On Calibration in Multi-Distribution Learning

Rajeev Verma / University of Amsterdam

UvA - BOSCH
DELTA LAB

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning

$$\frac{1}{N}\sum_{i=1}^{N}\ell\left(h\left(x_i\right),y_i\right)$$

Machine Learning

$$x_i \quad , y_i$$

Machine Learning

$$h\left(x_i\right), y_i$$

Machine Learning

$$\frac{1}{N}\sum_{i=1}^{N}\ell\left(h\left(x_i\right),y_i\right)$$

Machine Learning

$$x_i\ , y_i \sim P$$

Machine Learning

$$\frac{1}{N}\sum_{i=1}^{N}\ell\left(h\left(x_i\right),y_i\right)$$

$$\xrightarrow[\text{LLN}]{} \quad \mathbb{E}_P\left[\ell\left(h\left(X\right),Y\right)\right]$$

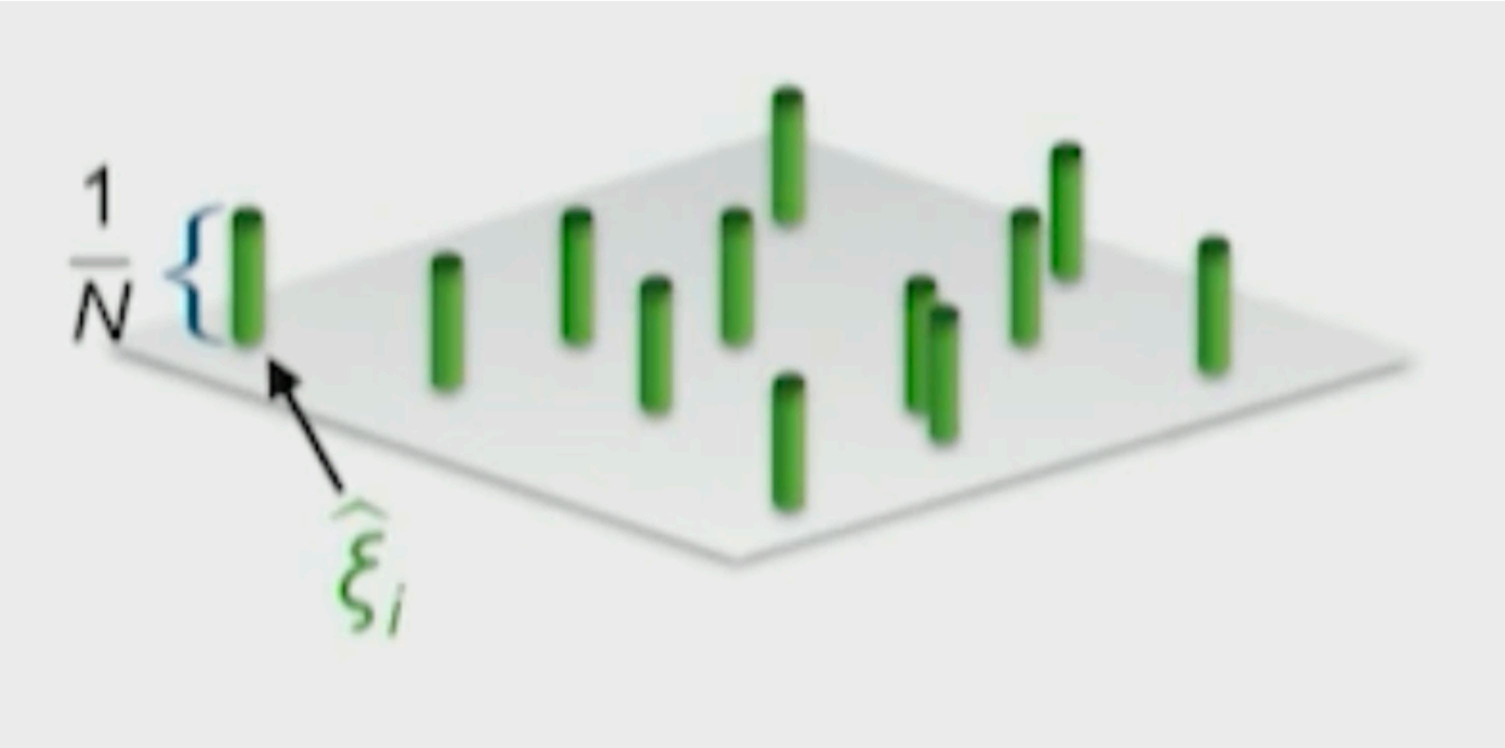
Machine Learning

$$x_i \;, y_i \, \square \sim P$$

$$\mathbb{E}_P\left[\ell\left(h(X), Y\right)\right]$$

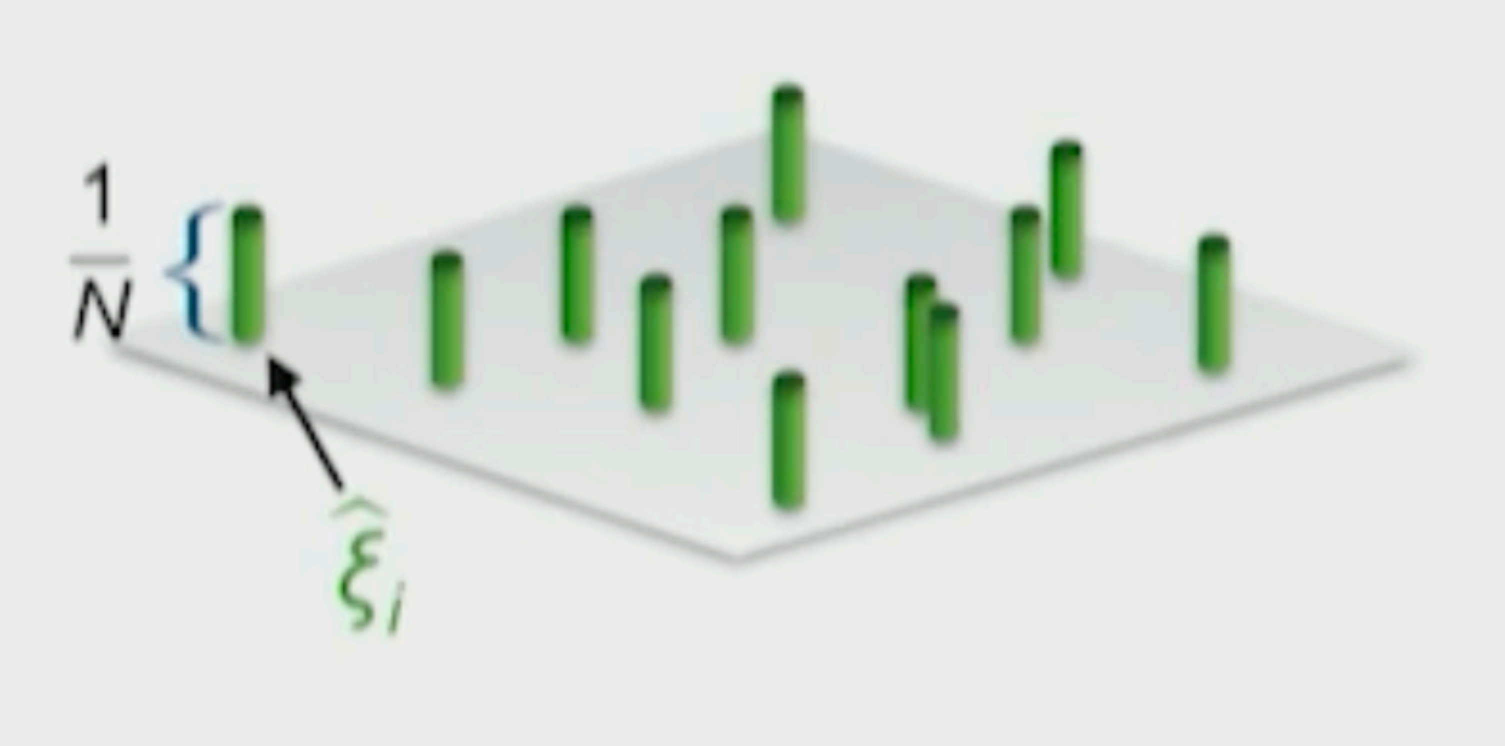
Machine Learning: *distributional robust optimisation*

Machine Learning: *distributional robust optimisation*



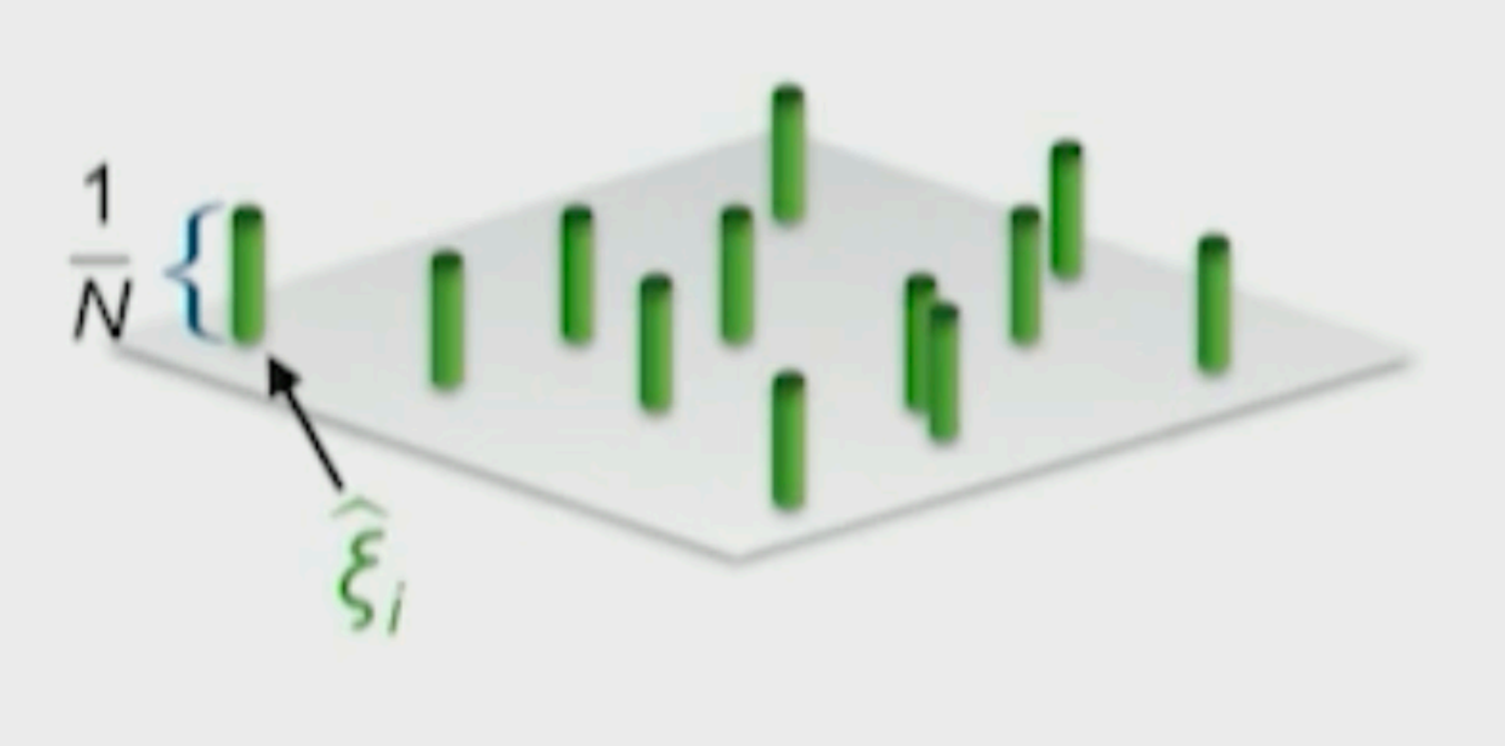
Machine Learning: *distributional robust optimisation*

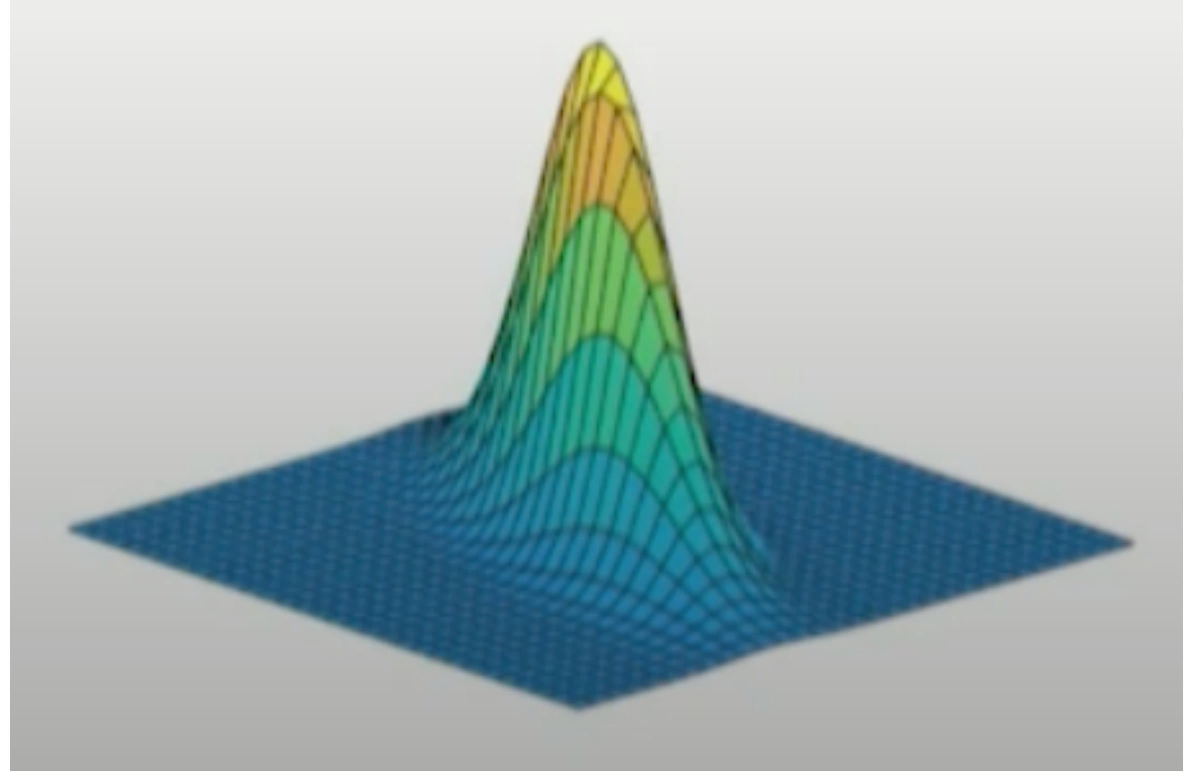
$$\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widehat{\xi}_i}$$



Machine Learning: *distributional robust optimisation*

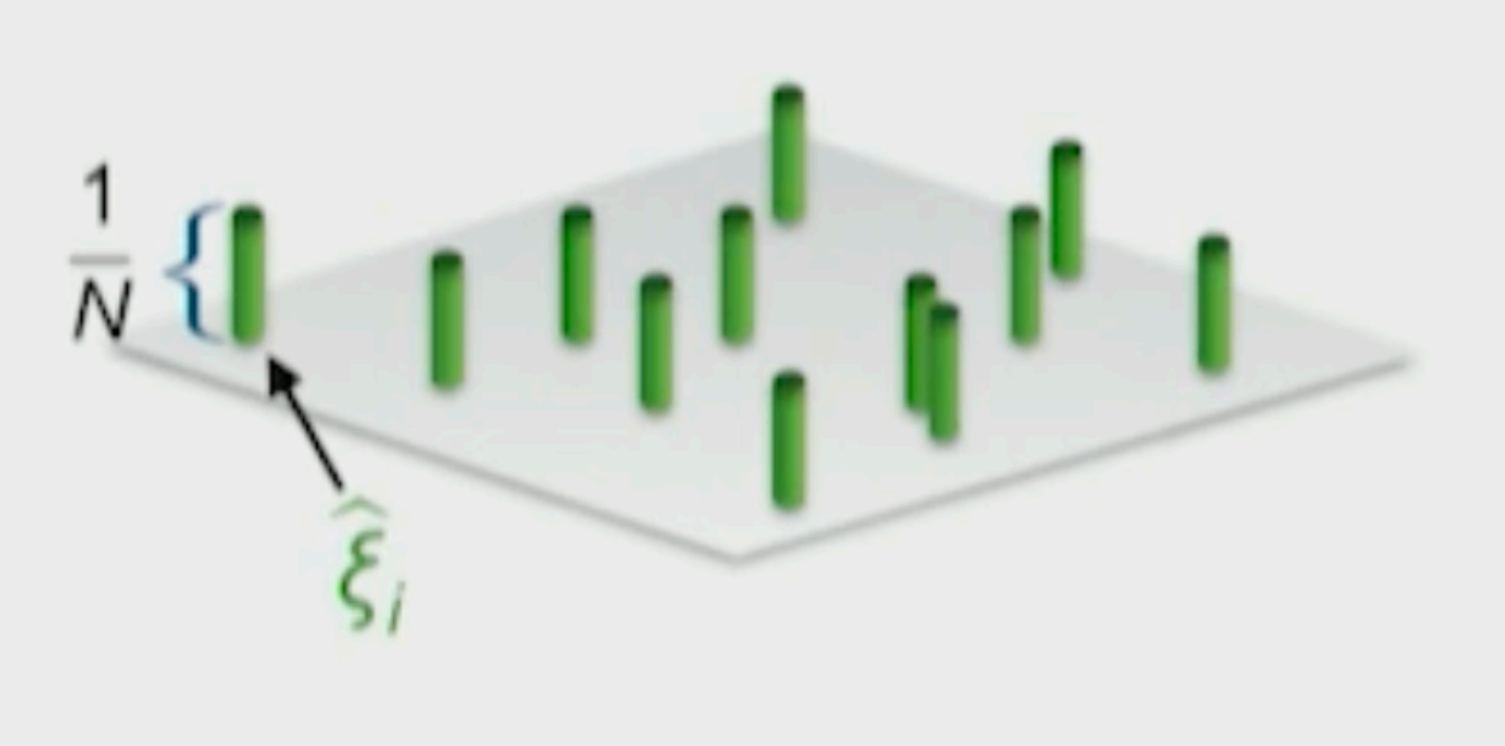
$$\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widehat{\xi}_i}$$

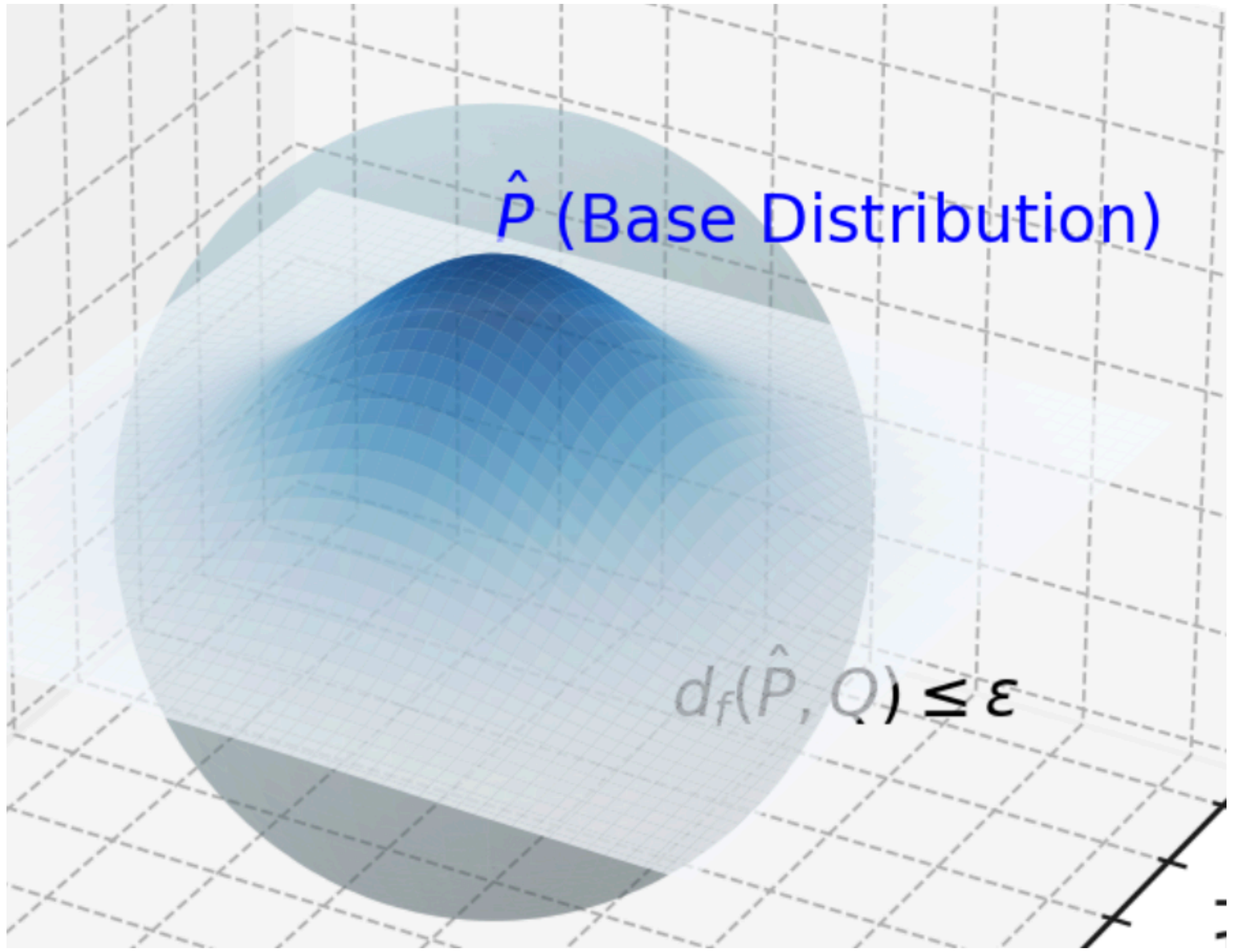




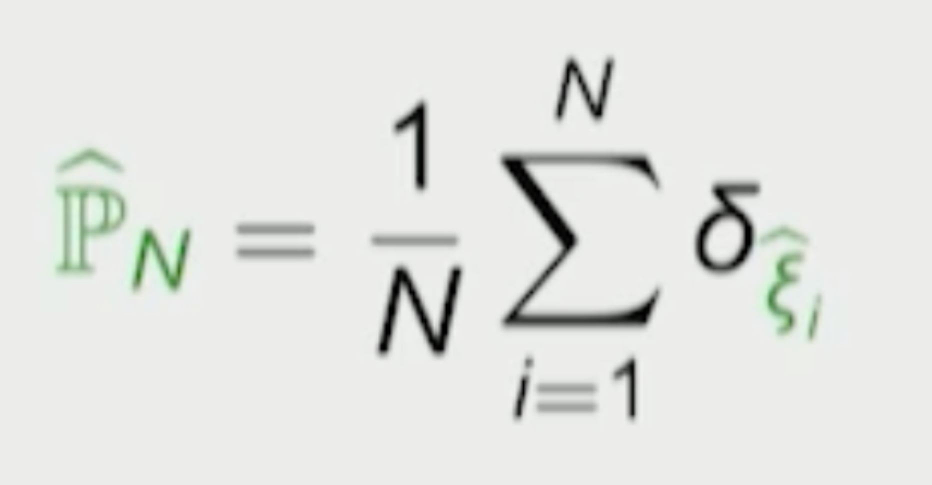
Machine Learning: *distributional robust optimisation*

$$\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widehat{\xi}_i}$$




$\hat{P}$ (Base Distribution)
$d_f(\hat{P}, Q) \leq \varepsilon$

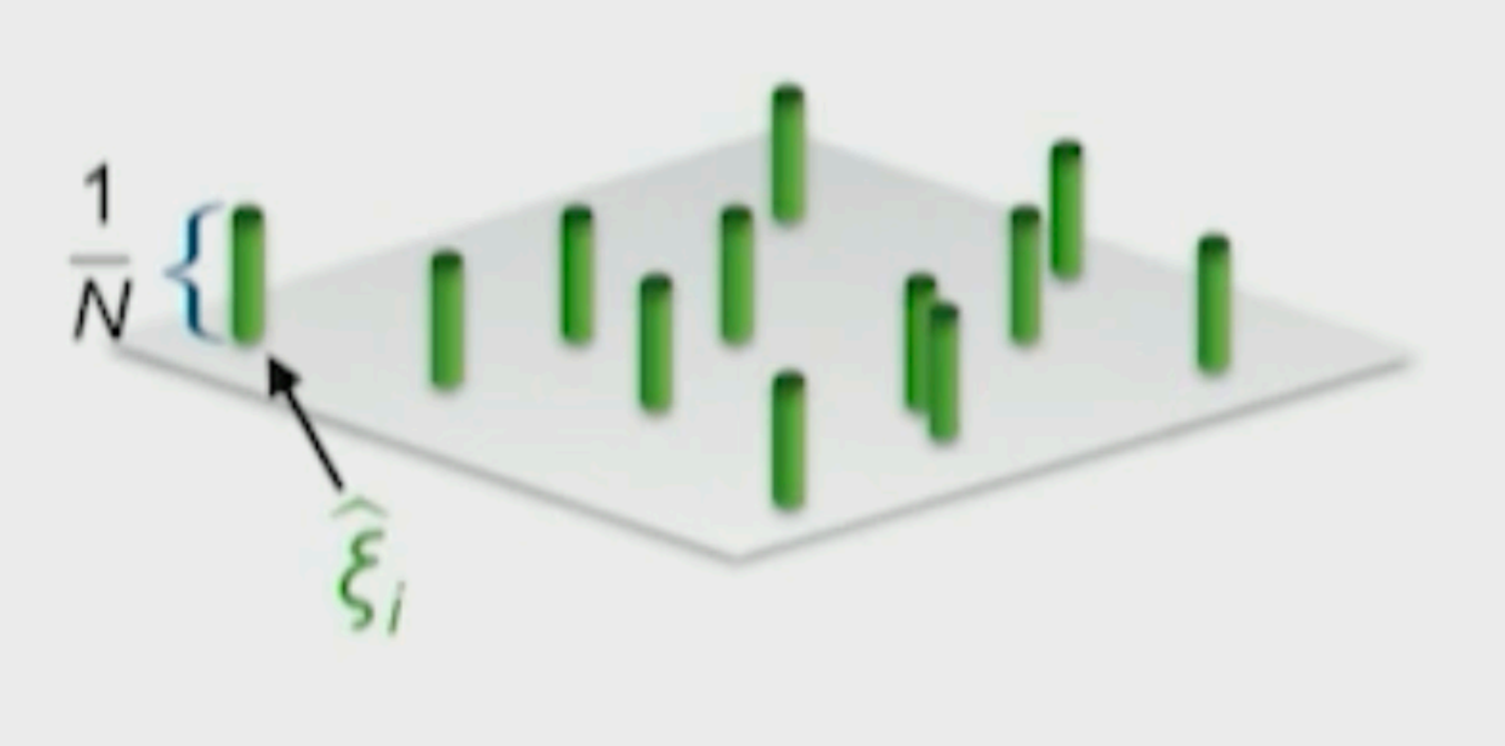
Machine Learning: *distributional robust optimisation*



$$\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widehat{\xi}_i}$$

$$\mathcal{Q} = \{Q \ : \ d_f\left(\hat{P}, Q\right) \leq \epsilon\}$$

ambiguity set of distributions.

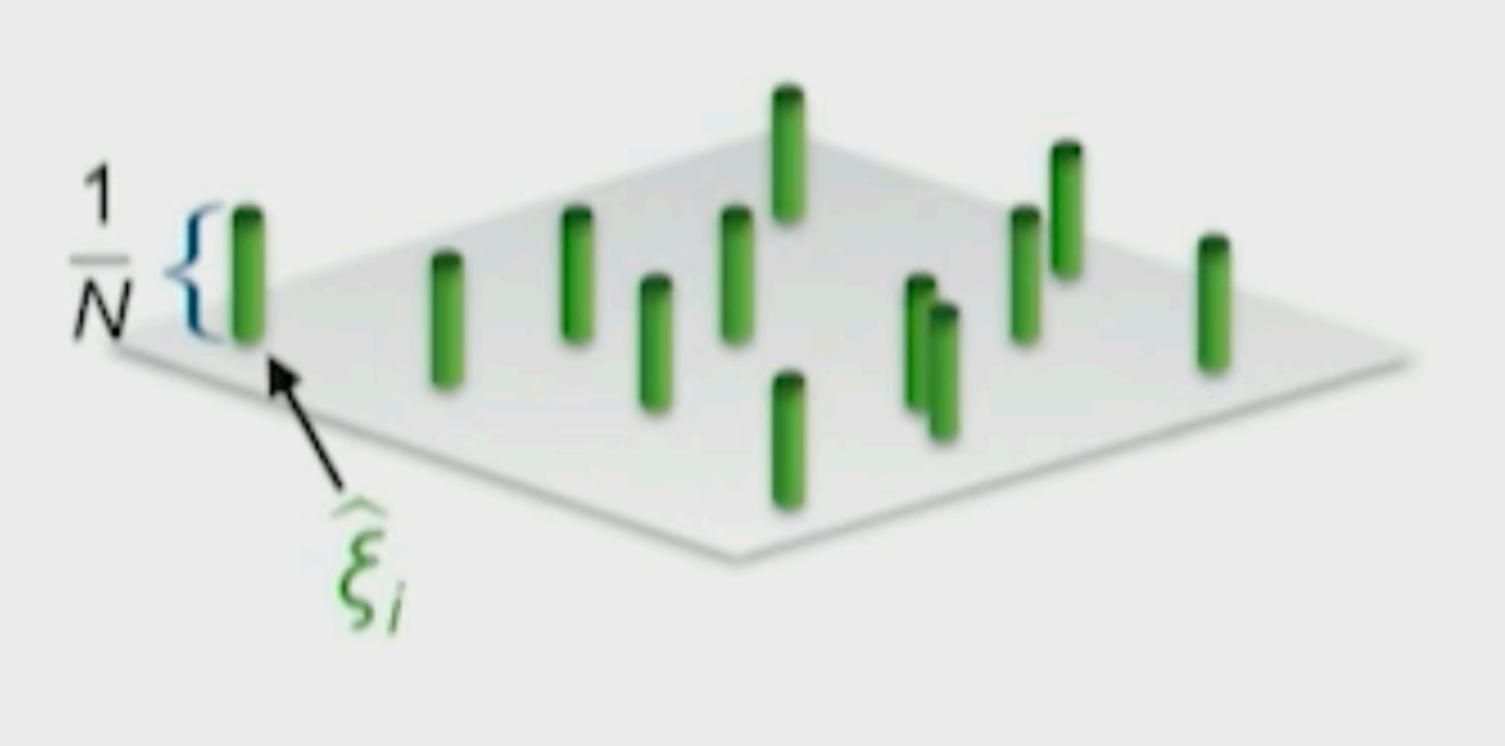


Machine Learning: *distributional robust optimisation*

$$\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widehat{\xi}_i}$$

$$\mathcal{Q} = \{Q \ : \ d_f\left(\hat{P}, Q\right) \le \epsilon\}$$

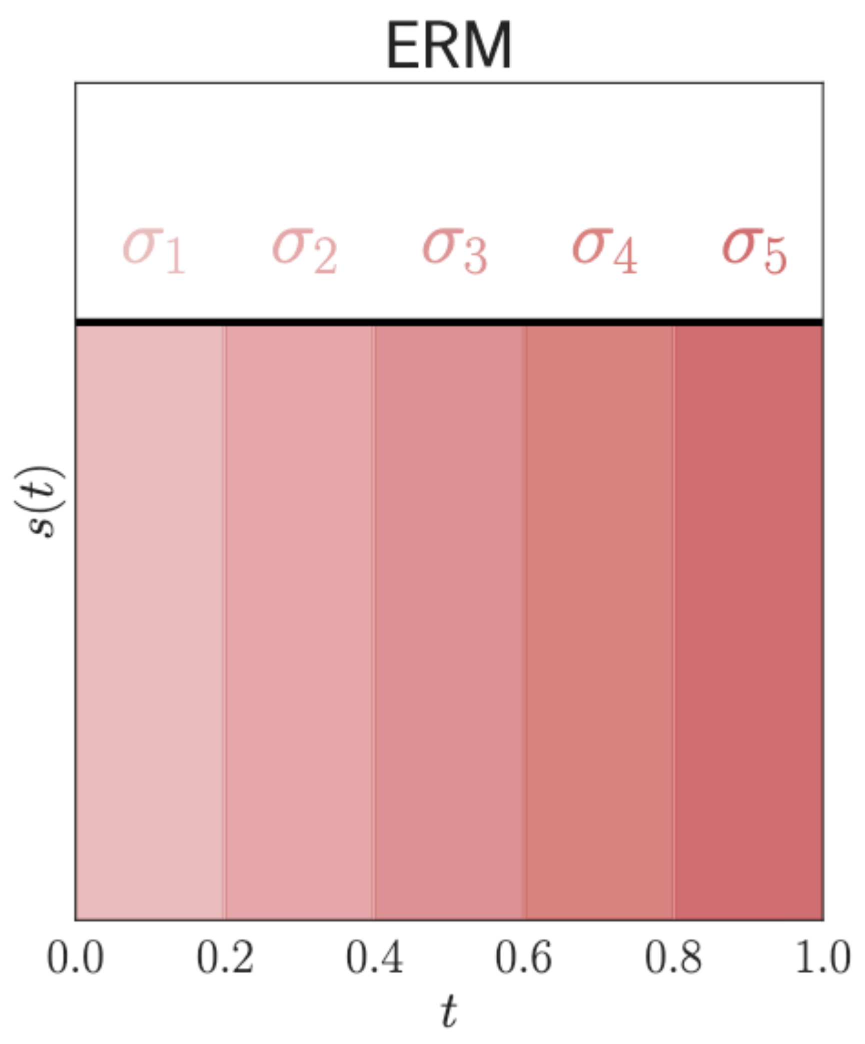
ambiguity set of distributions.



$$x_i \ , y_i \sim P$$

$$\mathbb{E}_P\left[\ell\left(h(X), Y\right)\right]$$

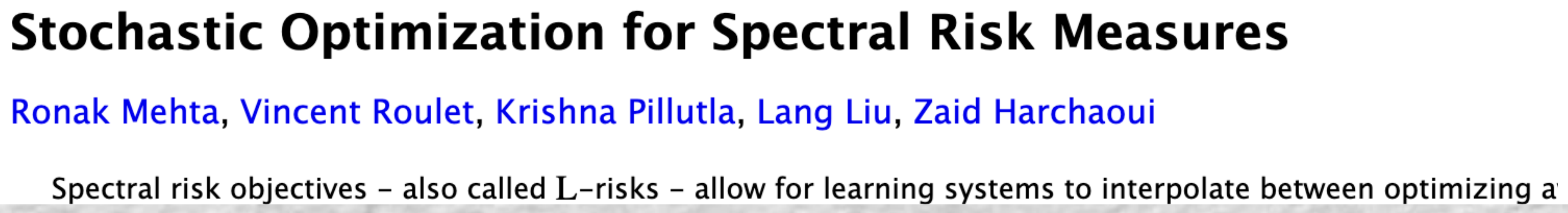
Machine Learning: *risk measures*



Traditional machine learning algorithms designed to work well on *average* across the population. Not suitable for heterogeneous populations.

$$\mathbb{E}_P\left[\ell\left(h(X),Y\right)\right]$$

Figure from:



Stochastic Optimization for Spectral Risk Measures

Ronak Mehta, Vincent Roulet, Krishna Pillutla, Lang Liu, Zaid Harchaoui

Spectral risk objectives – also called L-risks – allow for learning systems to interpolate between optimizing a

Machine Learning: risk measures

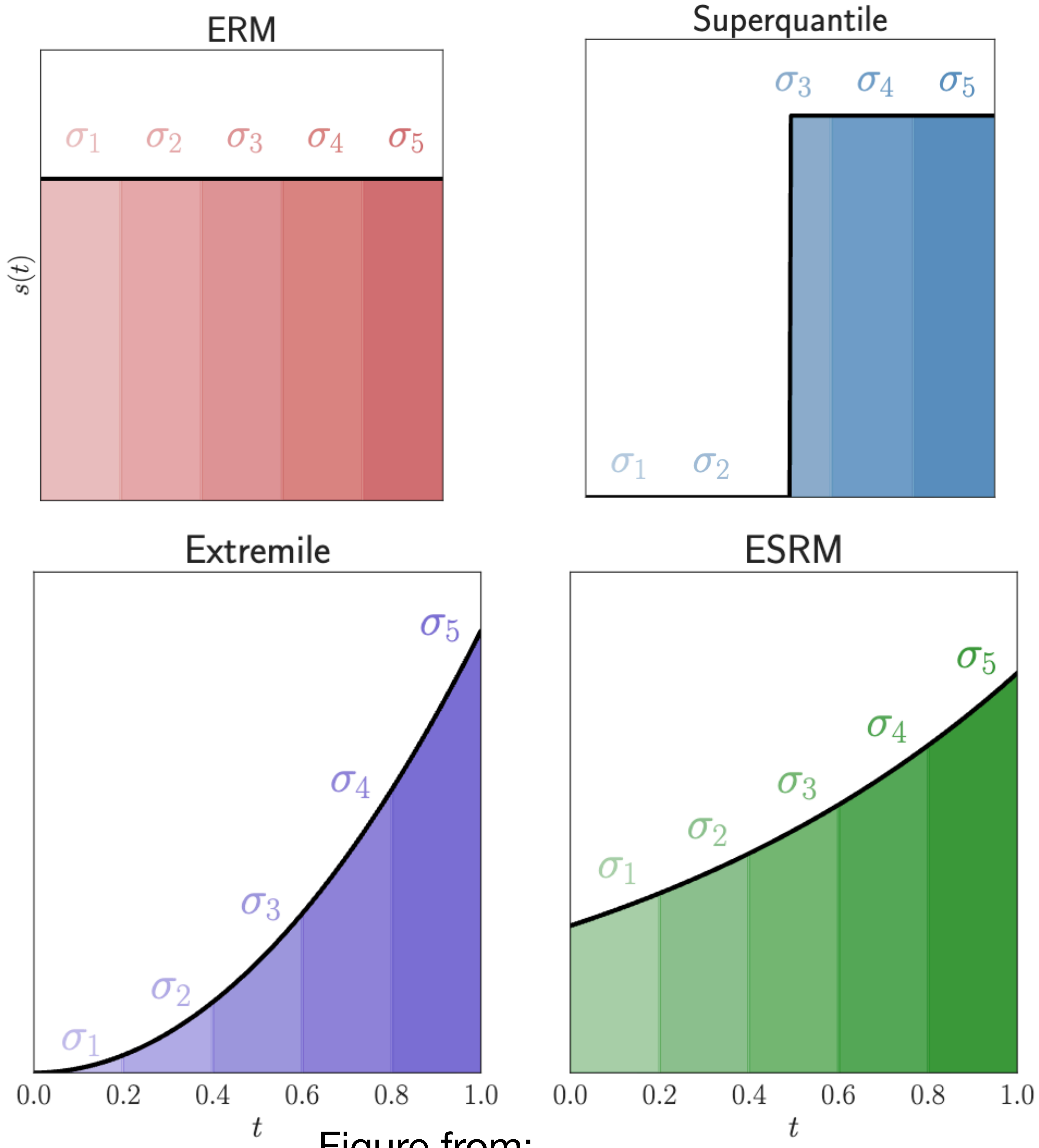
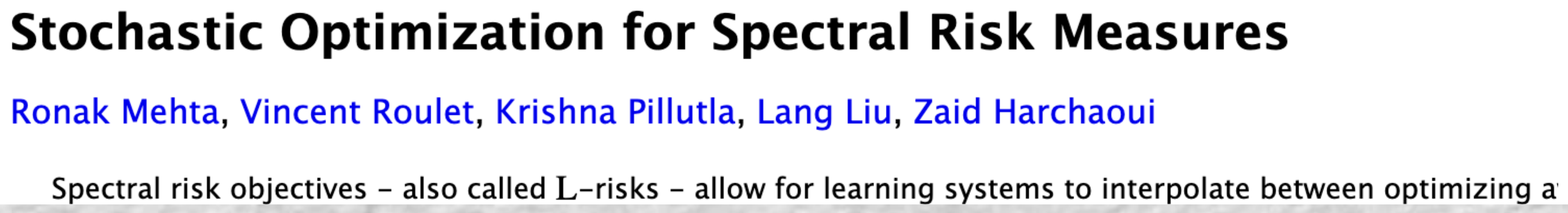


Figure from:



Stochastic Optimization for Spectral Risk Measures

Ronak Mehta, Vincent Roulet, Krishna Pillutla, Lang Liu, Zaid Harchaoui

Spectral risk objectives - also called L-risks - allow for learning systems to interpolate between optimizing a

$$\mathbb{E}_P\left[\ell\left(h(X),Y\right)\right]$$

Machine Learning: *risk measures*

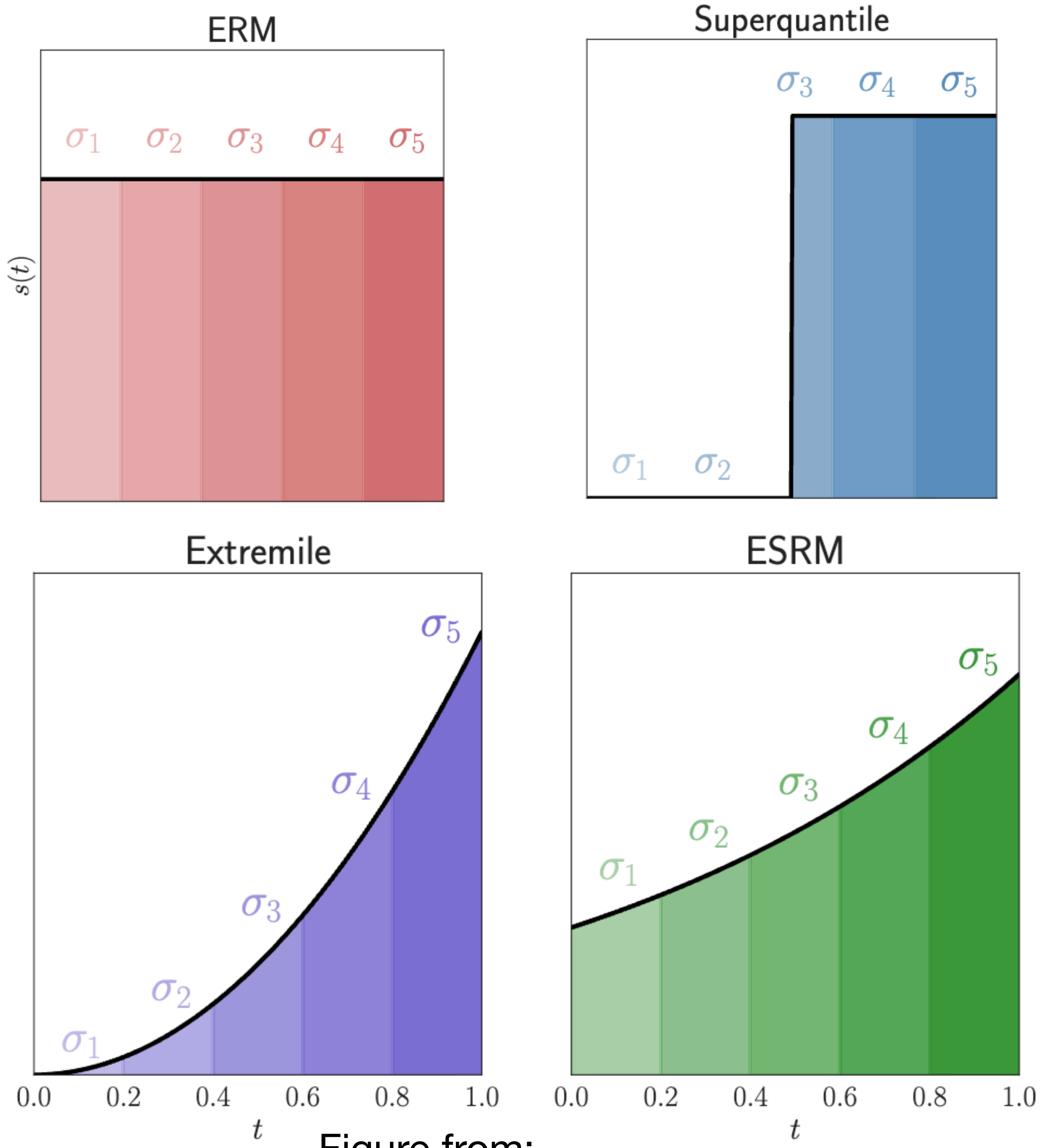
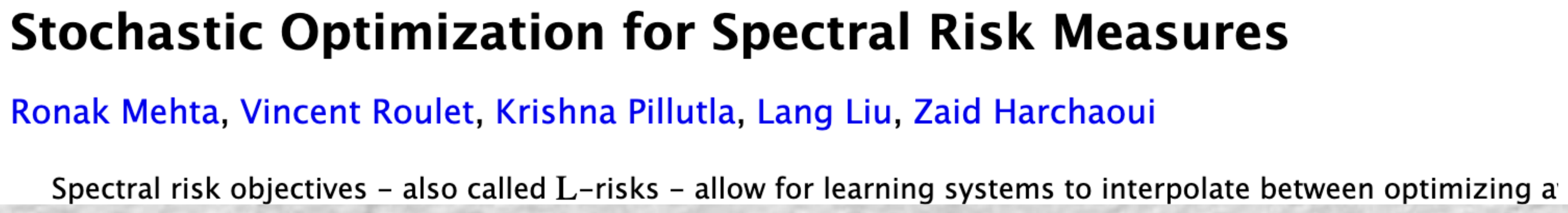


Figure from:



Stochastic Optimization for Spectral Risk Measures

Ronak Mehta, Vincent Roulet, Krishna Pillutla, Lang Liu, Zaid Harchaoui

Spectral risk objectives - also called $\mathbb{L}$-risks - allow for learning systems to interpolate between optimizing a

(*Rockafellar*). *Coherent risk measures have a unique representation as a set of probability distributions, $\mathcal{Q}$.*

$$\mathbb{E}_P\left[\ell\left(h(X),Y\right)\right]$$

Machine Learning: *Multi-distribution learning*

Given a set $\mathscr{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathscr{Q}$.

Machine Learning

$$\mathbb{E}_P\left[\ell\left(h(X), Y\right)\right]$$

Machine Learning

$$\mathbb{E}_P\left[\ell\left(h(X), Y\right)\right]$$

Machine Learning

$$\mathbb{E}_P\left[\ell\left(h\left(X\right), Y\right)\right]$$

Machine Learning: *Institutional separation*

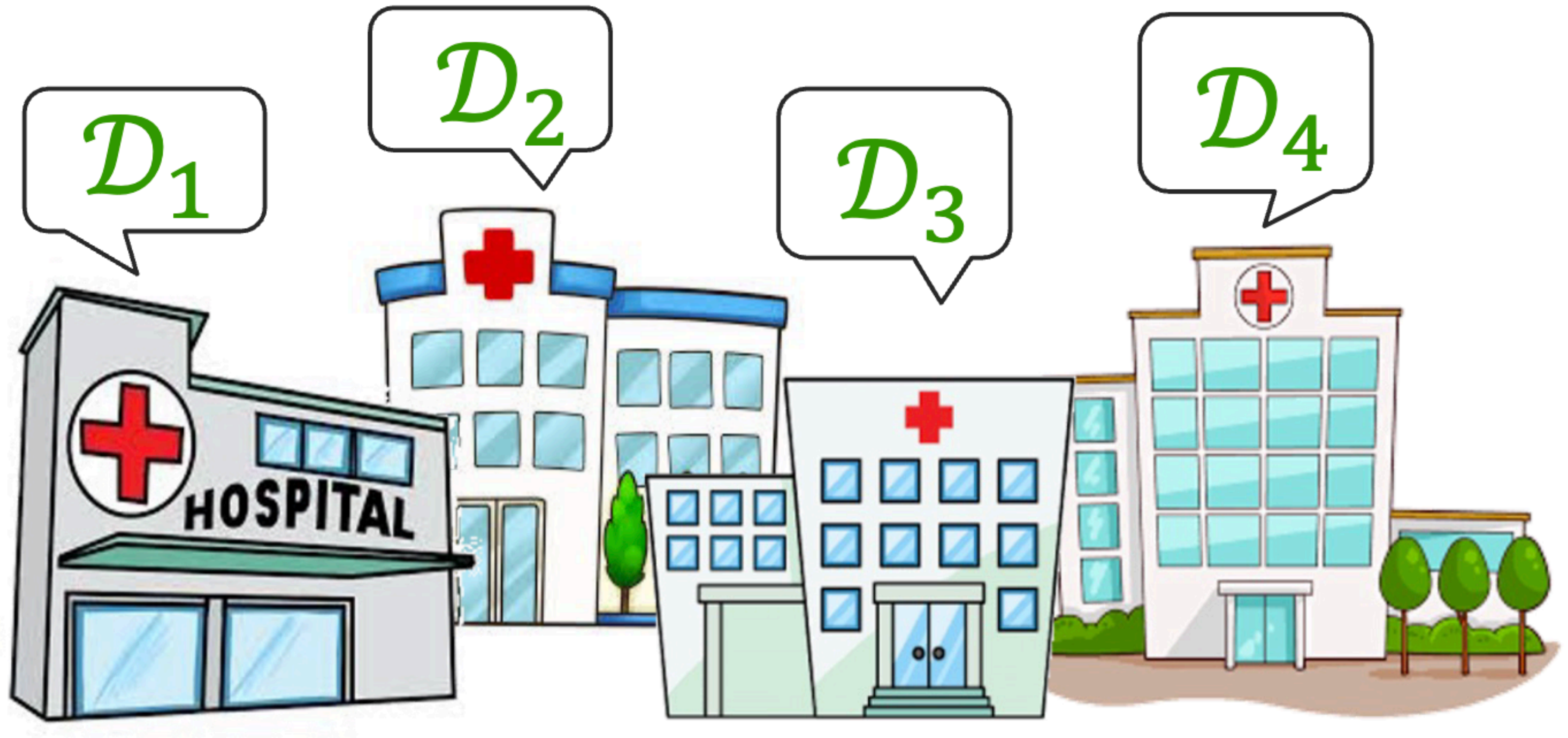
designers

$$\mathbb{E}_P\left[\ell\left(h(X), Y\right)\right]$$

Machine Learning: *Institutional separation*

designers

decision-makers



$$\mathbb{E}_P\left[\ell\left(h\left(X\right), Y\right)\right]$$

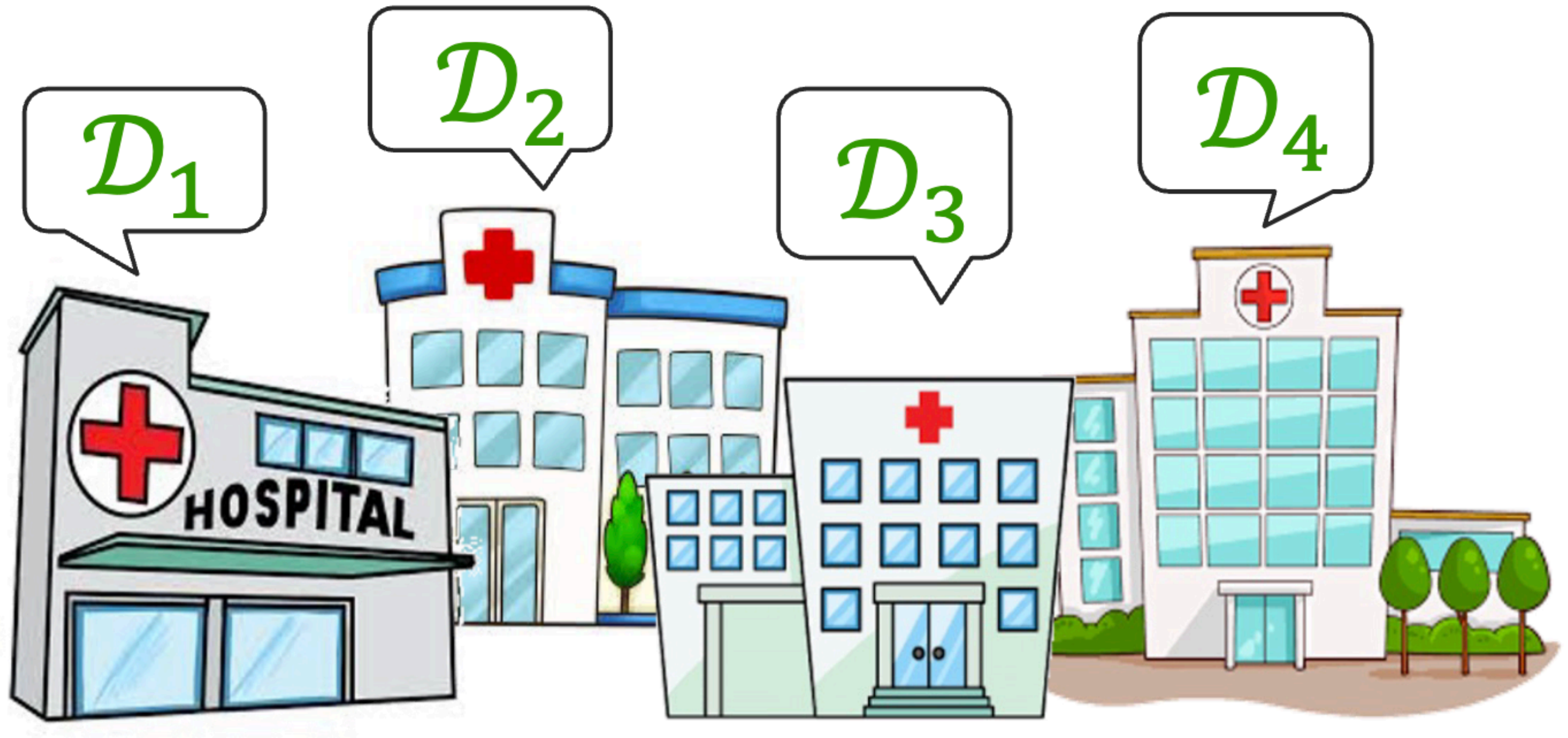
arbitrary loss / utility functions

Machine Learning: *Institutional separation*

designers

bridge?

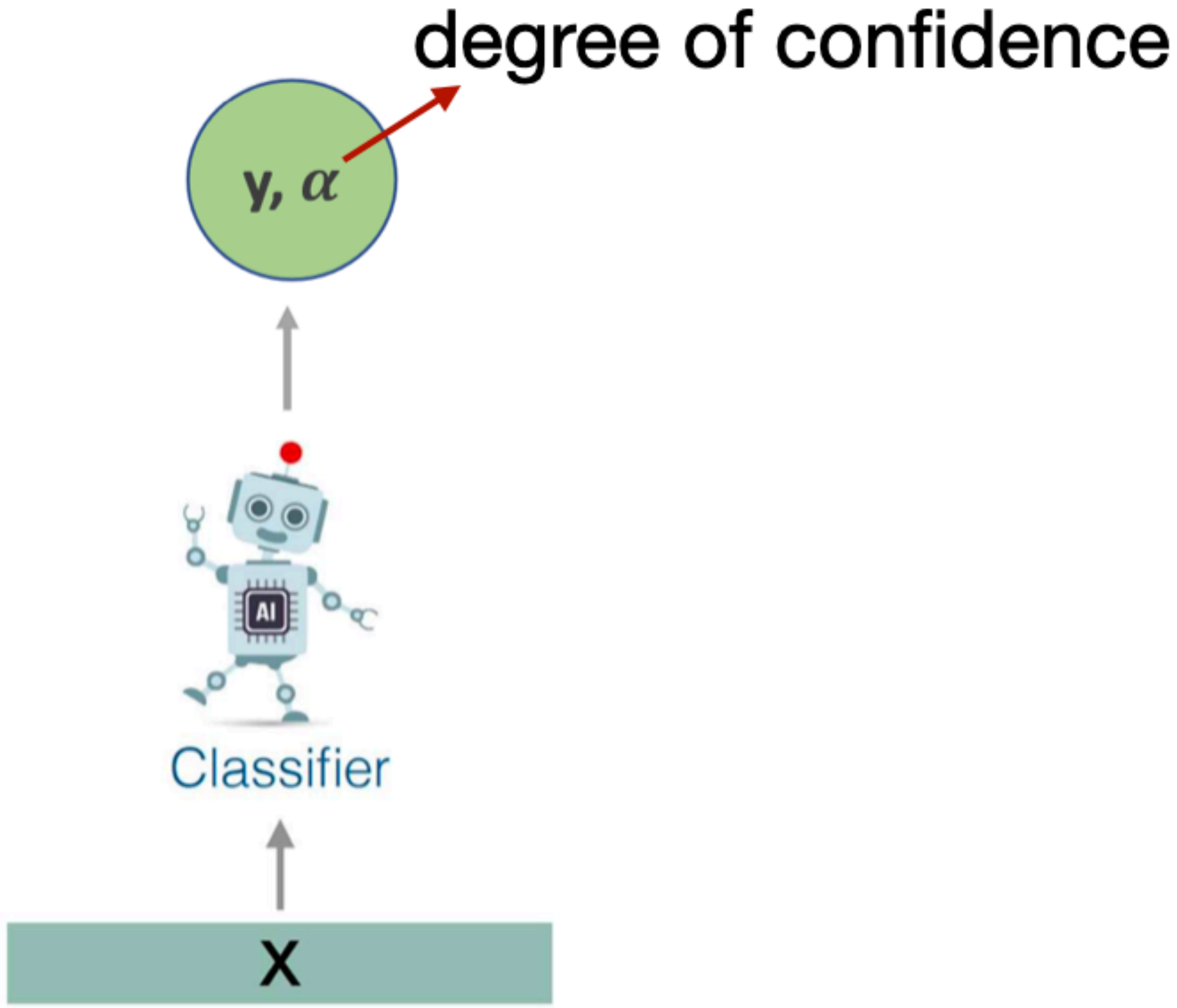
decision-makers



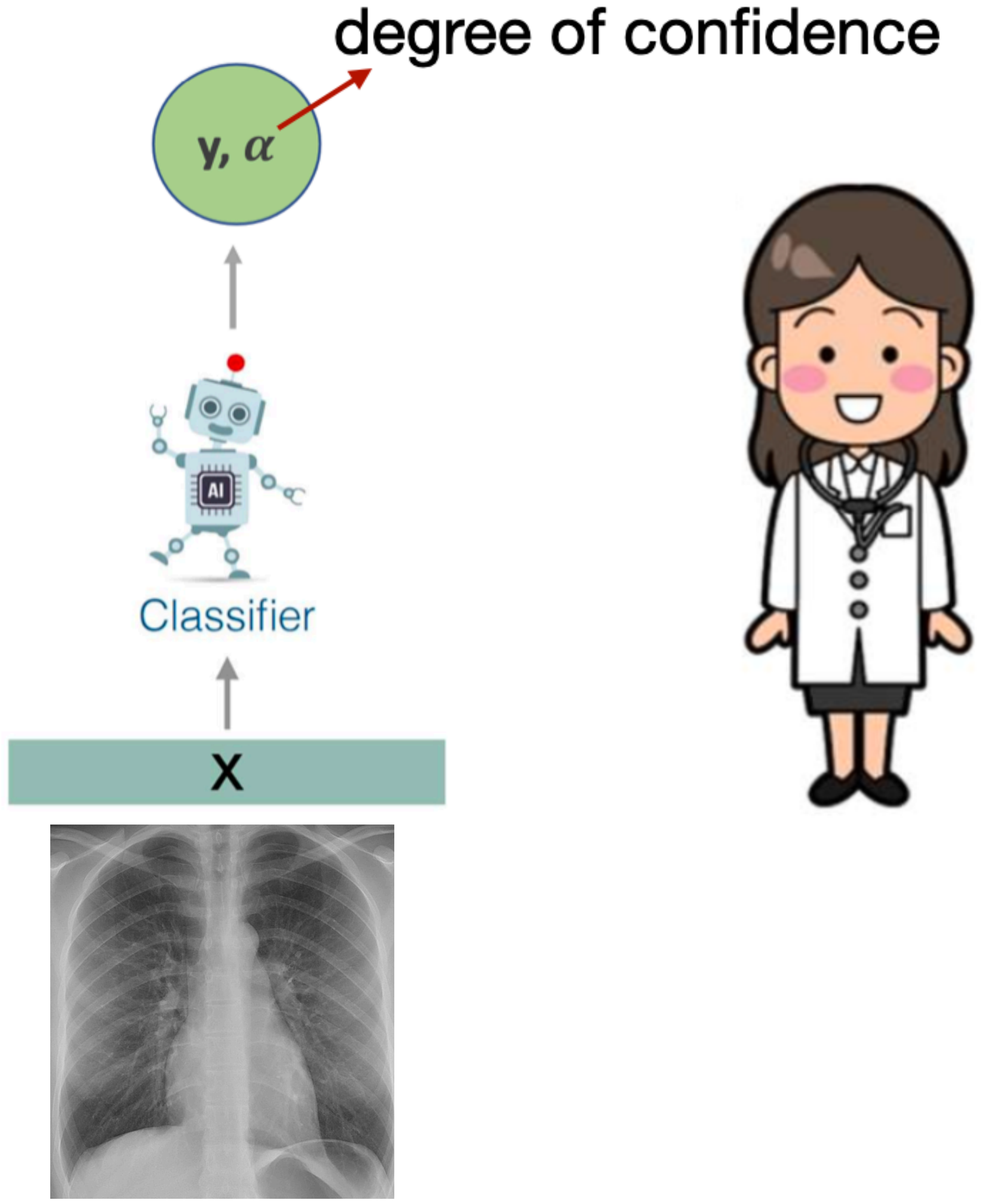
$$\mathbb{E}_P\left[\ell\left(h(X), Y\right)\right]$$

arbitrary loss / utility functions

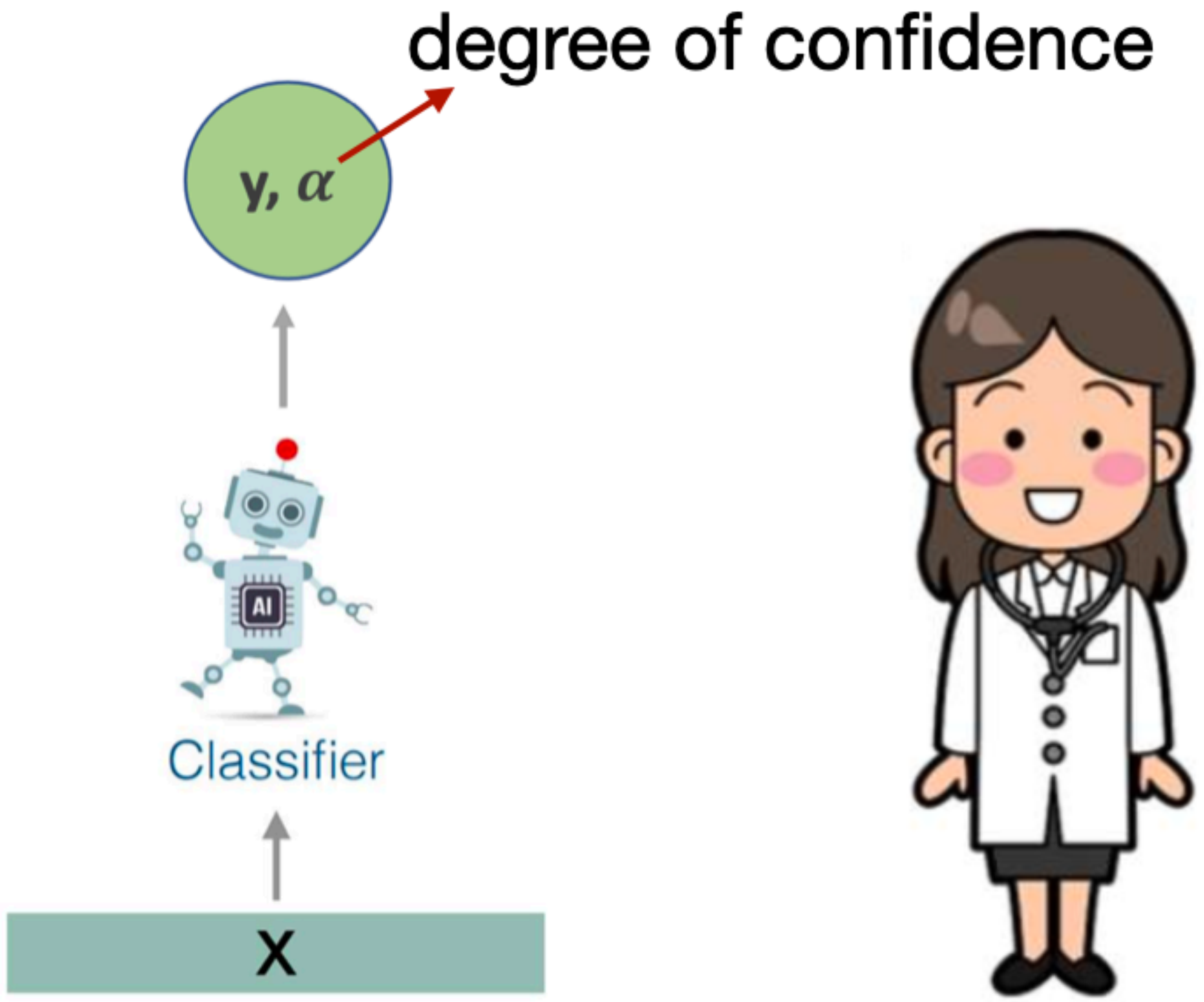
Machine Learning: *Institutional separation*



Machine Learning: *Institutional separation*



Machine Learning: *Institutional separation*



Given the prediction, a doctor has to act:

to treat the patient or not
treatment A or treatment B

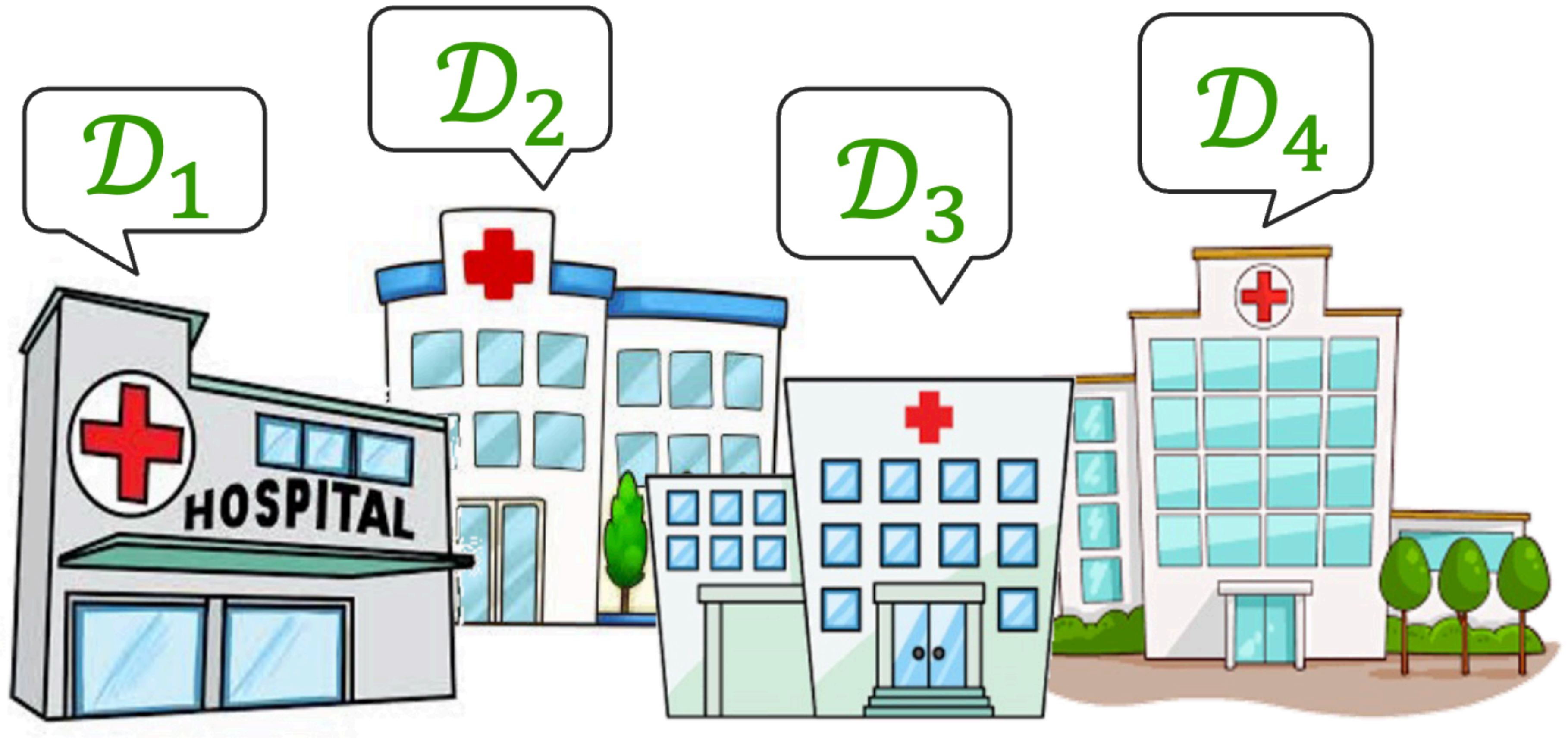
arbitrary loss / utility functions

Machine Learning: *Calibration*

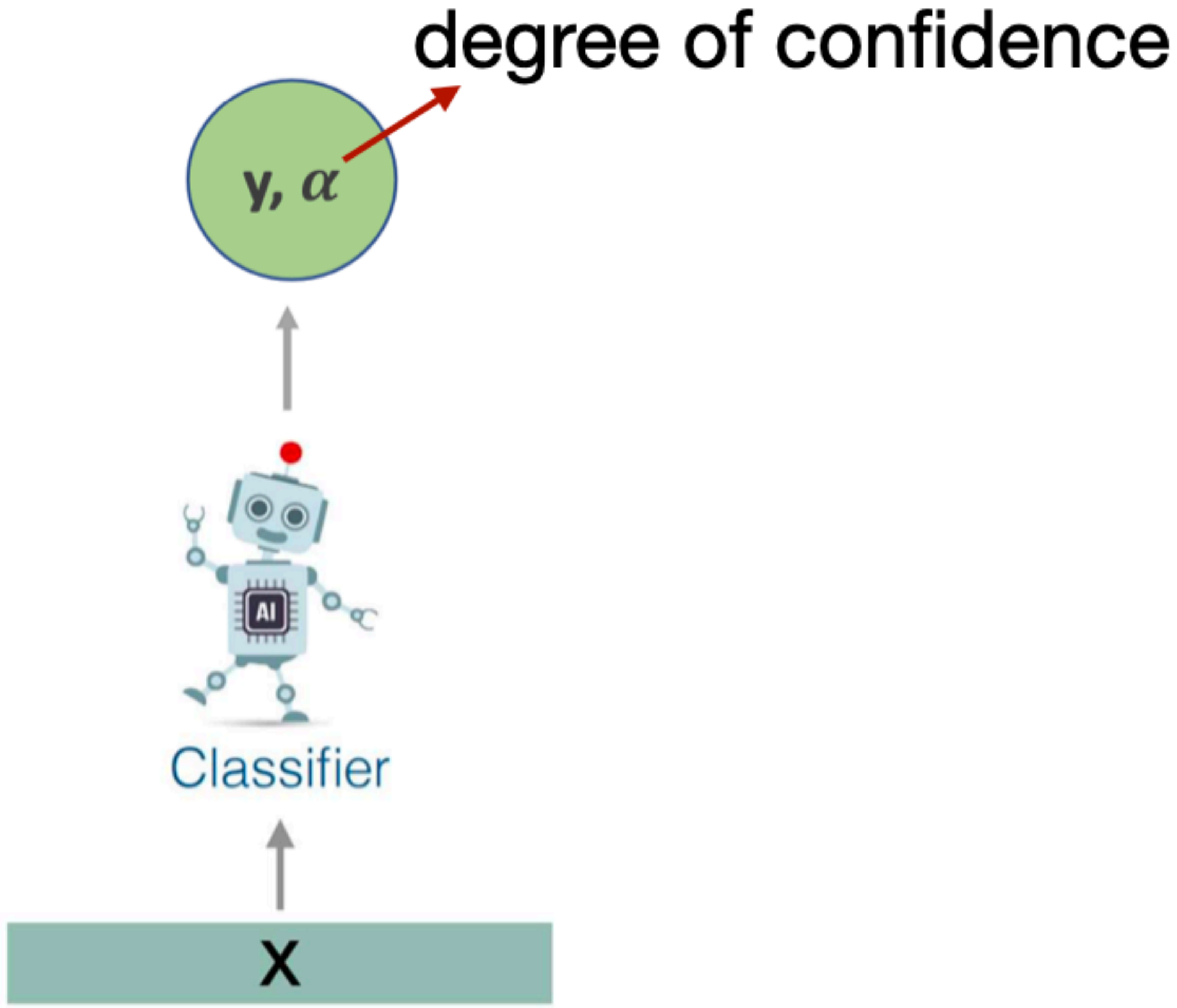
designers

bridge? →

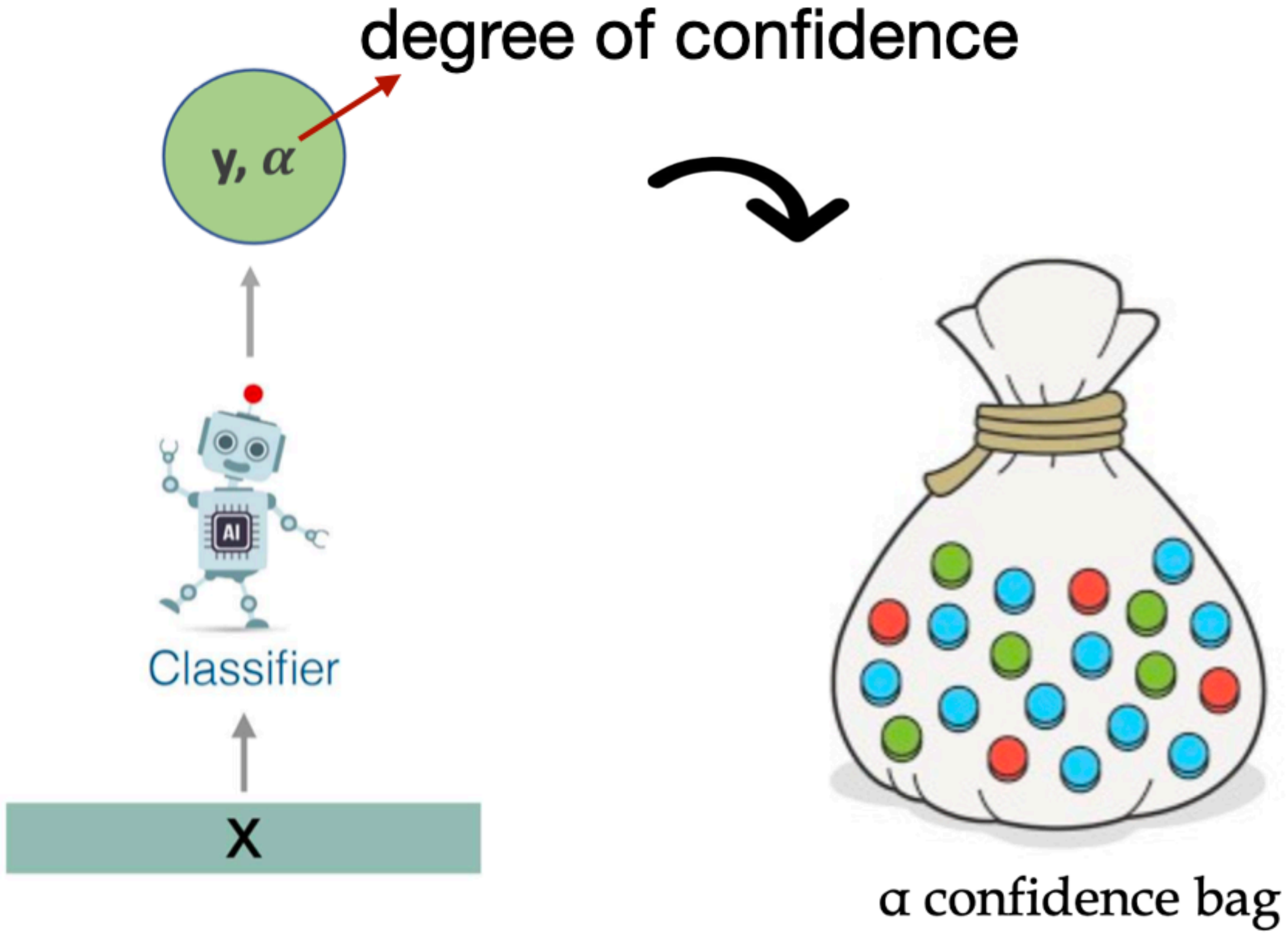
decision-makers



Machine Learning: *Calibration*



Machine Learning: *Calibration*



degree of confidence

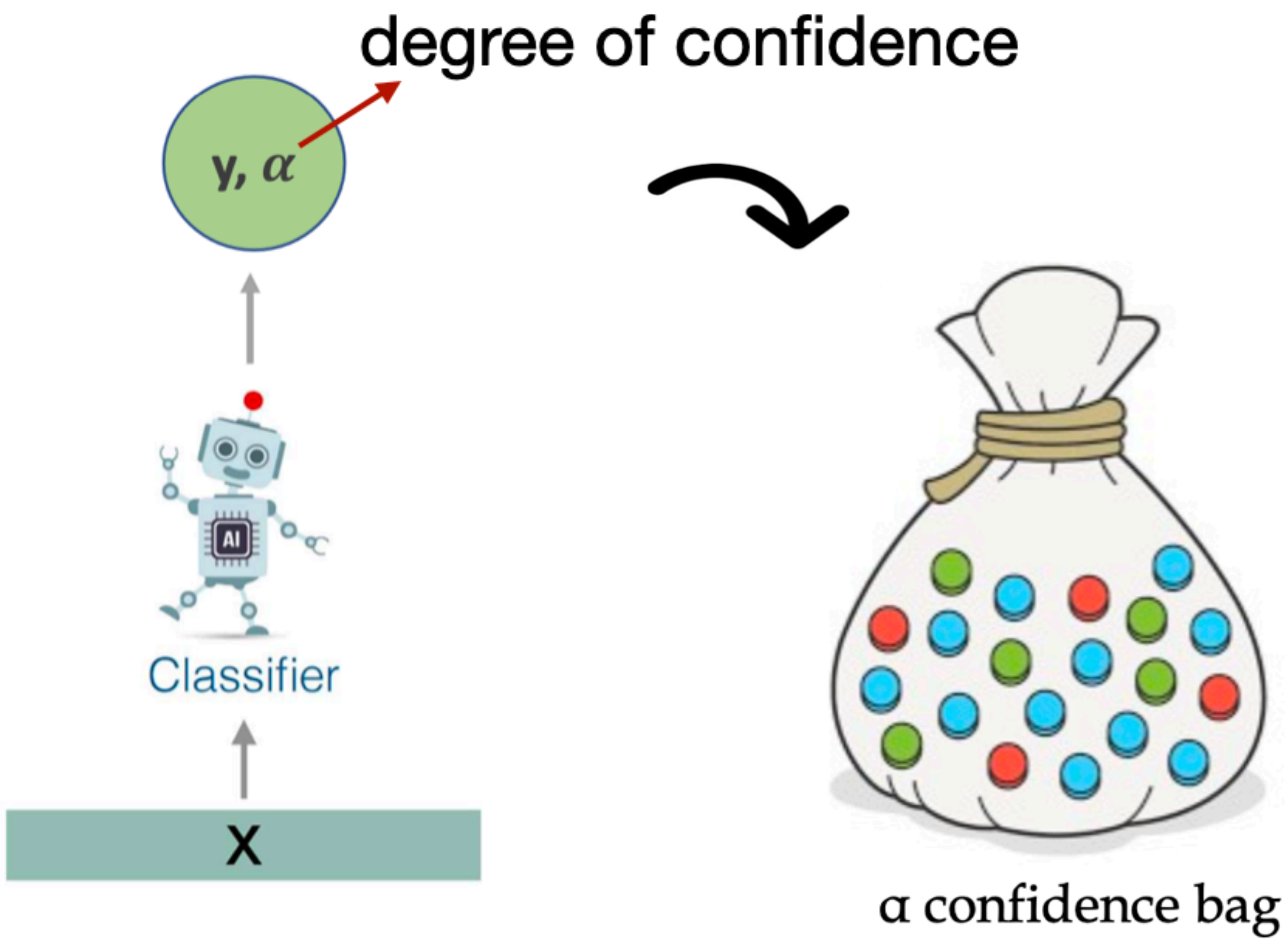
y, α

Classifier

X

α confidence bag

Machine Learning: *Calibration*



degree of confidence

y, α

Classifier

x

α confidence bag

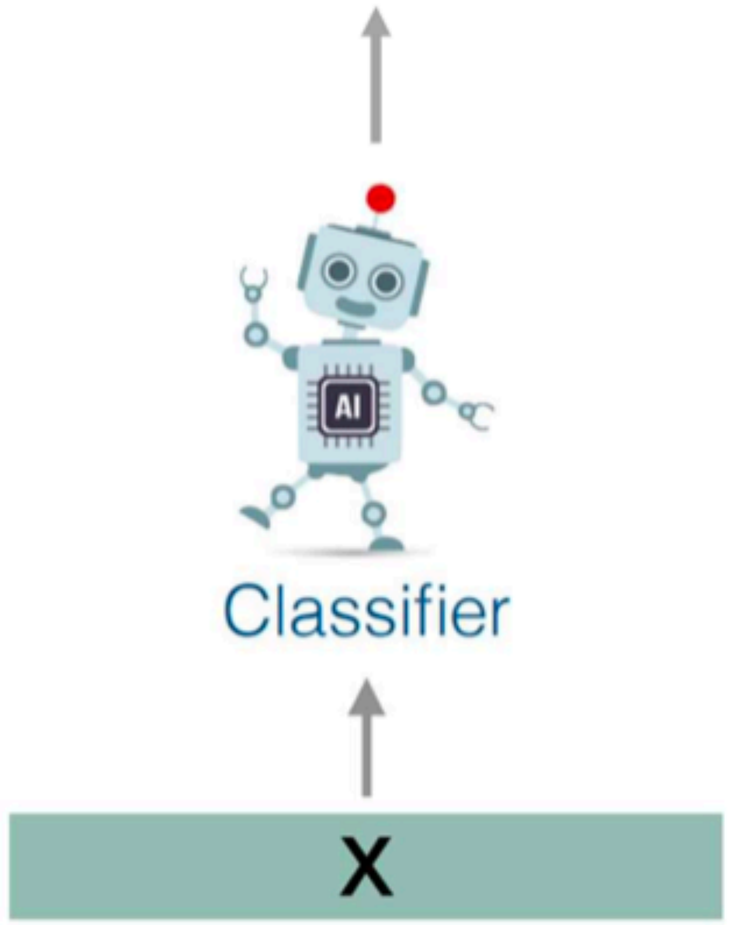
Informal:

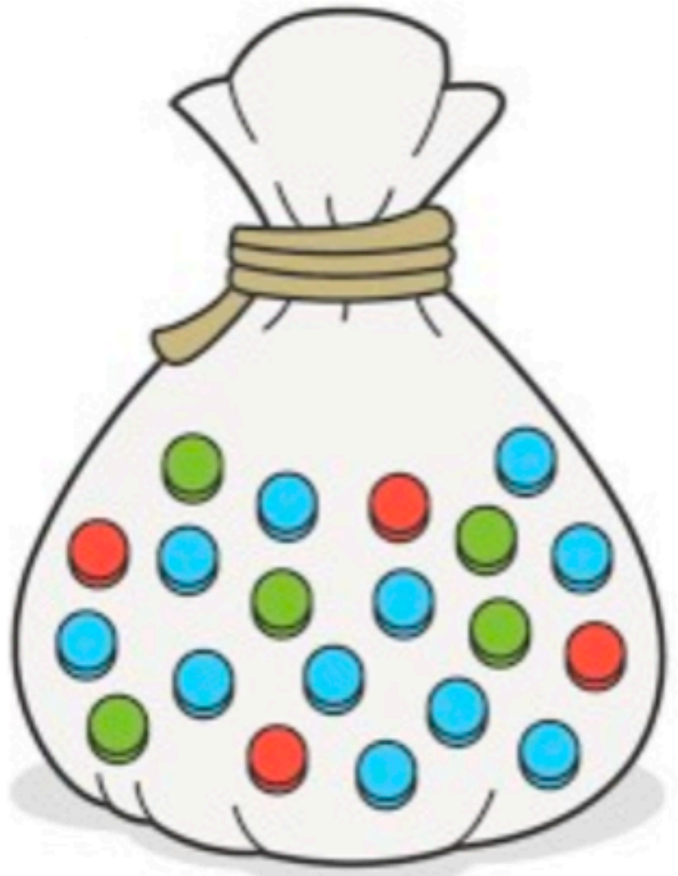
Confidence calibration means that the proportion of samples for which the classifier makes *correct* prediction must be α.

Machine Learning: *Calibration*

Definition 2.1. *(Canonical Calibration).* Given d some divergence measure, e.g. squared error, a confidence predictor $h : \mathcal{X} \rightarrow \Delta^{|\mathcal{Y}|}$ is said to be (perfectly) canonically calibrated if the following holds true:

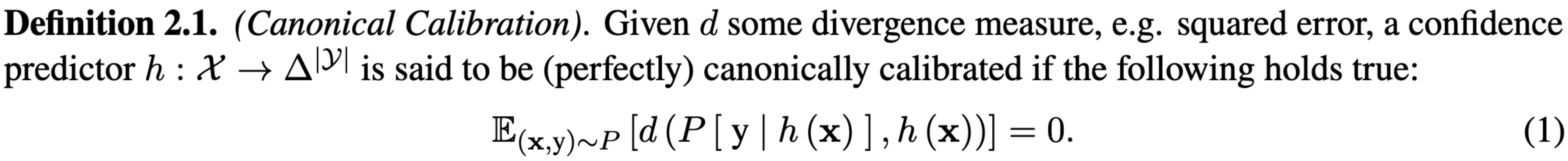
$$\mathbb{E}_{(\mathbf{x},\mathrm{y})\sim P}\left[d\left(P\left[\,\mathrm{y} \mid h\left(\mathbf{x}\right)\,\right], h\left(\mathbf{x}\right)\right)\right] = 0. \tag{1}$$





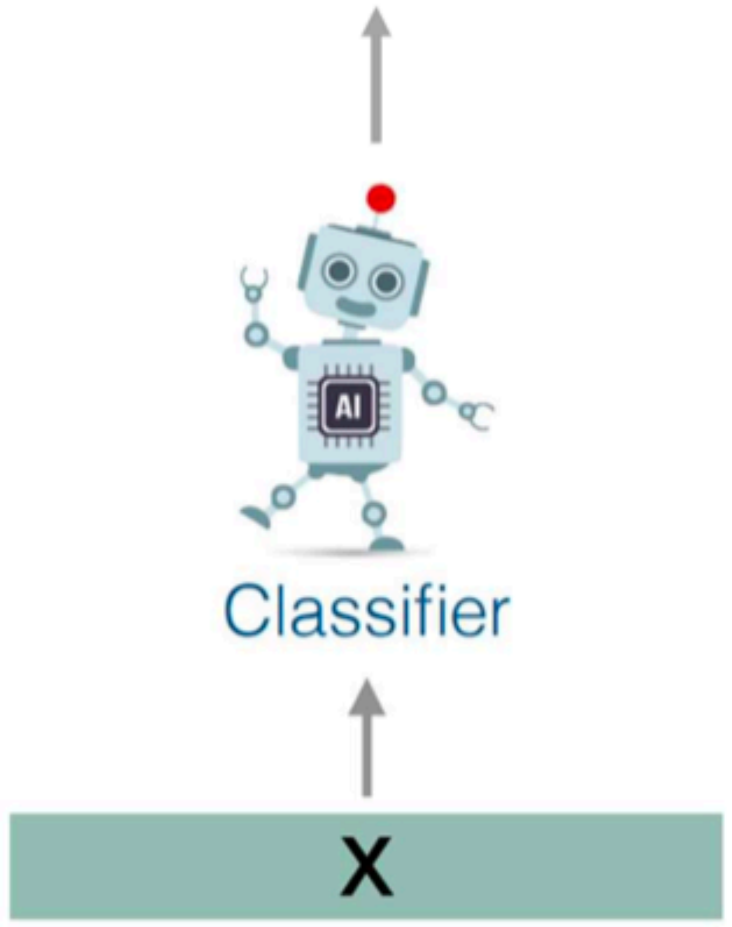
α confidence bag

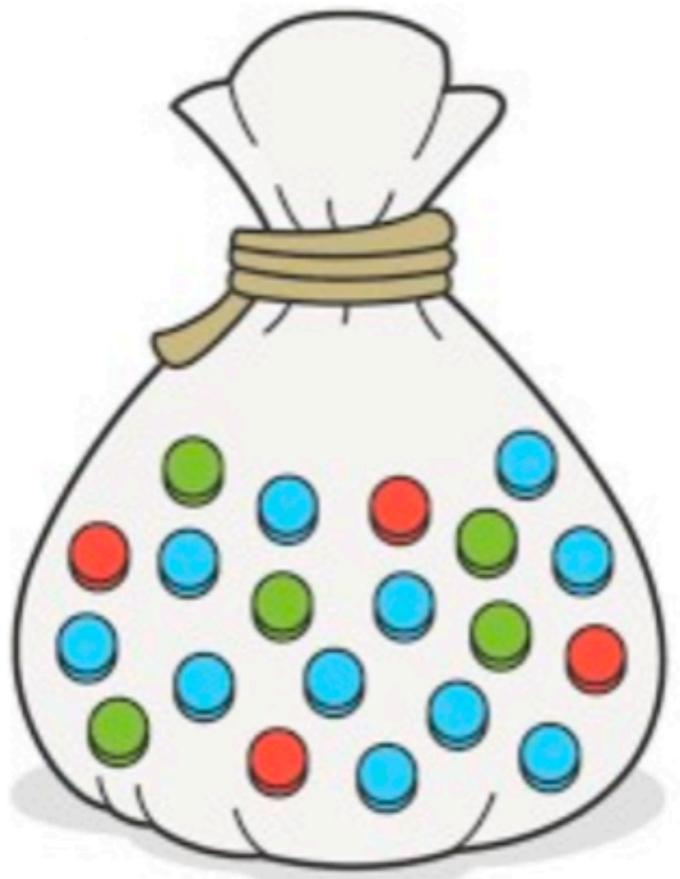
Machine Learning: *Calibration*



Definition 2.1. *(Canonical Calibration)*. Given d some divergence measure, e.g. squared error, a confidence predictor $h : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ is said to be (perfectly) canonically calibrated if the following holds true:

$$\mathbb{E}_{(\mathbf{x},\mathrm{y})\sim P}\left[d\left(P\left[\,\mathrm{y} \mid h\left(\mathbf{x}\right)\,\right], h\left(\mathbf{x}\right)\right)\right] = 0. \tag{1}$$

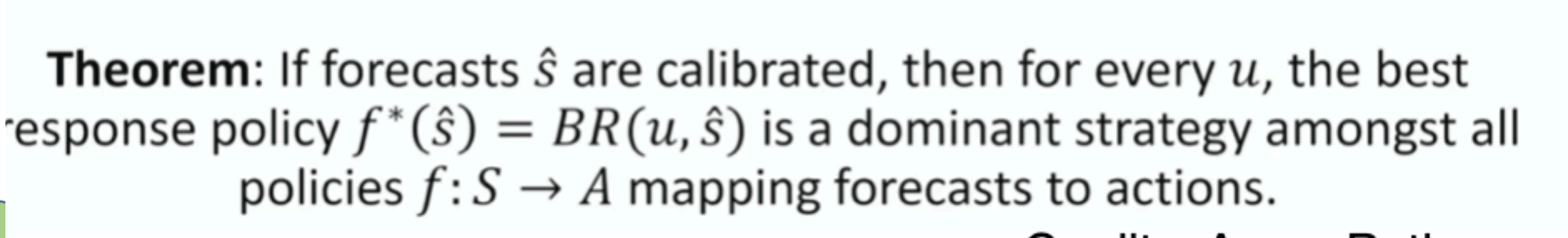




α confidence bag

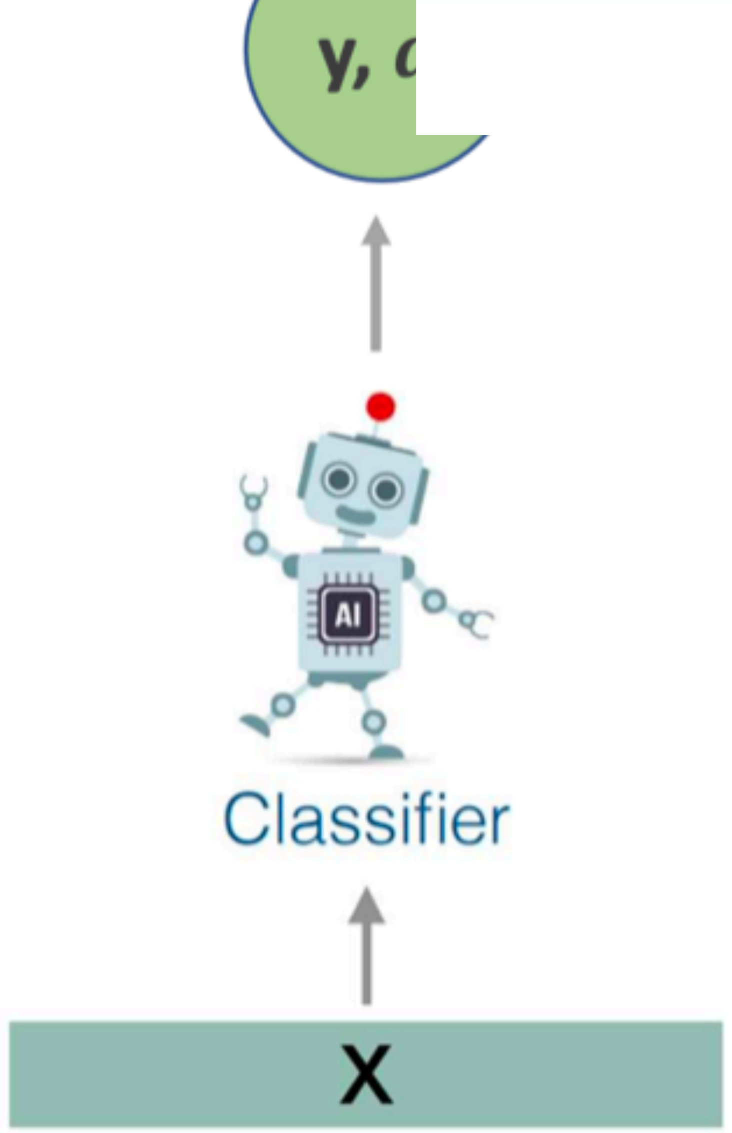
Alignment of the prediction $h(X)$ and the reality $P(Y \mid h(X))$.

Machine Learning: *Calibration bridges institutional separation*



Theorem: If forecasts $\hat{s}$ are calibrated, then for every u, the best response policy $f^*(\hat{s}) = BR(u, \hat{s})$ is a dominant strategy amongst all policies $f: S \to A$ mapping forecasts to actions.

Credits: Aaron Roth



Alignment of the prediction $h(X)$ and the reality $P(Y \mid h(X))$.

Machine Learning: *Calibration bridges institutional separation*

For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Machine Learning: *Calibration bridges institutional separation*

For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Calibration demands

$$P(Y = 1 \,|\, h(X) = \nu) = \nu$$

Of all the days, it was announced that it'll rain with confidence ν, it actually rained ν times.

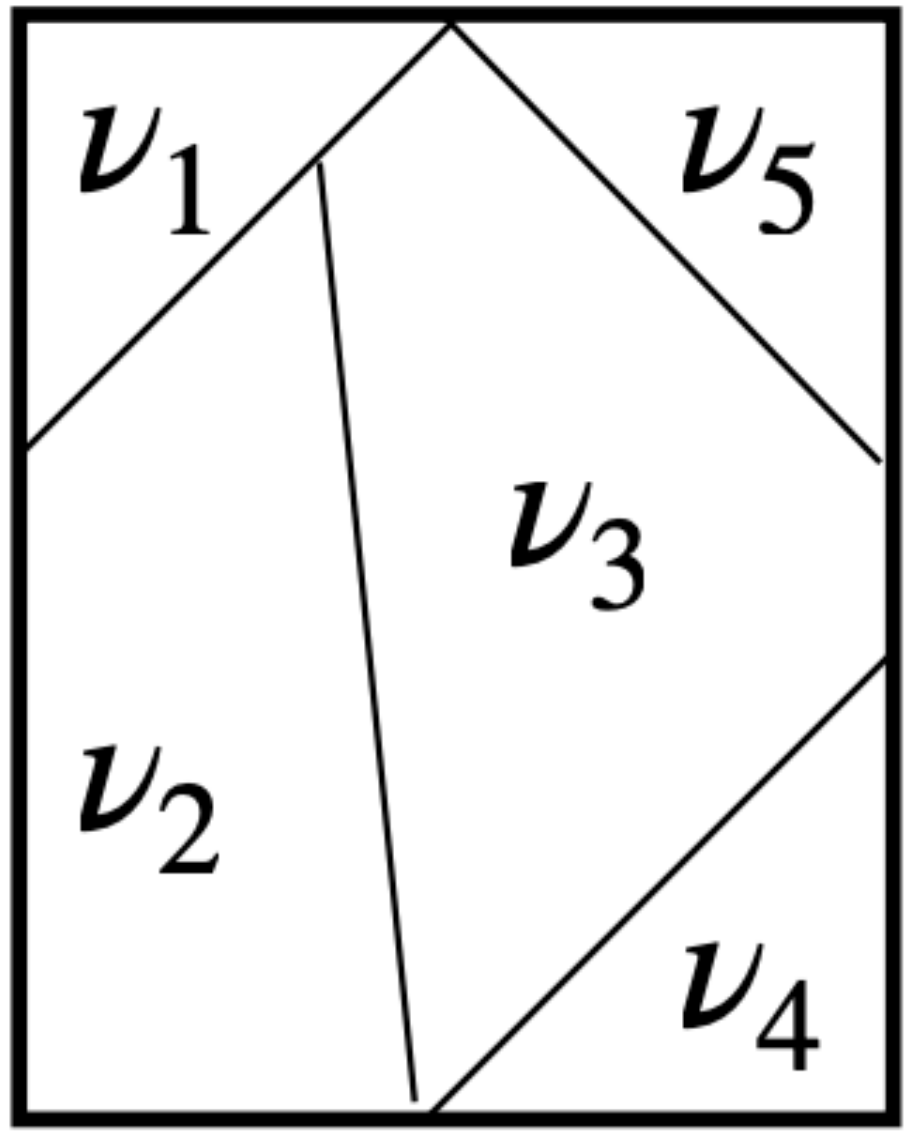
Machine Learning: *Calibration bridges institutional separation*

For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Calibration demands

$$P(Y = 1 \mid h(X) = \nu) = \nu$$

Of all the days, it was announced that it'll rain with confidence ν, it actually rained ν times.



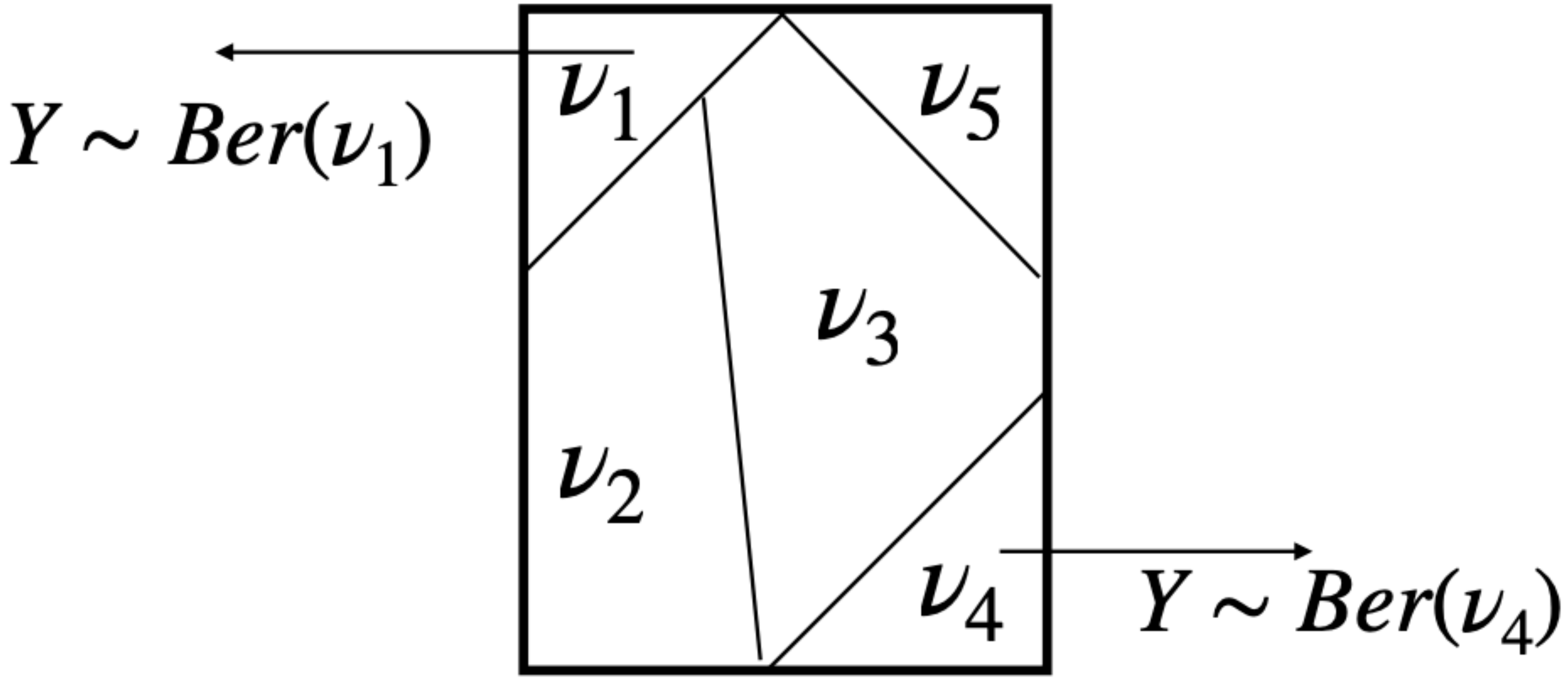
Machine Learning: *Calibration bridges institutional separation*

For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Calibration demands

$$P(Y = 1 \,|\, h(X) = \nu) = \nu$$

Of all the days, it was announced that it'll rain with confidence ν, it actually rained ν times.



Alignment of the prediction $h(X)$ and the reality $P(Y \mid h(X))$.

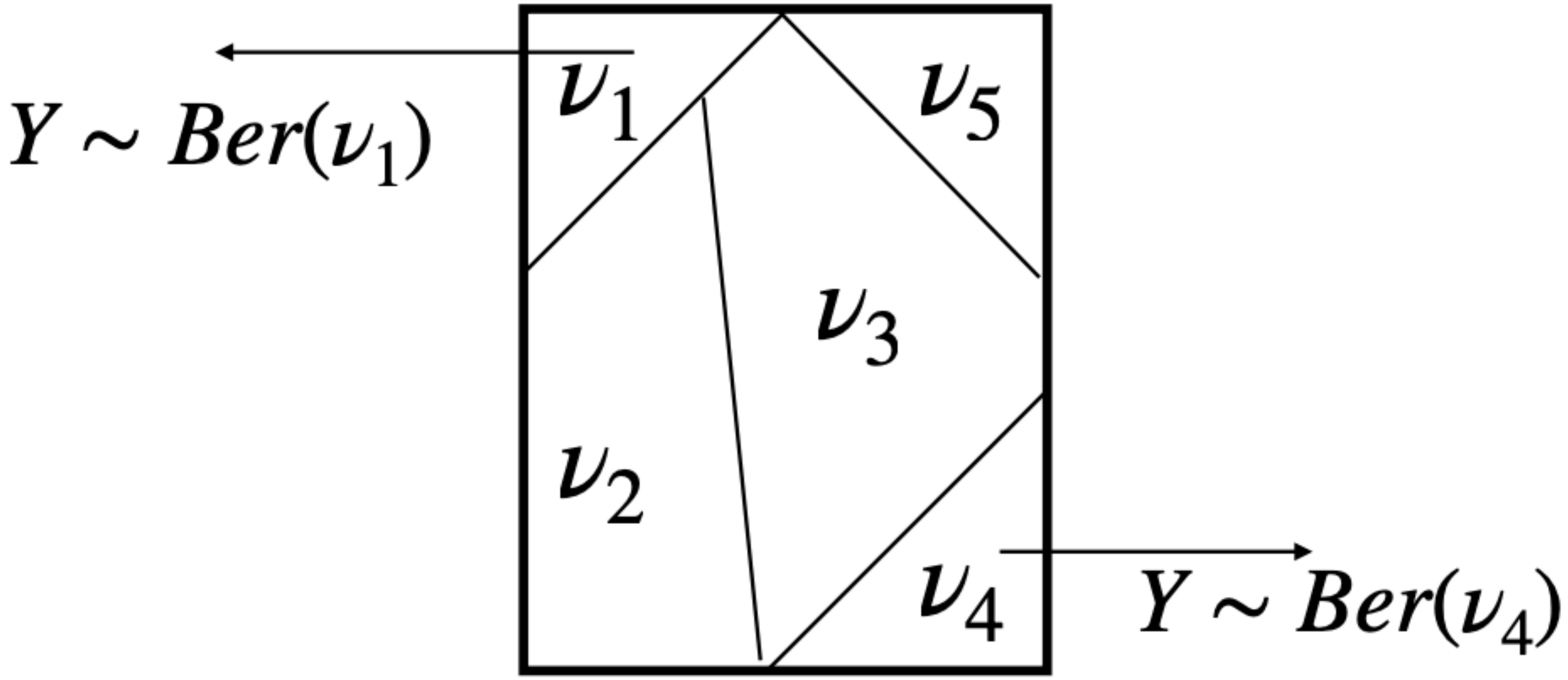
Machine Learning: *Calibration bridges institutional separation*

For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Calibration demands

$$P(Y = 1 \mid h(X) = \nu) = \nu$$

Of all the days, it was announced that it'll rain with confidence ν, it actually rained ν times.



$$a^* = \arg\max_{a \in A} \mathbb{E}_{Y \sim \nu}[u(Y, a)]$$

Best response policy is the dominant policy.

Alignment of the prediction $h(X)$ and the reality $P(Y \mid h(X))$.

Machine Learning: *Calibration bridges institutional separation*

But it is not enough.

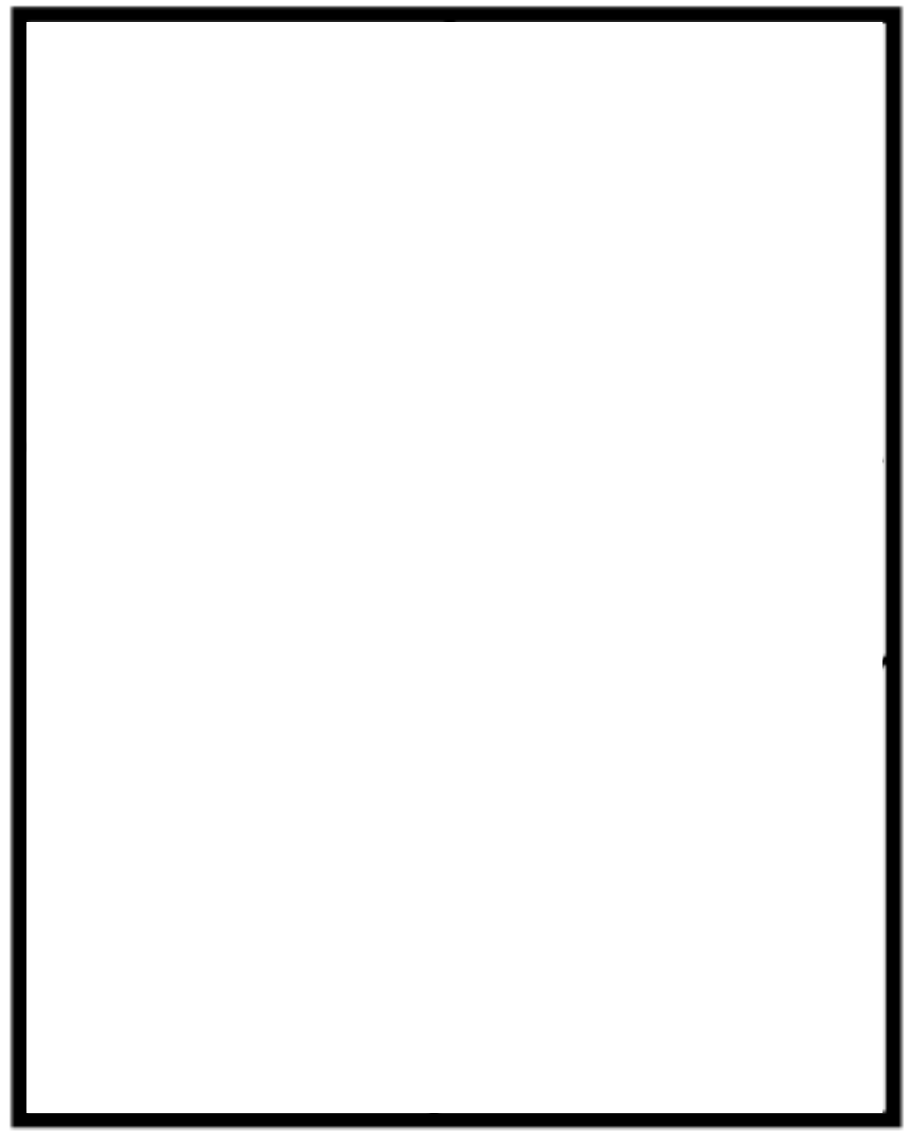
For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Calibration demands

$$P(Y = 1 \mid h(X) = \nu) = \nu$$

Of all the days, it was announced that it'll rain with confidence ν, it actually rained ν times.

$$\nu = \mathbb{E}[Y]$$



Alignment of the prediction $h(X)$ and the reality $P(Y \mid h(X))$.

Machine Learning: *Calibration bridges institutional separation*

But it is not enough.

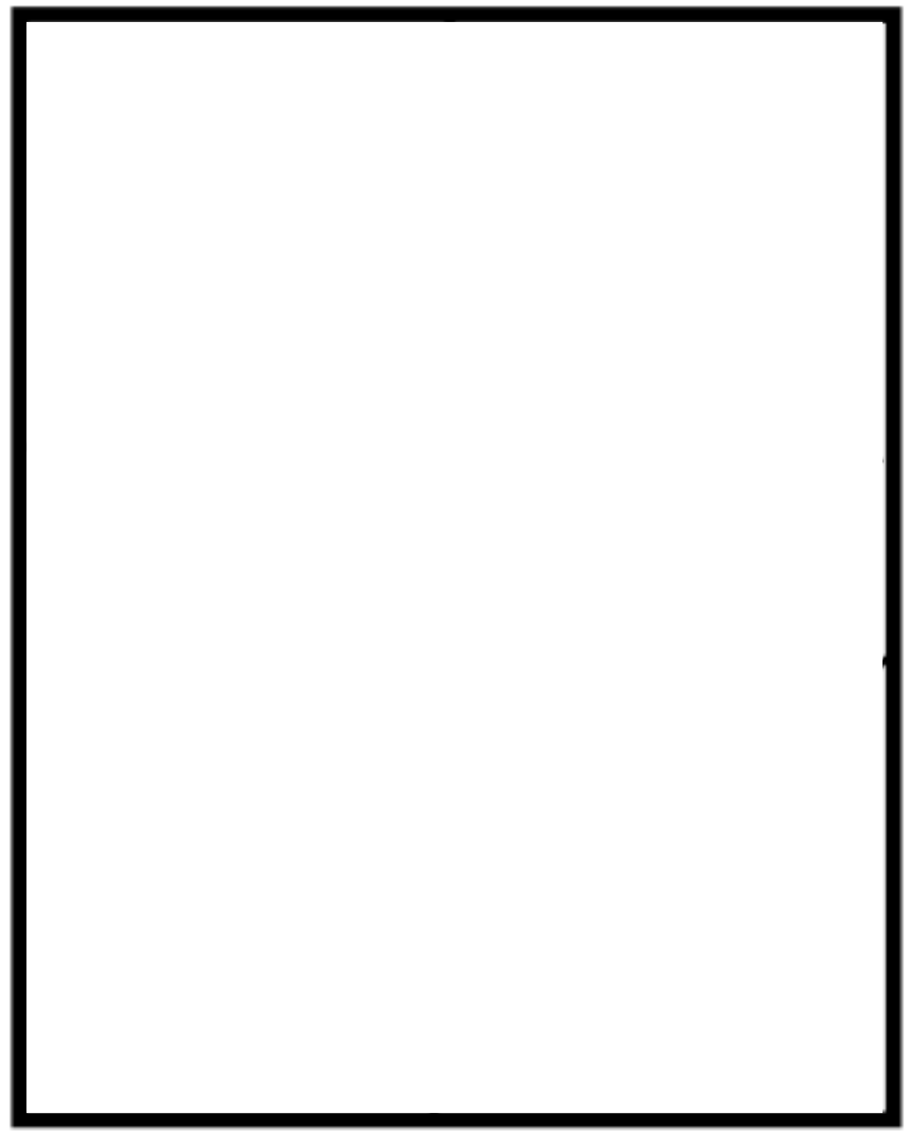
For simplicity, take $\mathscr{Y} = \{0,1\}, h : \mathscr{X} \to \Delta$

Calibration demands

$$P(Y = 1 \,|\, h(X) = \nu) = \nu$$

Of all the days, it was announced that it'll rain with confidence ν, it actually rained ν times.

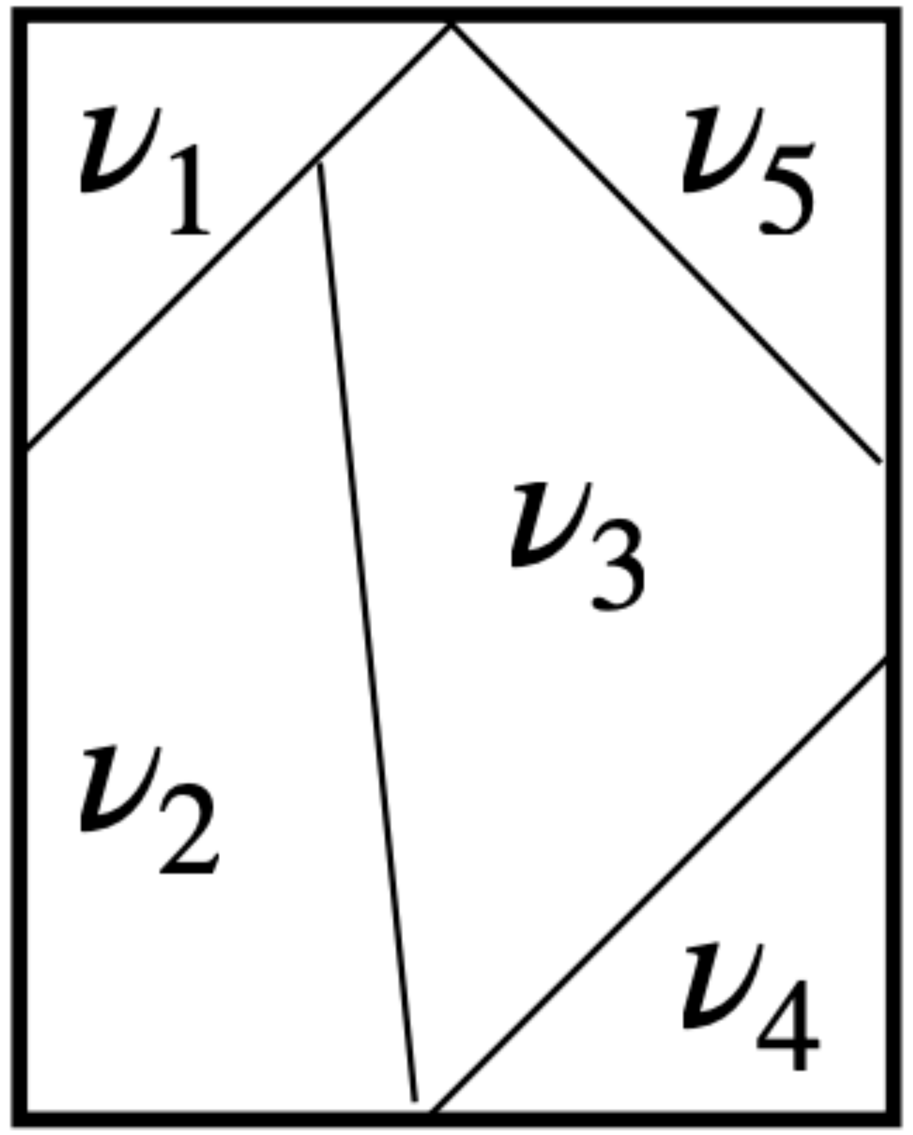
$$\nu = \mathbb{E}[Y]$$



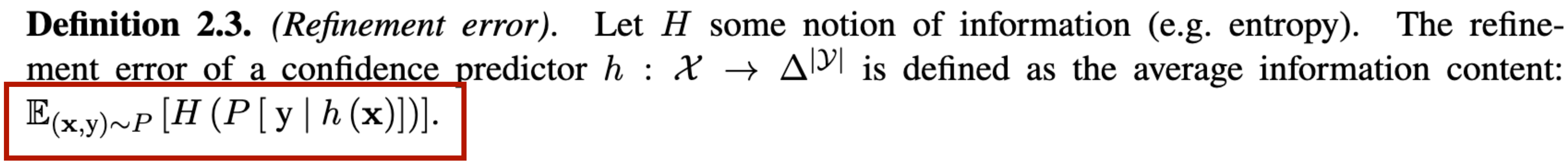
Not informative

Alignment of the prediction $h(X)$ and the reality $P(Y \mid h(X))$.

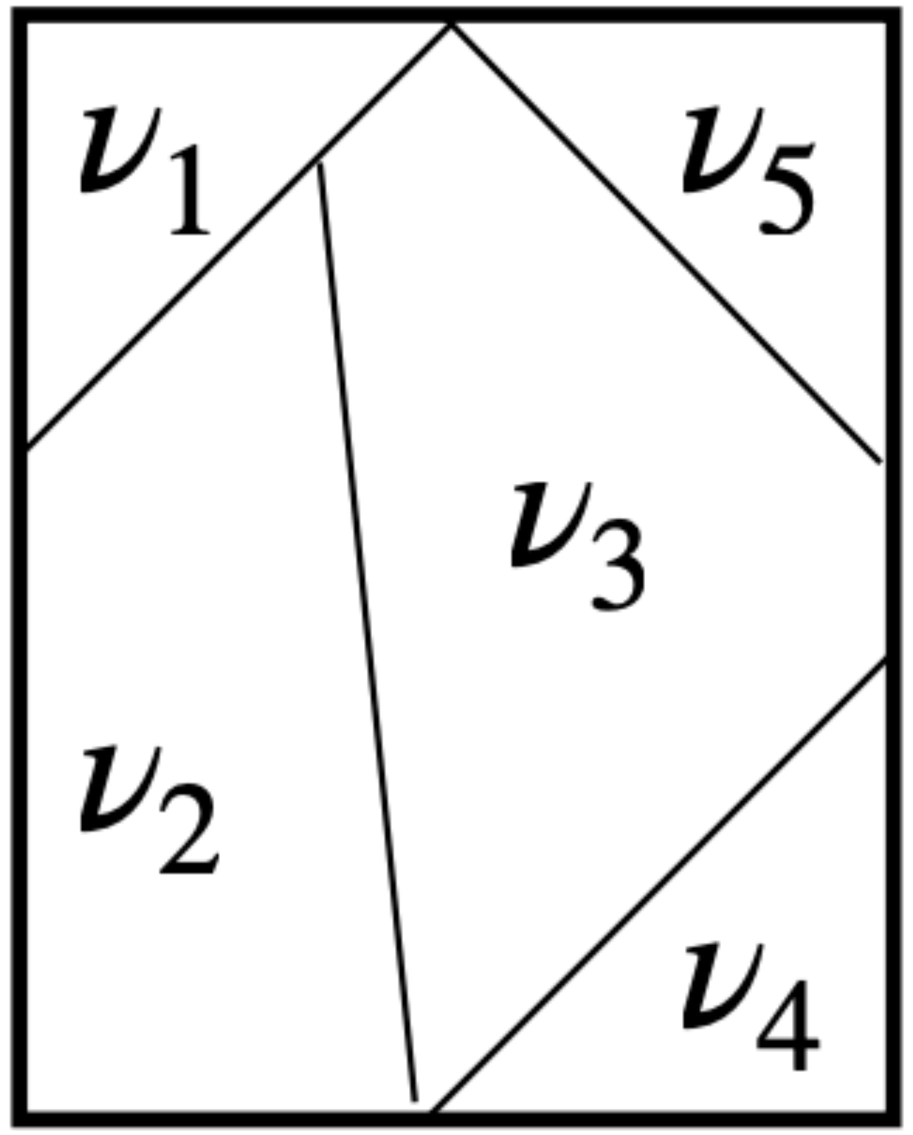
Machine Learning: *Refinement*



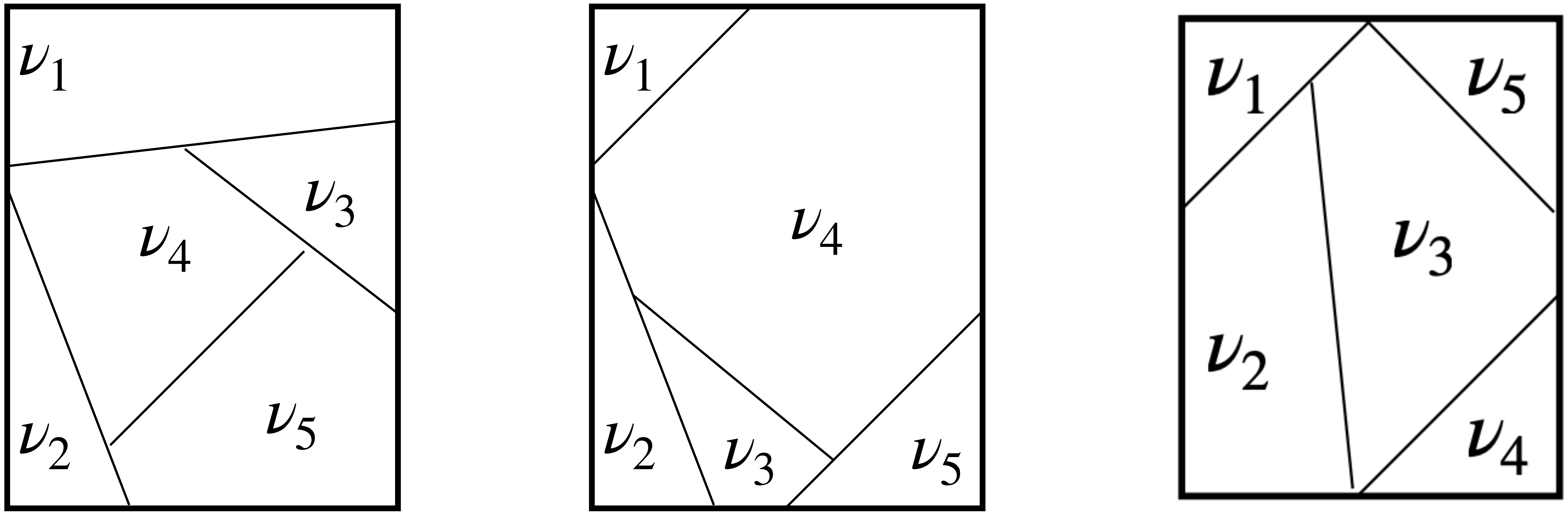
Machine Learning: *Refinement*



Definition 2.3. *(Refinement error).* Let H some notion of information (e.g. entropy). The refinement error of a confidence predictor $h : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ is defined as the average information content: $\mathbb{E}_{(\mathbf{x},\mathrm{y})\sim P}\left[H\left(P\left[\,\mathrm{y} \mid h(\mathbf{x})\right]\right)\right]$.

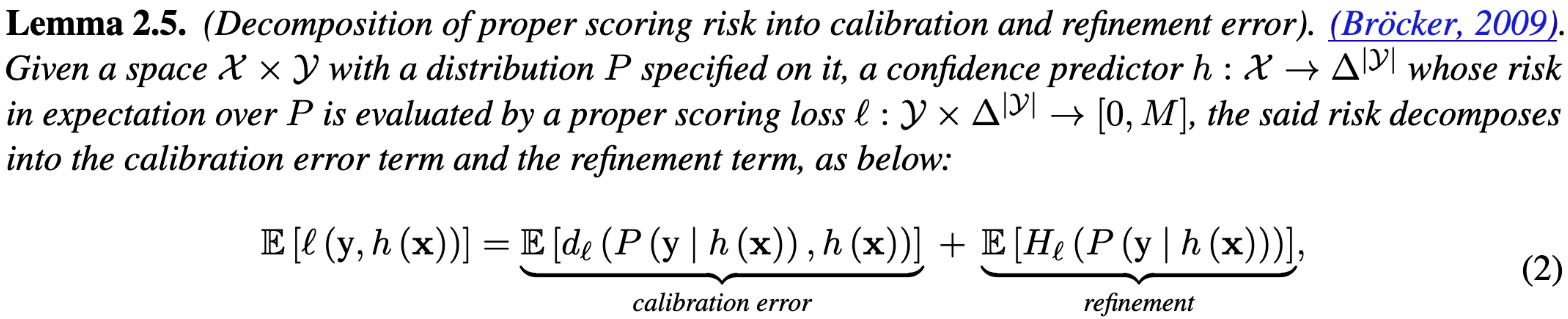


Machine Learning: *Refinement*



Partitioning should be informative or discriminatory of the reality.

Machine Learning: *Proper loss functions decompose into calibration and refinement errors*



Lemma 2.5. *(Decomposition of proper scoring risk into calibration and refinement error).* (Bröcker, 2009). *Given a space $\mathcal{X} \times \mathcal{Y}$ with a distribution P specified on it, a confidence predictor $h : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ whose risk in expectation over P is evaluated by a proper scoring loss $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to [0, M]$, the said risk decomposes into the calibration error term and the refinement term, as below:*

$$\mathbb{E}\left[\ell\left(\mathrm{y}, h\left(\mathbf{x}\right)\right)\right] = \underbrace{\mathbb{E}\left[d_\ell\left(P\left(\mathrm{y} \mid h\left(\mathbf{x}\right)\right), h\left(\mathbf{x}\right)\right)\right]}_{\textit{calibration error}} + \underbrace{\mathbb{E}\left[H_\ell\left(P\left(\mathrm{y} \mid h\left(\mathbf{x}\right)\right)\right)\right]}_{\textit{refinement}}, \tag{2}$$

Machine Learning: *Proper loss functions*

For x, a proper loss function ℓ evaluates the forecast h against the target $\eta = P(Y|x)$

$$L(\eta, h) = \mathbb{E}_{Y \sim \eta}[\ell(Y, h)]$$

Machine Learning: *Proper loss functions*

For x, a proper loss function ℓ evaluates the forecast h against the target $\eta = P(Y|x)$

$$L(\eta, h) = \mathbb{E}_{Y\sim\eta}[\ell(Y, h)]$$

Properiety means: $L(\eta, h) - L(\eta, \eta) \geq 0 \quad \forall \eta, h$

Machine Learning: *Proper loss functions*

For x, a proper loss function ℓ evaluates the forecast h against the target $\eta = P(Y|x)$

$$L(\eta, h) = \mathbb{E}_{Y\sim\eta}[\ell(Y, h)]$$

Properiety means: $L(\eta, h) - L(\eta, \eta) \geq 0 \quad \forall \eta, h$

$$L(\eta, \eta) = \inf_h \mathbb{E}_{Y\sim\eta}[\ell(Y, h)]$$

Machine Learning: *Proper loss functions*

For x, a proper loss function ℓ evaluates the forecast h against the target $\eta = P(Y \mid x)$

$$L(\eta, h) = \mathbb{E}_{Y \sim \eta}[\ell(Y, h)]$$

Properiety means: $L(\eta, h) - L(\eta, \eta) \geq 0 \quad \forall \eta, h$

$$L(\eta, \eta) = \inf_{h} \mathbb{E}_{Y \sim \eta}[\ell(Y, h)]$$

Proper loss functions elicit truthful forecasts.

Machine Learning: *Proper loss functions*

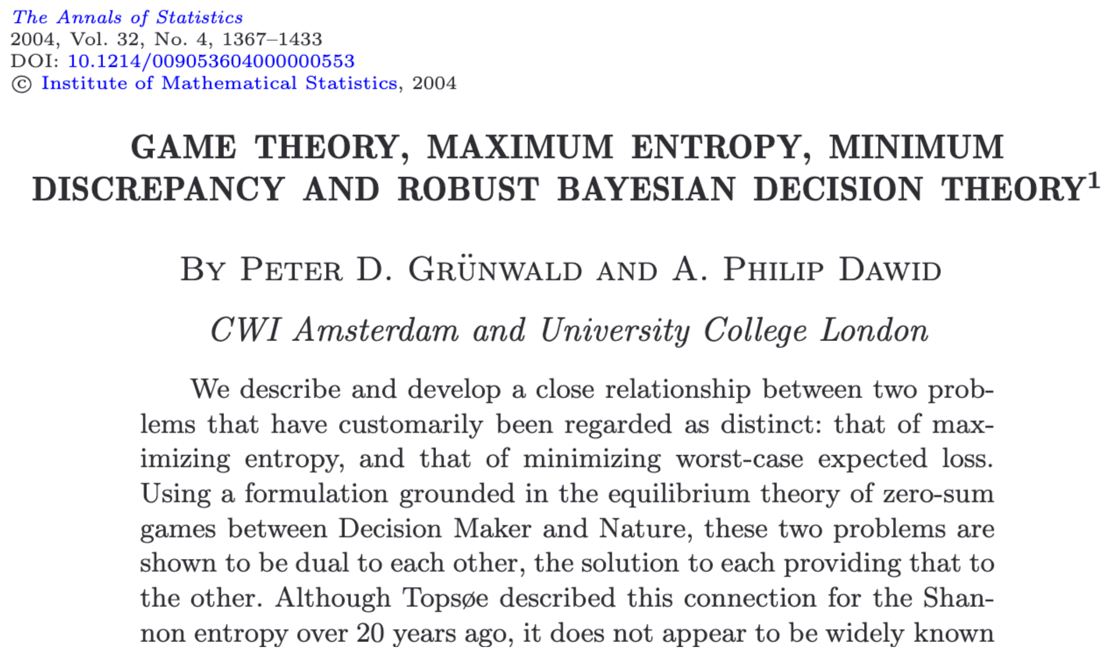
For x, a proper loss function ℓ evaluates the forecast h against the target $\eta = P(Y|x)$

$$L(\eta, h) = \mathbb{E}_{Y\sim\eta}[\ell(Y, h)]$$

$$\boxed{\text{Properiety means:} \quad L(\eta, h) - L(\eta, \eta) \geq 0 \quad \forall \eta, h}$$

$$L(\eta, \eta) = \inf_h \mathbb{E}_{Y\sim\eta}[\ell(Y, h)]$$

$$:= H_\ell(\eta)$$



The Annals of Statistics
2004, Vol. 32, No. 4, 1367–1433
DOI: 10.1214/009053604000000553
© Institute of Mathematical Statistics, 2004

GAME THEORY, MAXIMUM ENTROPY, MINIMUM DISCREPANCY AND ROBUST BAYESIAN DECISION THEORY[1]

BY PETER D. GRÜNWALD AND A. PHILIP DAWID

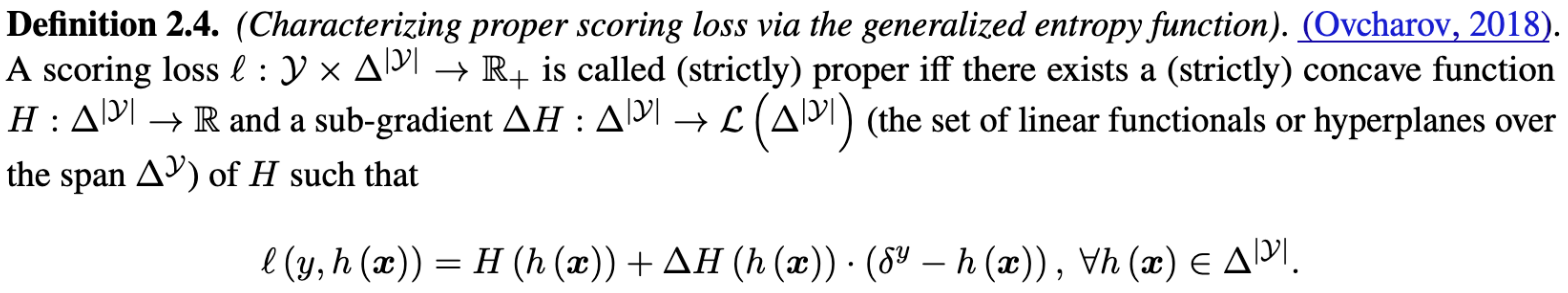
CWI Amsterdam and University College London

We describe and develop a close relationship between two problems that have customarily been regarded as distinct: that of maximizing entropy, and that of minimizing worst-case expected loss. Using a formulation grounded in the equilibrium theory of zero-sum games between Decision Maker and Nature, these two problems are shown to be dual to each other, the solution to each providing that to the other. Although Topsøe described this connection for the Shannon entropy over 20 years ago, it does not appear to be widely known

Generalized entropy function associated with ℓ

Proper loss functions elicit truthful forecasts.

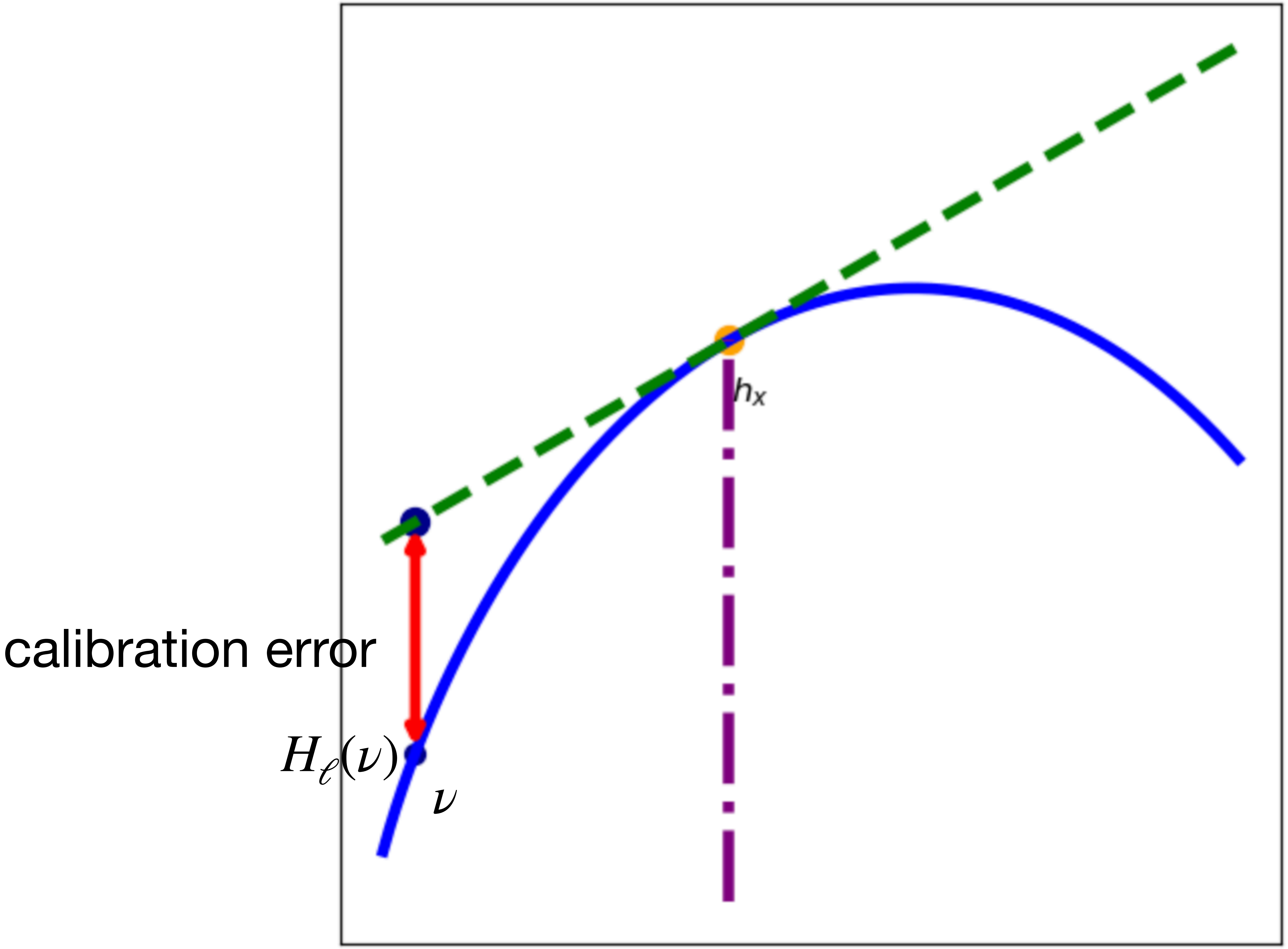
Machine Learning: *Geometry of Proper loss functions*



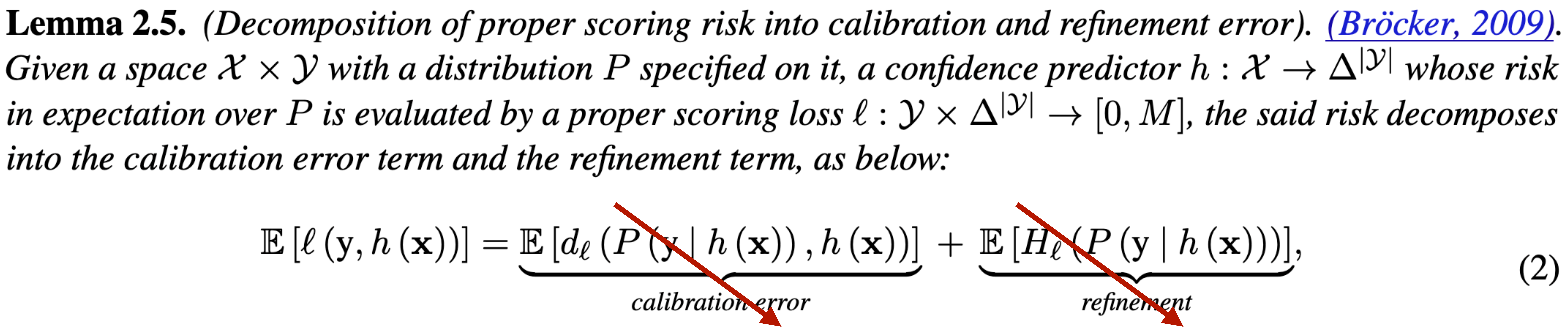
Definition 2.4. *(Characterizing proper scoring loss via the generalized entropy function).* (Ovcharov, 2018). A scoring loss $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to \mathbb{R}_+$ is called (strictly) proper iff there exists a (strictly) concave function $H : \Delta^{|\mathcal{Y}|} \to \mathbb{R}$ and a sub-gradient $\Delta H : \Delta^{|\mathcal{Y}|} \to \mathcal{L}\left(\Delta^{|\mathcal{Y}|}\right)$ (the set of linear functionals or hyperplanes over the span $\Delta^{\mathcal{Y}}$) of H such that

$$\ell\left(y, h\left(\boldsymbol{x}\right)\right) = H\left(h\left(\boldsymbol{x}\right)\right) + \Delta H\left(h\left(\boldsymbol{x}\right)\right) \cdot \left(\delta^{y} - h\left(\boldsymbol{x}\right)\right), \ \forall h\left(\boldsymbol{x}\right) \in \Delta^{|\mathcal{Y}|}.$$

Machine Learning: *Geometry of Proper loss functions*



Machine Learning: *Proper loss functions decompose into calibration and refinement errors*



Lemma 2.5. *(Decomposition of proper scoring risk into calibration and refinement error).* (Bröcker, 2009). *Given a space $\mathcal{X} \times \mathcal{Y}$ with a distribution P specified on it, a confidence predictor $h : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ whose risk in expectation over P is evaluated by a proper scoring loss $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to [0, M]$, the said risk decomposes into the calibration error term and the refinement term, as below:*

$$\mathbb{E}[\ell(\mathbf{y}, h(\mathbf{x}))] = \underbrace{\mathbb{E}[d_\ell(P(\mathbf{y} \mid h(\mathbf{x})), h(\mathbf{x}))]}_{\textit{calibration error}} + \underbrace{\mathbb{E}[H_\ell(P(\mathbf{y} \mid h(\mathbf{x})))]}_{\textit{refinement}}, \tag{2}$$

0

$\mathbb{E}[H_\ell(P(Y|X))]$

Bayes risk

Machine Learning: *Single distribution learning summary*

Learning with proper loss functions has the benchmark as $\mathbb{E}[H_\ell(\eta)]$

Calibrated and well-refined predictors naturally emerge due to loss minimisation.

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

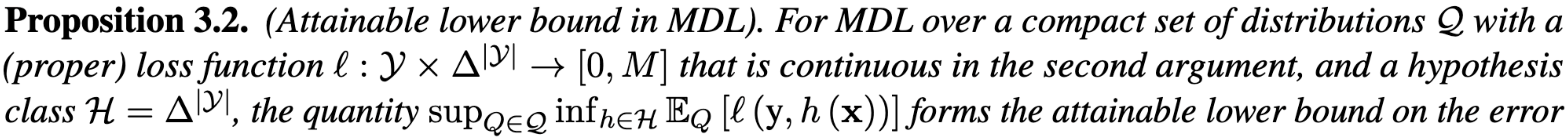
What is the benchmark?

Relation between loss minimisation and calibration?

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

What is the benchmark?

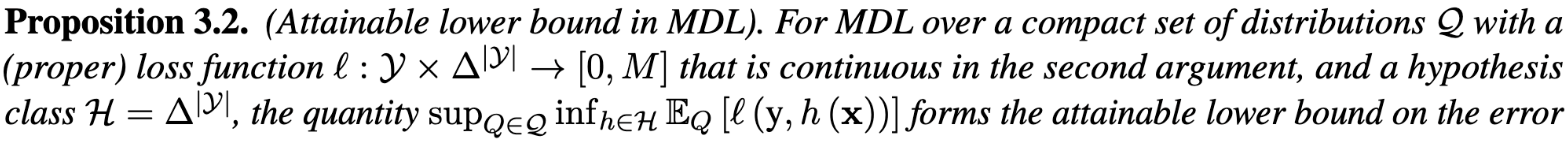


Proposition 3.2. *(Attainable lower bound in MDL). For MDL over a compact set of distributions $\mathcal{Q}$ with a (proper) loss function $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to [0, M]$ that is continuous in the second argument, and a hypothesis class $\mathcal{H} = \Delta^{|\mathcal{Y}|}$, the quantity $\sup_{Q \in \mathcal{Q}} \inf_{h \in \mathcal{H}} \mathbb{E}_Q \left[\ell\left(\mathrm{y}, h\left(\mathbf{x}\right)\right)\right]$ forms the attainable lower bound on the error*

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

What is the benchmark?



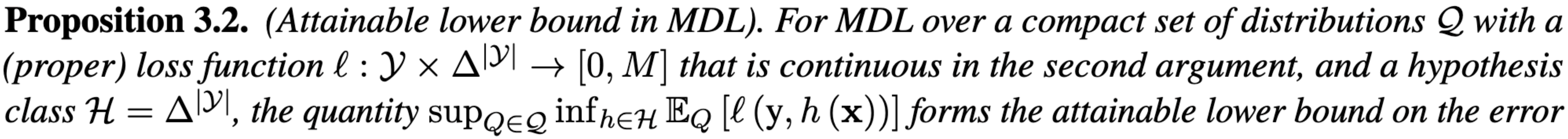
Proposition 3.2. *(Attainable lower bound in MDL). For MDL over a compact set of distributions $\mathcal{Q}$ with a (proper) loss function $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to [0, M]$ that is continuous in the second argument, and a hypothesis class $\mathcal{H} = \Delta^{|\mathcal{Y}|}$, the quantity $\sup_{Q\in\mathcal{Q}} \inf_{h\in\mathcal{H}} \mathbb{E}_Q\left[\ell\left(\mathrm{y}, h\left(\mathbf{x}\right)\right)\right]$ forms the attainable lower bound on the error*

$$\sup_{Q\in\mathcal{Q}} \inf_{h\in\mathscr{H}} \mathbb{E}_Q\left[\ell\left(Y, h(X)\right)\right]$$

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

What is the benchmark?



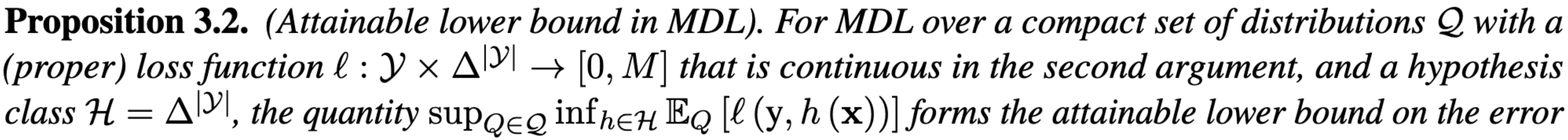
Proposition 3.2. *(Attainable lower bound in MDL). For MDL over a compact set of distributions $\mathcal{Q}$ with a (proper) loss function $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to [0, M]$ that is continuous in the second argument, and a hypothesis class $\mathcal{H} = \Delta^{|\mathcal{Y}|}$, the quantity $\sup_{Q\in\mathcal{Q}} \inf_{h\in\mathcal{H}} \mathbb{E}_Q\left[\ell\left(\mathrm{y}, h\left(\mathbf{x}\right)\right)\right]$ forms the attainable lower bound on the error*

$$\sup_{Q\in\mathcal{Q}} \underbrace{\inf_{h\in\mathscr{H}} \mathbb{E}_Q\left[\ell\left(Y, h(X)\right)\right]}_{\mathbb{E}[H_\ell(Q(Y|X))]}$$

Machine Learning: *Multi-distribution learning*

Given a set $\mathscr{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathscr{Q}$.

What is the benchmark?



Proposition 3.2. *(Attainable lower bound in MDL). For MDL over a compact set of distributions $\mathcal{Q}$ with a (proper) loss function $\ell : \mathcal{Y} \times \Delta^{|\mathcal{Y}|} \to [0, M]$ that is continuous in the second argument, and a hypothesis class $\mathcal{H} = \Delta^{|\mathcal{Y}|}$, the quantity $\sup_{Q\in\mathcal{Q}} \inf_{h\in\mathcal{H}} \mathbb{E}_Q \left[\ell\left(\mathrm{y}, h\left(\mathbf{x}\right)\right)\right]$ forms the attainable lower bound on the error*

$$\sup_{Q\in Q} \inf_{h\in\mathscr{H}} \mathbb{E}_Q \left[\ell\left(Y, h(X)\right)\right]$$

$$= \sup_{Q\in\mathscr{Q}} \mathbb{E}[H_\ell(Q(Y\,|\,X))]$$

maximum expected generalized entropy over $\mathscr{Q}$

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

$$\sup_{Q \in Q} \inf_{h \in \mathscr{H}} \mathbb{E}_Q \left[\ell \left(Y, h(X) \right) \right]$$

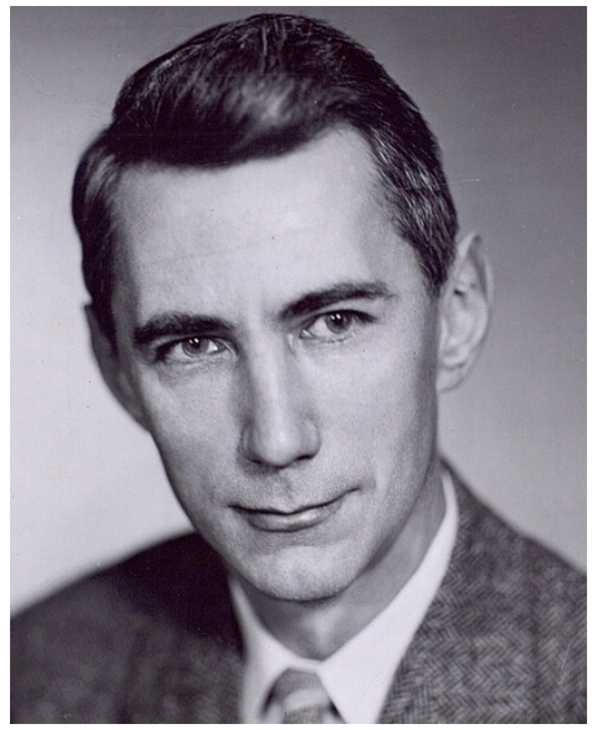
$$= \sup_{Q \in \mathcal{Q}} \mathbb{E}[H_\ell(Q(Y \mid X))]$$

maximum generalized expected entropy over $\mathcal{Q}$

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.



Shannon's source coding theorem.

$$\sup_{Q \in Q} \inf_{h \in \mathscr{H}} \mathbb{E}_Q \left[\ell \left(Y, h(X) \right) \right]$$

$$= \sup_{Q \in \mathcal{Q}} \mathbb{E}[H_\ell(Q(Y|X))]$$

maximum generalized expected entropy over $\mathcal{Q}$

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

Could this lower bound be attained in practice?

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

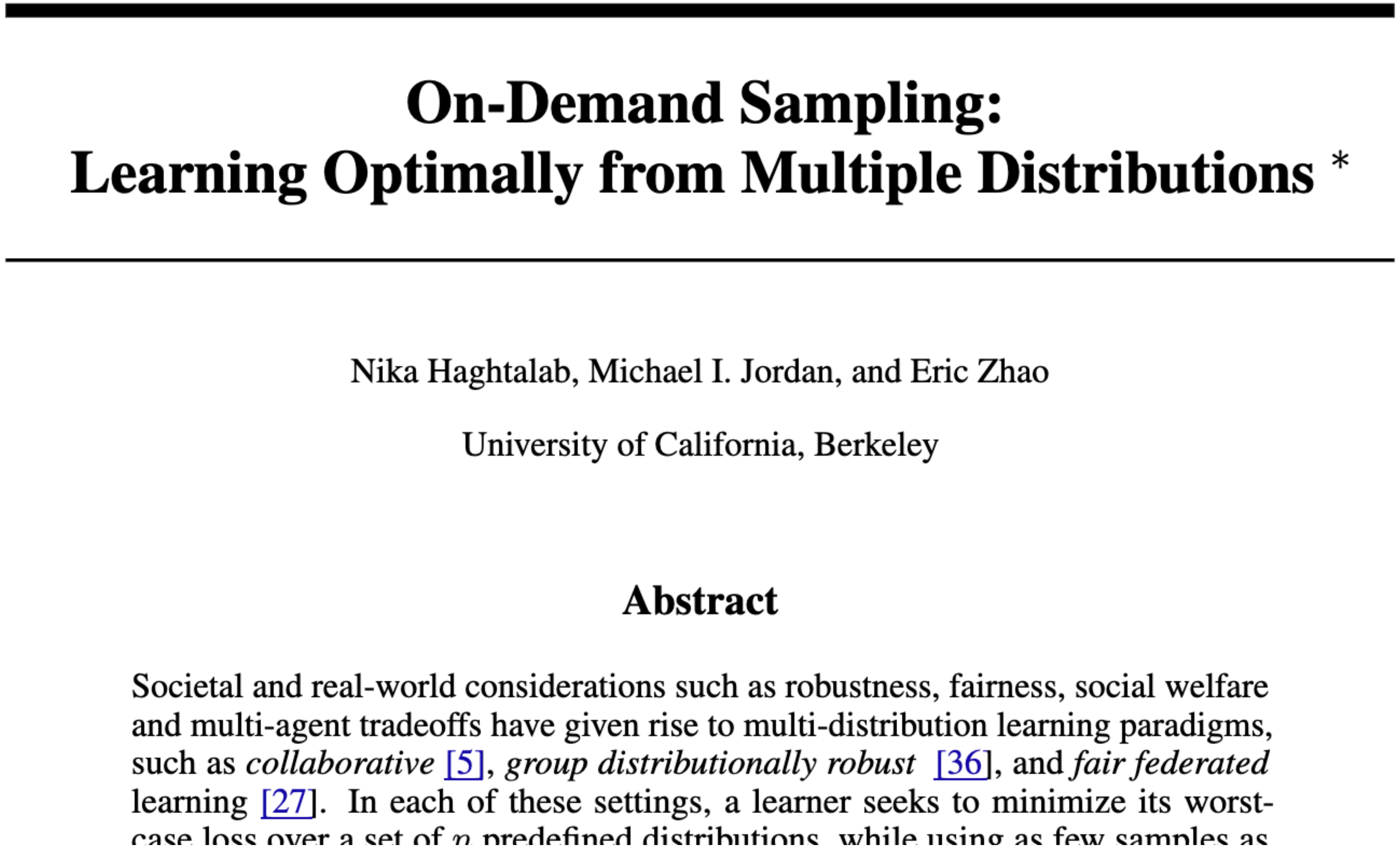
Could this lower bound be attained in practice?

Zero-sum games and Nash equilibrium

Machine Learning: *Multi-distribution learning*

Could this lower bound be attained in practice?

Zero-sum games and Nash equilibrium



On-Demand Sampling: Learning Optimally from Multiple Distributions *

Nika Haghtalab, Michael I. Jordan, and Eric Zhao

University of California, Berkeley

Abstract

Societal and real-world considerations such as robustness, fairness, social welfare and multi-agent tradeoffs have given rise to multi-distribution learning paradigms, such as *collaborative* [5], *group distributionally robust* [36], and *fair federated* learning [27]. In each of these settings, a learner seeks to minimize its worst-case loss over a set of n predefined distributions, while using as few samples as

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

What is the benchmark?

Relation between loss minimisation and calibration?

Machine Learning: *Multi-distribution learning*

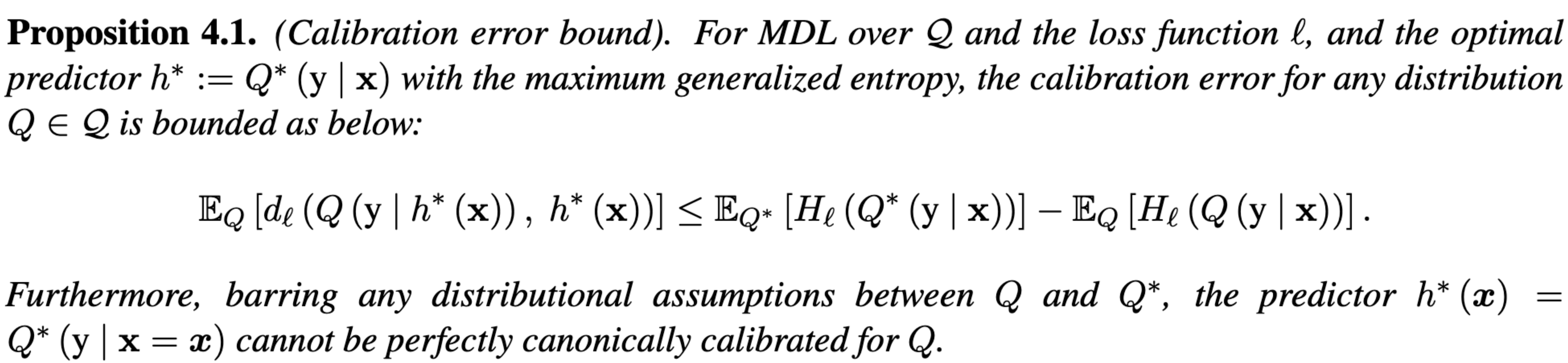
Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

Relation between loss minimisation and calibration?

Machine Learning: *Multi-distribution learning*

Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

Relation between loss minimisation and calibration?



Proposition 4.1. *(Calibration error bound). For MDL over $\mathcal{Q}$ and the loss function ℓ, and the optimal predictor $h^* := Q^*(\mathrm{y} \mid \mathbf{x})$ with the maximum generalized entropy, the calibration error for any distribution $Q \in \mathcal{Q}$ is bounded as below:*

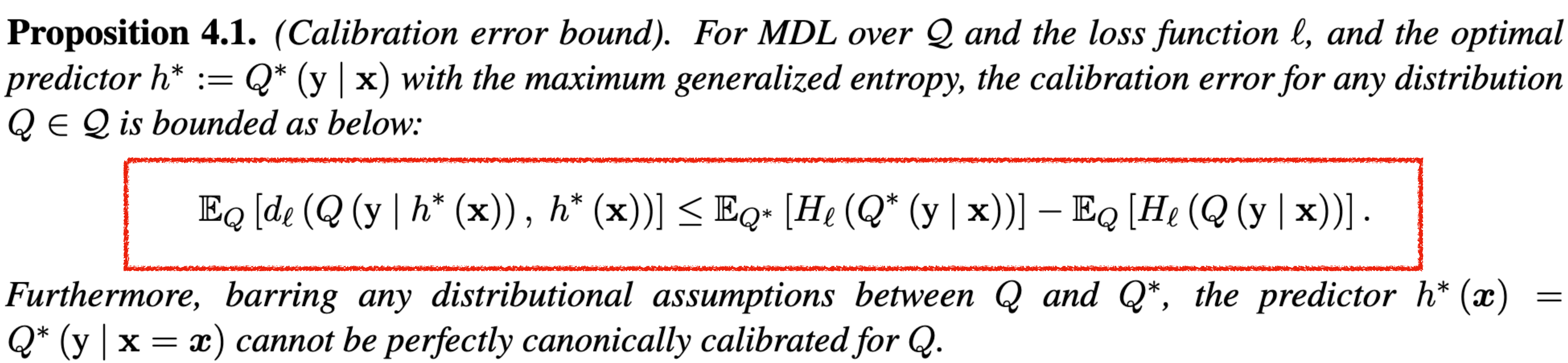
$$\mathbb{E}_Q \left[d_\ell \left(Q\left(\mathrm{y} \mid h^*(\mathbf{x})\right), \; h^*(\mathbf{x}) \right) \right] \leq \mathbb{E}_{Q^*} \left[H_\ell \left(Q^*(\mathrm{y} \mid \mathbf{x}) \right) \right] - \mathbb{E}_Q \left[H_\ell \left(Q(\mathrm{y} \mid \mathbf{x}) \right) \right].$$

Furthermore, barring any distributional assumptions between Q and Q^*, the predictor $h^*(\boldsymbol{x}) = Q^*(\mathrm{y} \mid \mathbf{x} = \boldsymbol{x})$ cannot be perfectly canonically calibrated for Q.

Machine Learning: *Multi-distribution learning*

Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

Relation between loss minimisation and calibration?



Proposition 4.1. *(Calibration error bound). For MDL over $\mathcal{Q}$ and the loss function ℓ, and the optimal predictor $h^* := Q^*(\mathrm{y} \mid \mathbf{x})$ with the maximum generalized entropy, the calibration error for any distribution $Q \in \mathcal{Q}$ is bounded as below:*

$$\boxed{\mathbb{E}_Q\left[d_\ell\left(Q\left(\mathrm{y} \mid h^*(\mathbf{x})\right), \, h^*(\mathbf{x})\right)\right] \le \mathbb{E}_{Q^*}\left[H_\ell\left(Q^*(\mathrm{y} \mid \mathbf{x})\right)\right] - \mathbb{E}_Q\left[H_\ell\left(Q(\mathrm{y} \mid \mathbf{x})\right)\right].}$$

Furthermore, barring any distributional assumptions between Q and Q^*, the predictor $h^*(\boldsymbol{x}) = Q^*(\mathrm{y} \mid \mathbf{x} = \boldsymbol{x})$ cannot be perfectly canonically calibrated for Q.

Calibration error bounded above by expected generalized entropy difference.

Machine Learning: *Multi-distribution learning*

Relation between loss minimisation and calibration?

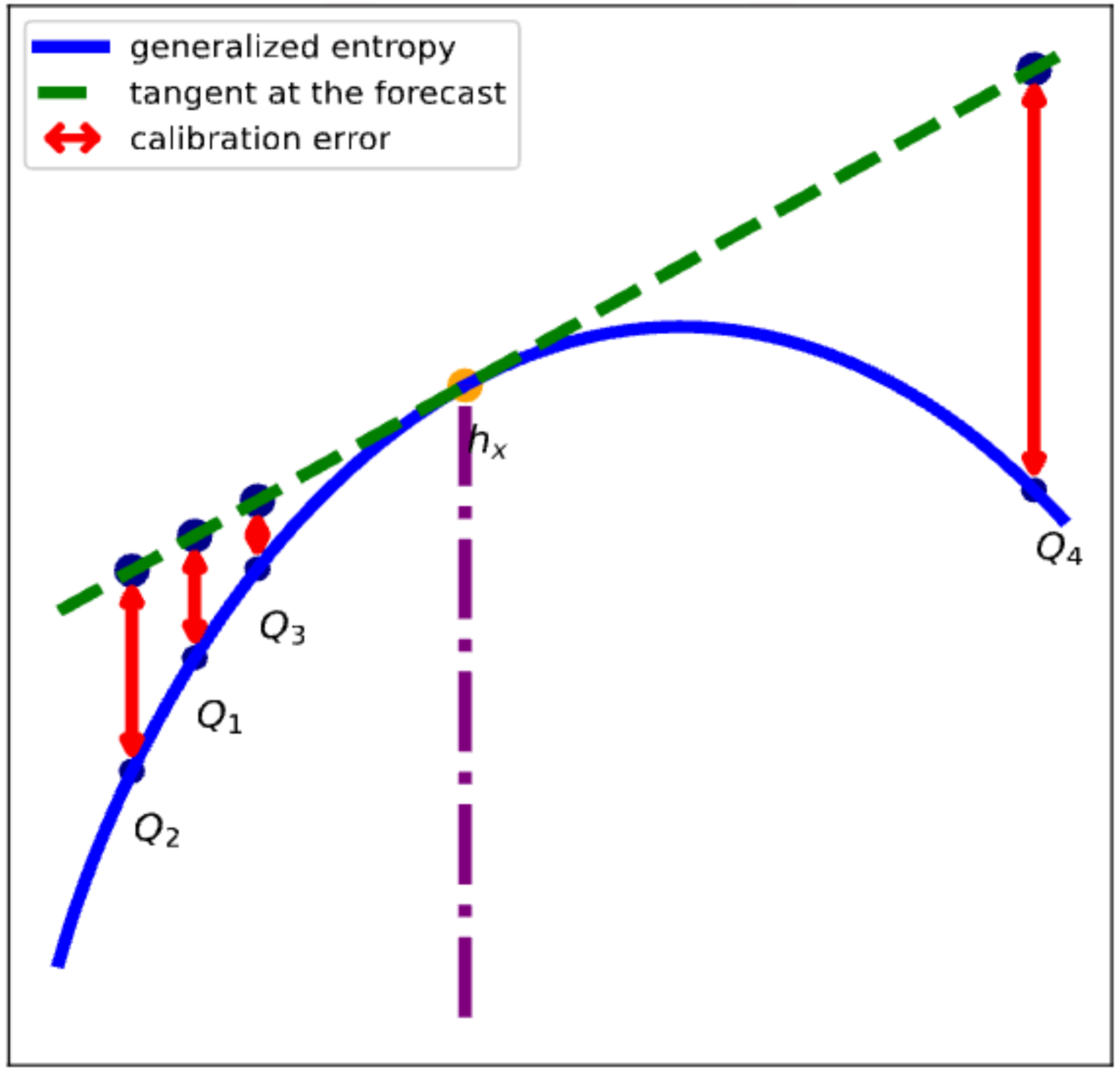
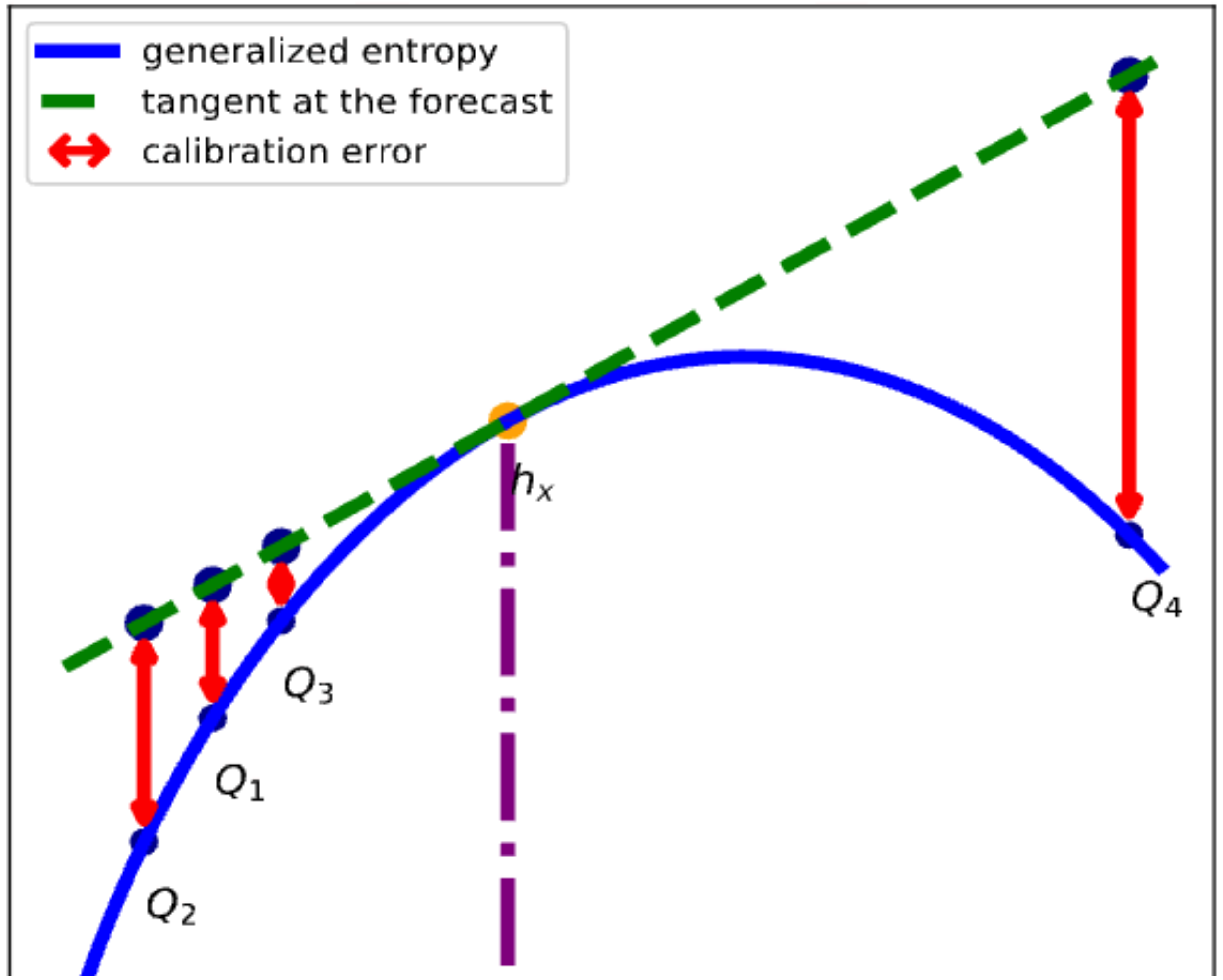


Figure 1: *Calibration disparity intuition in MDL:*

Machine Learning: *Multi-distribution learning*

Relation between loss minimisation and calibration?

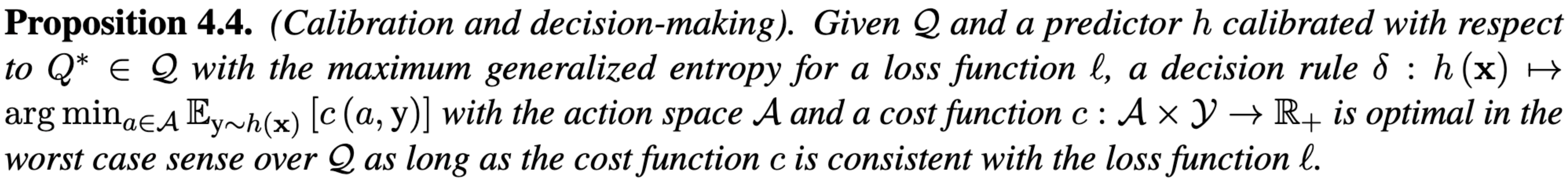


Corollary 4.3. *There is a fundamental calibration-refinement trade-off in MDL, even at optimality. Furthermore, a prediction has different meaning for different distributions.*

Figure 1: *Calibration disparity intuition in MDL:*

Machine Learning: *Multi-distribution learning and Institutional Separation*

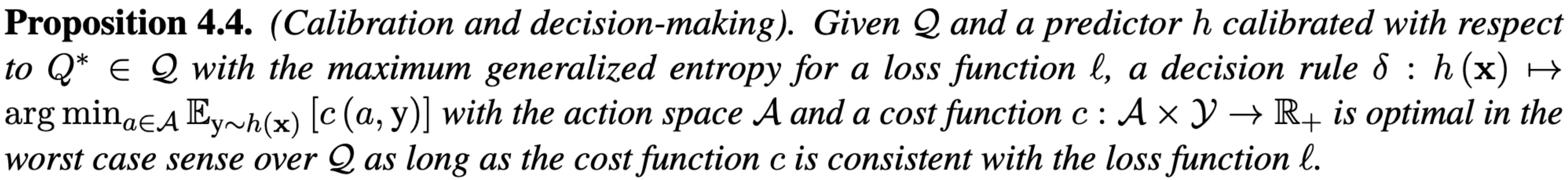
Relation between loss minimisation and calibration?



Proposition 4.4. *(Calibration and decision-making). Given $\mathcal{Q}$ and a predictor h calibrated with respect to $Q^* \in \mathcal{Q}$ with the maximum generalized entropy for a loss function ℓ, a decision rule $\delta : h(\mathbf{x}) \mapsto \arg\min_{a \in \mathcal{A}} \mathbb{E}_{\mathrm{y} \sim h(\mathbf{x})} \left[c(a, \mathrm{y}) \right]$ with the action space $\mathcal{A}$ and a cost function $c : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}_+$ is optimal in the worst case sense over $\mathcal{Q}$ as long as the cost function c is consistent with the loss function ℓ.*

Machine Learning: *Multi-distribution learning and institutional separation*

Relation between loss minimisation and calibration?



Proposition 4.4. *(Calibration and decision-making). Given $\mathcal{Q}$ and a predictor h calibrated with respect to $Q^* \in \mathcal{Q}$ with the maximum generalized entropy for a loss function ℓ, a decision rule $\delta : h(\mathbf{x}) \mapsto \arg\min_{a \in \mathcal{A}} \mathbb{E}_{\mathrm{y} \sim h(\mathbf{x})} [c(a, \mathrm{y})]$ with the action space $\mathcal{A}$ and a cost function $c : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}_+$ is optimal in the worst case sense over $\mathcal{Q}$ as long as the cost function c is consistent with the loss function ℓ.*

Contrary to calibration in the single distribution case, calibration in MDL bridges institutional separation for a limited class of loss functions.

Machine Learning: *Multi-distribution learning*

Given a set $\mathcal{Q}$ of possible distributions one can sample from, design a predictor that works well for all $Q \in \mathcal{Q}$.

What is the benchmark?

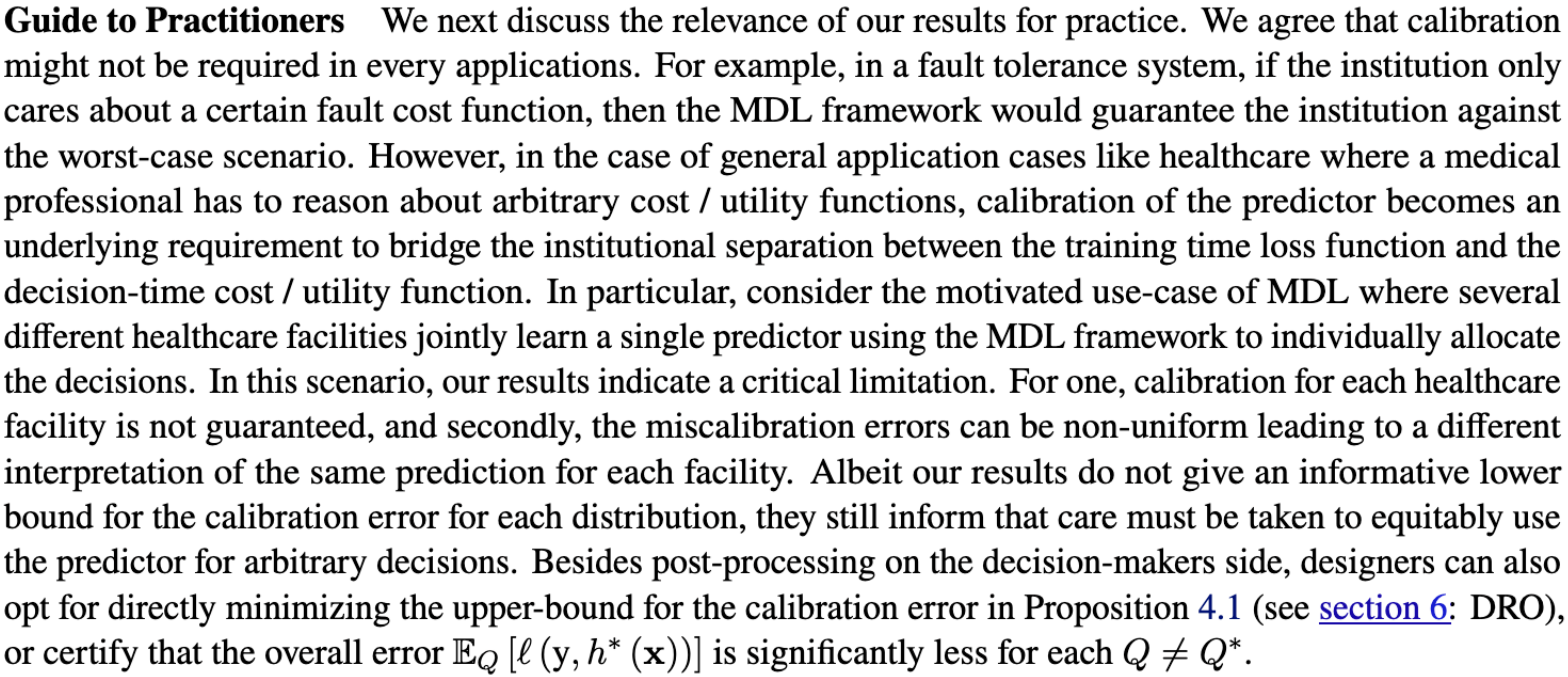
Takeaway 1: In MDL, learning with respect to the distribution with the maximum (generalized) entropy associated with the considered loss function gives the attainable lower bound for MDL.

Relation between loss minimisation and calibration?

Takeaway 2: We have discovered a fundamental calibration-refinement trade-off in the MDL framework. This trade-off is determined by the heterogeneity in terms of the (generalized) entropy in the considered envelope of distributions. Furthermore, learning (optimally) over multiple distributions also does not truly bridge the institutional separation between model designers and decision-makers. A decision-maker is always constrained by the types of cost functions they can consider when exploiting MDL.

Machine Learning: *Multi-distribution learning*

Main takeaway:



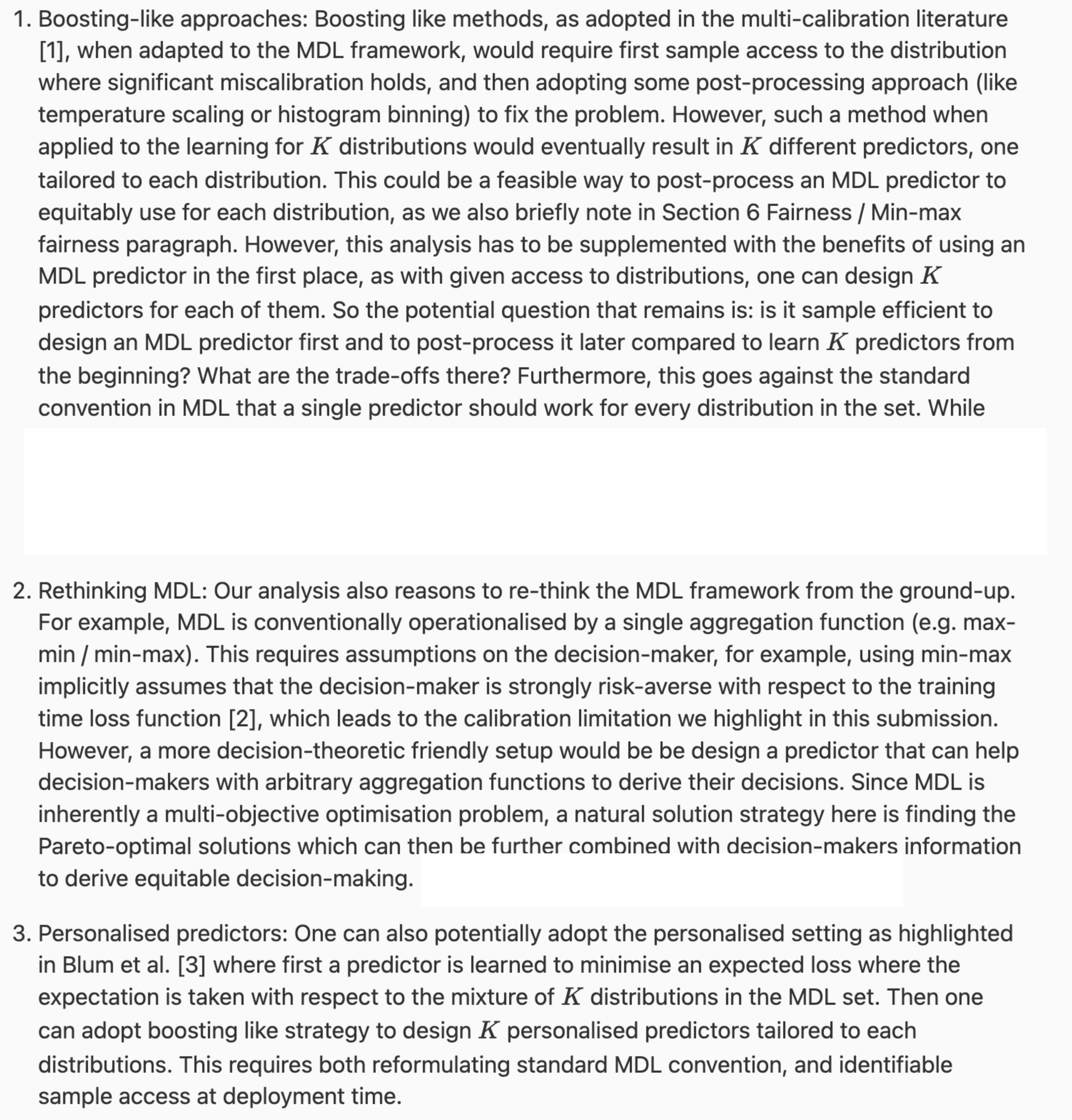
Guide to Practitioners We next discuss the relevance of our results for practice. We agree that calibration might not be required in every applications. For example, in a fault tolerance system, if the institution only cares about a certain fault cost function, then the MDL framework would guarantee the institution against the worst-case scenario. However, in the case of general application cases like healthcare where a medical professional has to reason about arbitrary cost / utility functions, calibration of the predictor becomes an underlying requirement to bridge the institutional separation between the training time loss function and the decision-time cost / utility function. In particular, consider the motivated use-case of MDL where several different healthcare facilities jointly learn a single predictor using the MDL framework to individually allocate the decisions. In this scenario, our results indicate a critical limitation. For one, calibration for each healthcare facility is not guaranteed, and secondly, the miscalibration errors can be non-uniform leading to a different interpretation of the same prediction for each facility. Albeit our results do not give an informative lower bound for the calibration error for each distribution, they still inform that care must be taken to equitably use the predictor for arbitrary decisions. Besides post-processing on the decision-makers side, designers can also opt for directly minimizing the upper-bound for the calibration error in Proposition 4.1 (see section 6: DRO), or certify that the overall error $\mathbb{E}_Q\left[\ell\left(\mathrm{y}, h^*\left(\mathbf{x}\right)\right)\right]$ is significantly less for each $Q \neq Q^*$.

Machine Learning: *Multi-distribution learning*

Main takeaway:

1. Calibration is useful for decision-making, but is not required in every application.
2. MDL guarantees one against the worst-case scenario. For example, if some institution only cares about certain cost function, like a fault tolerance loss, then MDL gives error control.
3. However, for general applications like healthcare, there is a critical limitation.

Machine Learning: *Multi-distribution learning (Future work)*



1. Boosting-like approaches: Boosting like methods, as adopted in the multi-calibration literature [1], when adapted to the MDL framework, would require first sample access to the distribution where significant miscalibration holds, and then adopting some post-processing approach (like temperature scaling or histogram binning) to fix the problem. However, such a method when applied to the learning for K distributions would eventually result in K different predictors, one tailored to each distribution. This could be a feasible way to post-process an MDL predictor to equitably use for each distribution, as we also briefly note in Section 6 Fairness / Min-max fairness paragraph. However, this analysis has to be supplemented with the benefits of using an MDL predictor in the first place, as with given access to distributions, one can design K predictors for each of them. So the potential question that remains is: is it sample efficient to design an MDL predictor first and to post-process it later compared to learn K predictors from the beginning? What are the trade-offs there? Furthermore, this goes against the standard convention in MDL that a single predictor should work for every distribution in the set. While

2. Rethinking MDL: Our analysis also reasons to re-think the MDL framework from the ground-up. For example, MDL is conventionally operationalised by a single aggregation function (e.g. max-min / min-max). This requires assumptions on the decision-maker, for example, using min-max implicitly assumes that the decision-maker is strongly risk-averse with respect to the training time loss function [2], which leads to the calibration limitation we highlight in this submission. However, a more decision-theoretic friendly setup would be be design a predictor that can help decision-makers with arbitrary aggregation functions to derive their decisions. Since MDL is inherently a multi-objective optimisation problem, a natural solution strategy here is finding the Pareto-optimal solutions which can then be further combined with decision-makers information to derive equitable decision-making.

3. Personalised predictors: One can also potentially adopt the personalised setting as highlighted in Blum et al. [3] where first a predictor is learned to minimise an expected loss where the expectation is taken with respect to the mixture of K distributions in the MDL set. Then one can adopt boosting like strategy to design K personalised predictors tailored to each distributions. This requires both reformulating standard MDL convention, and identifiable sample access at deployment time.

Boosting like approaches to fix calibration disparities.

Rethinking MDL

Personalised predictors

On Calibration in Multi-Distribution Learning

Rajeev Verma *r.verma@uva.nl*
UvA-Bosch Delta Lab
University of Amsterdam

Volker Fischer *volker.fischer@de.bosch.com*
Bosch Center for Artificial Intelligence
Renningen, Germany

Eric Nalisnick *nalisnick@jhu.edu*
Department of Computer Science
Johns Hopkins University

Abstract

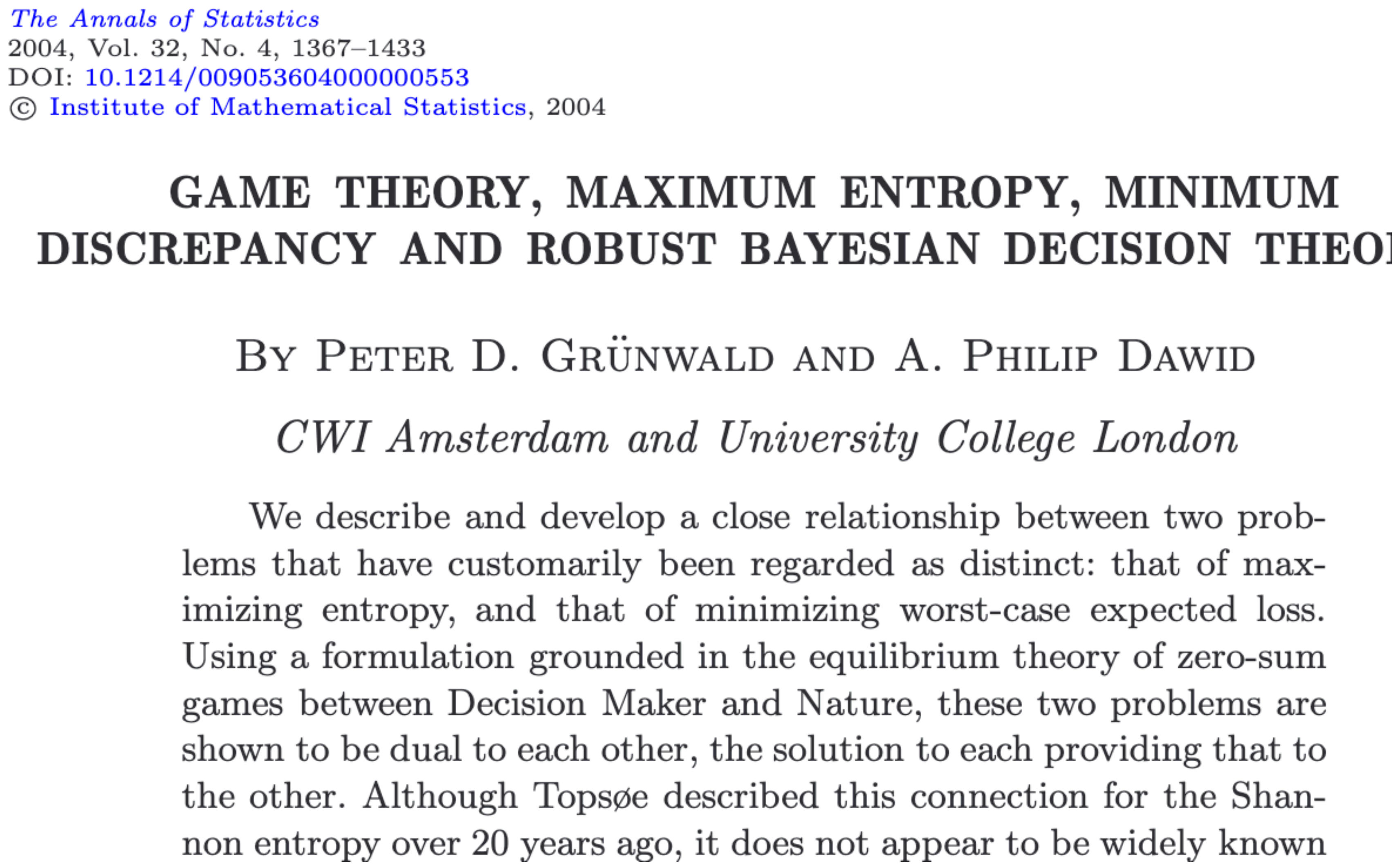
Modern challenges of robustness, fairness, and decision-making in machine learning have led to the formulation of *multi-distribution learning* (MDL) frameworks in which a predictor is optimized across multiple distributions. We study the calibration properties of MDL to better understand how the predictor performs uniformly across the multiple distributions. Through classical results on decomposing proper scoring losses, we first derive the Bayes optimal rule for MDL, demonstrating that it maximizes the generalized entropy of the associated loss function. Our analysis reveals that while this approach ensures minimal worst-case loss, it can lead to non-uniform calibration errors across the multiple distributions and there is an

Acknowledgements:

Peter Grünwald

Nika Haghtalab

Christian Fröhlich



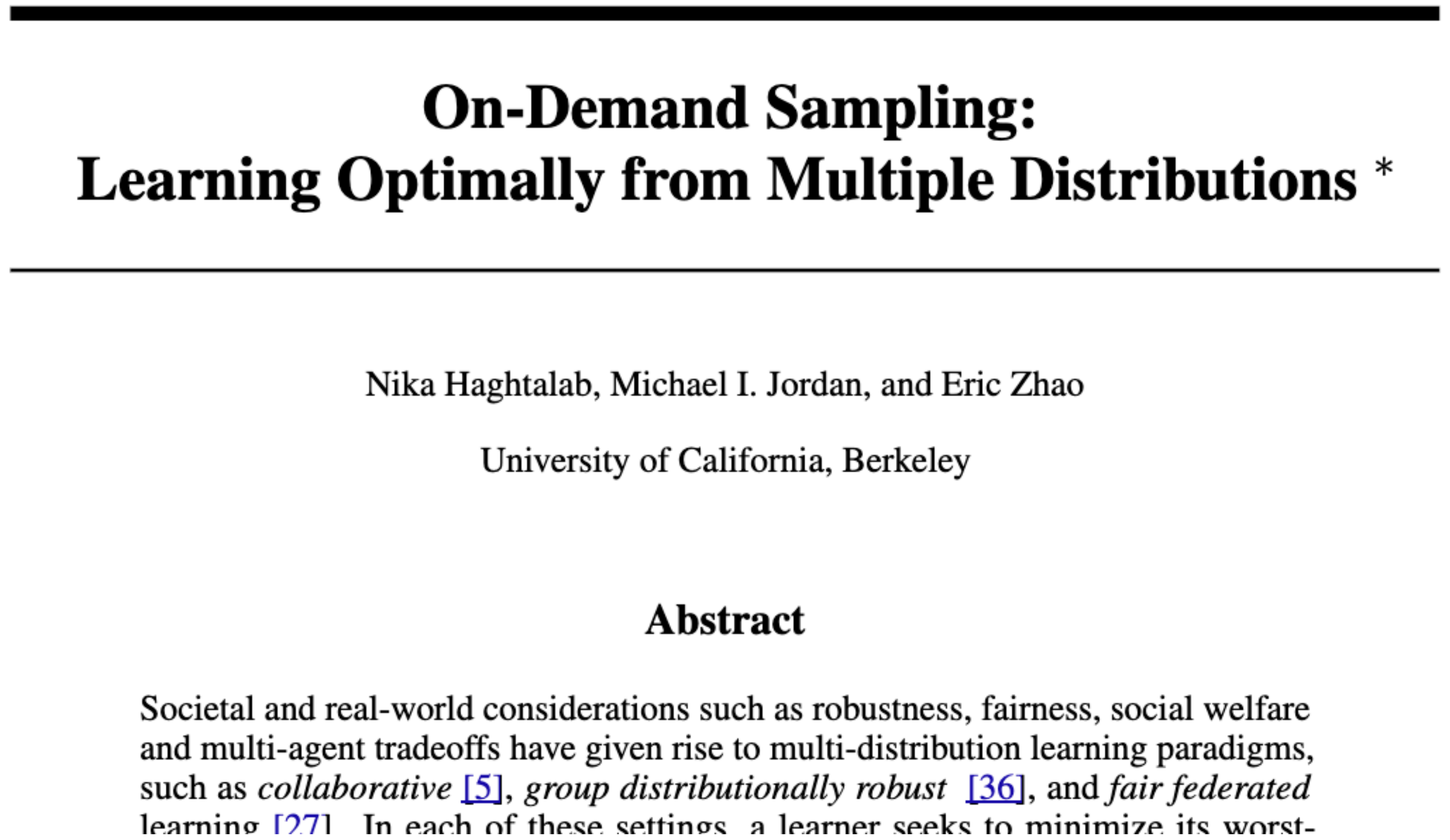
The Annals of Statistics
2004, Vol. 32, No. 4, 1367–1433
DOI: 10.1214/009053604000000553
© Institute of Mathematical Statistics, 2004

GAME THEORY, MAXIMUM ENTROPY, MINIMUM DISCREPANCY AND ROBUST BAYESIAN DECISION THEO

BY PETER D. GRÜNWALD AND A. PHILIP DAWID

CWI Amsterdam and University College London

We describe and develop a close relationship between two problems that have customarily been regarded as distinct: that of maximizing entropy, and that of minimizing worst-case expected loss. Using a formulation grounded in the equilibrium theory of zero-sum games between Decision Maker and Nature, these two problems are shown to be dual to each other, the solution to each providing that to the other. Although Topsøe described this connection for the Shannon entropy over 20 years ago, it does not appear to be widely known



On-Demand Sampling:
Learning Optimally from Multiple Distributions *

Nika Haghtalab, Michael I. Jordan, and Eric Zhao

University of California, Berkeley

Abstract

Societal and real-world considerations such as robustness, fairness, social welfare and multi-agent tradeoffs have given rise to multi-distribution learning paradigms, such as *collaborative* [5], *group distributionally robust* [36], and *fair federated* learning [27]. In each of these settings, a learner seeks to minimize its worst-