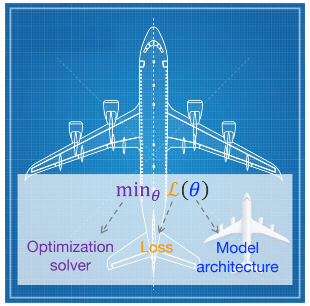
Next: practical implications of the ML "Standard Model"

Schedule

- **Lecture#1:** Theory: The Standard Model of ML

  A blueprint of ML paradigms for ALL experience

  (Jan 19 Thursday, 4:45pm-6:15pm UK Time)



- **Lecture#2:** Tooling: Operationalizing The Standard Model

  Compose your ML solutions like playing Lego

  (Jan 20 Thursday, 1:00pm-2:30pm)

- **Lecture#3:** Computing: Modern infrastructure for productive ML

  Automatic tuning, distributing, and scheduling

  (Jan 20 Thursday, 4:45pm-6:15pm)