J. Econ. Theory 224 (2025) 105970

Contents lists available at ScienceDirect

Journal of Economic Theory

journal homepage: www.elsevier.com/locate/jet

Modeling machine learning: A cognitive economic approach ☆

Andrew Caplin [a], Daniel Martin [b,*], Philip Marx [c]

[a] Department of Economics, New York University, United States of America
[b] Department of Economics, University of California, Santa Barbara, United States of America
[c] Department of Economics, Louisiana State University, United States of America

ARTICLE INFO

JEL classification:
D83
D91

Keywords:
Algorithms
Artificial intelligence
Machine learning
Information frictions
Information economics
Rational inattention

ABSTRACT

We investigate whether the predictions of modern machine learning algorithms are consistent with economic models of human cognition. To test these models we run an experiment in which we vary the loss function used in training a leading deep learning convolutional neural network to predict pneumonia from chest X-rays. The first cognitive economic model we test, capacity-constrained learning, corresponds with an intuitive notion of machine learning: that an algorithm chooses among a feasible set of learning strategies in order to minimize the loss function used in training. Our experiment shows systematic deviations from the testable implications of this model. Instead, we find that changes in the loss function impact learning just as they might if the algorithm was a human being who found learning costly.

1. Introduction

Machine learning is increasingly central to the modern economy. Virtually all industries, jobs, and consumer experiences have been impacted in some way by the rapid rise of this technology. Economically important applications of machine learning include facial recognition, language translation, credit scoring and loan default predictions, medical diagnoses, product recommendations, driving routes, fraud alerts, and so on.

In this paper, we test whether the predictions generated by a leading machine learning algorithm are consistent with standard economic models of human cognition. This is a natural class of models to investigate because they are built on a mechanistic information-theoretic foundation. In these models, individual decision makers engage in gathering signals, updating beliefs according to Bayes rule, and maximizing expected utility.

The first model we test is *capacity-constrained learning*, characterized by Caplin et al. (2024), which generalizes the fixed capacity version of rational inattention theory proposed by Sims (2003) and the noisy cognition model proposed by Woodford (2014). With this model, the decision-maker chooses among a feasible set of ways to learn about observations. Importantly, capacity-constrained learning aligns with an intuitive notion of machine learning, which is that a machine learning algorithm learns by choosing from a feasible set of mathematical operations to best match the loss function used to train the algorithm.

☆ We thank audiences at the Sloan-NOMIS Summer School on Cognitive Foundations of Economic Behavior, ESSET in Gerzensee, ESIF AI+ML, BRIC, D-TEA, SAET, Maryland, USC, UNR, HKBU, Wharton, UBC, Colmex, UCSB, Northwestern, and Zurich; as well as the editor, the associate editor, and two anonymous referees for many helpful suggestions. We also thank the Sloan Foundation for support under the "Cognitive Economics at Work" grant, and Caplin thanks the NOMIS and Sloan Foundations more generally for their support for research on the cognitive foundations of economic behavior. We are also grateful to the LSU High Performance Computing and Northwestern Research Computing Services for providing computational resources.

* Corresponding author.
E-mail address: daniel@martinonline.org (D. Martin).

https://doi.org/10.1016/j.jet.2025.105970
Received 26 June 2024; Received in revised form 9 January 2025; Accepted 11 January 2025

Available online 14 January 2025

The second model of human cognition we test is *costly learning*, characterized by Caplin and Dean (2015), which itself generalizes the specialized version using Shannon entropy characterized by Matejka and McKay (2015) and Caplin et al. (2017). With this model, the decision-maker adjusts their learning in response to the relative costs of different ways of learning.[1] In the context of machine learning, we imagine that the algorithm chooses the mathematical operations that best balance losses and costs. We call these costs the algorithm's *pseudo-costs* because they may not have any relationship with any real resource costs incurred while running the algorithm. Instead, these are costs that an algorithm implicitly assigns to different ways of learning.

When testing these models with humans, it is typical to treat both the decision-maker's utility and learning costs as unobservable to the researcher. However, in the case of machine learning, we treat utility as observable (the loss function used to train the algorithm) and the algorithm's pseudo-costs as unobservable. Thus, our test of the costly learning model links to an active literature in computer science on implicit regularization (e.g., Neyshabur et al., 2015; Gunasekar et al., 2017; Arora et al., 2019; Barrett and Dherin, 2021). To explain why deep learning models generalize well even without explicit regularization, this literature argues that algorithms implicitly implement some regularization in optimization and seeks to understand these implicit incentives. In other words, they posit that an algorithm selects a model as if there is an extra term in the training loss function that penalizes model complexity. This implicit extra term in the training problem is analogous to having an implicit extra pseudo-cost in the learning problem.

To test capacity-constrained learning and costly learning in a machine learning context, we run an experiment with CheXNeXt, an influential deep learning convolutional neural network for predicting thoracic diseases from chest X-ray images (Rajpurkar et al., 2018). In addition to being widely-adopted in the field, this algorithm has also been used in studying joint human and AI decision-making. For example, a variant of CheXNeXt is leveraged by Agarwal et al. (2023) to study how expert radiologists make decisions when aided by artificial intelligence (AI) recommendations. In addition, Alur et al. (2024) use a variant of the CheXNeXt algorithm to assess ways of combining human decisions with AI-generated ones.

Our experimental intervention is to vary the algorithm's loss function, which we do by varying its class weights. These dictate the relative value a loss function places on correct and incorrect predictions across different class labels. In a binary setting, these weights dictate the benefits and costs of avoiding Type I and Type II errors. Class weighting is frequently used in machine learning when one class is less common (Thai-Nghe et al., 2010) or with the hope of achieving some external objective (Zadrozny et al., 2003). Rajpurkar et al. (2018) follow standard practice by choosing class weights that increase the value that the algorithm places on mistakes made for observations in the underrepresented class, which is important given that only 1.3% of chest X-rays are in fact labeled with pneumonia.

We first show that CheXNeXt predictions in this experiment are consistent with Bayesian updating and expected loss minimization, key assumptions that underlie both of the models we are testing.[2] It is consistent with these assumptions because it distorts its predictions in line with the incentives for making different predictions. This result can have important downstream consequences. For example, if we use the highly asymmetric class weights implemented by Rajpurkar et al. (2018), then the confidence scores reported by the algorithm are extremely misleading about the likelihood of pneumonia, which necessitates ex-post recalibration methods.

Next, we show that CheXNeXt predictions are *not* consistent with capacity-constrained learning. Capacity-constrained learning requires that lower losses could not have been achieved by training an algorithm with a different loss function than the one that it was trained with. This is because the alternative ways of learning produced by training with other loss functions were in principle feasible and thus could have been selected. In our experiment, we find a systematic deviation from this requirement of capacity-constrained learning, as losses would always be minimized by training with a lower weight on pneumonia instances.

However, while CheXNeXt predictions are not consistent with capacity-constrained learning, we show that they *are* consistent with costly learning. Costly learning makes a related prediction to capacity-constrained learning, but allows for ways of learning not to be selected, even when they generate lower losses, because they are viewed as too costly. In our experiment, while losses would always be minimized by training with lower weight on pneumonia instances, this can be rationalized with recourse to additional, unobserved pseudo-costs. Namely, by placing relatively lower weight on pneumonia instances the algorithm acts as if it is worth incurring higher learning costs, which is sensible given that doing so places relatively higher weight on non-pneumonia instances, which are far more common.

Our paper provides a natural join between three growing literatures: machine learning, cognitive economics, and rational inattention. Cognitive economics integrates economic and psychological research methods to better understand how human cognitive limits, such as on perception or information processing abilities, impact key decisions (Caplin, 2024). For instance, cognitive imprecision and efficient coding, long central in psychology, have been shown to underlie many of the heuristics and biases that have featured prominently in behavioral economics (Woodford, 2014). Sims (2003) introduced entropy-based limits to cognition in the context of sluggish and delayed responses to macroeconomic policy, and this launched a large subsequent literature on rational inattention that highlights the costs of cognition. In addition to new modeling approaches, psychological methods of measurement have increasingly been adopted by economists and have themselves inspired modeling advances. Included are decision times (Alós-Ferrer et al., 2021) and patterns in psychometric "state-dependent stochastic choice" (SDSC) data that records patterns of choice conditional on underlying facts about the world (Caplin and Martin, 2015).

[1] The first model is a special case of second because the feasible set can be represented by a cost function that is zero for feasible learning strategies and infinite otherwise.

[2] We thank a referee for noting that while there seems to be no consensus in the machine learning literature on whether machines are Bayesians or frequentists philosophically, this paper provides a testable condition for Bayesian learning and presents evidence that this condition, *loss calibration*, is satisfied in our experiment.

As a result, our paper is also connected to both a growing literature that considers stochastic choice to be essential for studying limited attention in human decision making (e.g., Manzini and Mariotti, 2014; Cattaneo et al., 2020) and a growing literature that studies the theoretical properties of costly learning (e.g., Gentzkow and Kamenica, 2014; De Oliveira et al., 2017; Hébert and Woodford, 2017; Denti, 2022; Lipnowski and Ravid, 2022). As with many of the models developed in the literature on costly learning, our models build off the core objects of information design (e.g., Kamenica and Gentzkow, 2011; Bergemann and Morris, 2019; Kamenica, 2019).[3]

The exercise in this paper is also in the spirit of a long literature in economics, psychology, and neuroscience that uses *human* choices in perceptual tasks to test models of human cognition. For instance, Dean and Neligh (2017) and Almog and Martin (2023) show that, like machine learners, human learners are consistent with a general model of costly learning and then study whether they are consistent with specific models of rational inattention, such as the Shannon model.[4] An alternative and complementary approach is to investigate whether humans can be modeled as machine learners. Samuelson and Steiner (2024) and Aridor et al. (2024) propose and study the variational autoencoder (VAE) model (Kingma and Welling, 2013) as a model of human cognition, making connections and comparisons with existing models of rational inattention.

Our analysis illustrates several advantages of applying cognitive economic methods and rational inattention theory to machine learners. First, the machine's loss function is a known and manipulable primitive of the decision problem, whereas a human's utility function must be inferred and is only indirectly manipulable. In contrast, Pattanayak and Krishnamurthy (2021) assume that each algorithm has an unobservable "utility" function that dictates the priorities it assigns to correct and incorrect predictions rather than treating the algorithm's objective as known and subject to external control as we do. A second advantage is that machines naturally generate SDSC data, which is particularly well-suited for analyzing such models. For human decision making, such data is harder to come by.[5] Finally, machines may better approximate and emulate the costly learning paradigm. Human decisions contain strong and possibly immutable deviations from Bayesian updating and optimal choice, as documented by an extensive literature in behavioral economics. For example, a human may update beliefs in a biased manner, possibly for self-protective reasons.

Despite these advantages, it is striking how little is known about algorithms as decision makers.[6] We help to fill this gap in the literature by showing that a leading modern machine learning algorithm is consistent and rationalizable according to standard models of human cognition. We hope that this opens the door to better understanding these increasingly important learners.

For instance, future work may uncover that machine learning algorithms can be modeled as having their own important set of biases that lead to departures from standard models of capacity-constrained and costly learning. It is known that some machine learning algorithms are consistently overconfident (Guo et al., 2017), which would violate the testable conditions for both of our proposed models. Can we model these algorithms as having a bias in their belief updating or a distortion in the loss function that they actually implement? Do machine learning algorithms fall prey to standard behavioral biases? Frank et al. (2023) show that algorithmic predictions of stock prices overreact to news using the standard test of overreaction, which is a conditional form of calibration.

Alternatively, future work could test more specialized models with specific functional forms of revealed algorithmic pseudo-cost. Restricting to classes of revealed algorithmic pseudo-cost, such as posterior separability (Caplin et al., 2022; Denti, 2022), might offer strengthened predictions. For example, using Shannon entropy as the algorithm's pseudo-cost would offer especially strong results about the predictions of the trained model under specific loss functions used in training.

The rest of the paper is as follows. Section 2 provides background details for the analysis and results that follow. Section 3 provides the foundation for the models and tests this foundation. Section 4 formally defines both the capacity-constrained learning model and the costly learning model and tests both models. Section 5.1 concludes with a discussion.

2. Background

In this section, we provide background details that are helpful for contextualizing the analysis and results that follow. First, we explain the difference between *as if* and *as is* approaches to modeling and why we follow the *as if* approach that is standard in economic theory. Second, we introduce the kind of data that we use in our analysis and why it is valuable for model testing. Finally, we provide details on the machine learning experiment that generates this data.

2.1. "As If" not "As Is"

Some might ask if there is any point in modeling machine learning using tools from economic theory. Why do we not just detail the actual inner workings of the algorithm? After all, that approach works for such algorithms as the simplex method for linear

[3] Liang et al. (2022) draw a point of connection between information design and machine learning to study the tradeoffs between accuracy and fairness.

[4] Unfortunately, machine learning experiments can be more expensive and resource intensive to run. For example, our experiment required approximately 1,800 wall-clock hours of compute time using Nvidia V100 and A100 GPUs on our university high-performance computing clusters (approximately 90 minutes a run × 400 runs per weight × 3 weights).

[5] Exceptions include sports (Archsmith et al., 2021; Bhattacharya and Howard, 2021; Almog et al., 2024), quality control settings, and lab experiments (Dean and Neligh, 2017; Almog and Martin, 2023). When observational data falls short because of issues such as selective labels, this requires further econometric work to address (Rambachan, 2021).

[6] Notable exceptions include Zhao et al. (2020), who embed behavioral forces in a neural net structure, Danan et al. (2020), who apply decision-theoretic approaches to recommendation systems, and Chen et al. (2023) and Kim et al. (2024), who look at whether Large Language Models (LLMs) satisfy lottery choice axioms.

programming. Why not do the same for modern machine learning algorithms, such as those based on neural nets or transformer architectures?

The fundamental reason is the extraordinary complexity of state-of-the-art machine learning algorithms. While we might understand and be able to fully characterize how the individual components of modern machine learning algorithms work, the entirety of the system is no longer analytically tractable. In addition, most enterprise machine learning protocols include many custom add-ons that make it even more challenging to understand how the totality of a system works.[7]

Therefore we use the *as if* approach that is standard in economic theory. With this approach, one asks whether the behavior of a complex system (a human, factory, economy, etc.) aligns with a simplified model that can be solved analytically. This alignment is tested using representation theorems that indicate that if a system's behavior follows certain properties or conditions, then it is "as if" they are behaving according to a specific model. For instance, economic models of human cognition do not summarize all visual saccades, neural firing rates, physical processes underlying information storage or processing, etc.; rather, they focus on actions to see if they aligned with a model in which choices are driven by optimal learning and utility maximization.

In this paper we adapt these same *as if* methods to characterize how machines learn. However, there is a critical difference with classical models of human choice and production. In these cases it is common to specify particular functional forms and estimate parametric classes of utility functions and production functions from appropriate data sets. The distinction is that we are at a more nascent phase in our understanding of how to model machine learning. Therefore our method involves testing applicability of broad classes of models, rather than estimating a single model within a fixed class.

In broad terms we pose three qualitative questions:

1. Does the algorithm learn in a manner that is optimal?
2. If so, what are the trade-offs implicit in that optimization?
3. Does the algorithm choose optimally given what it has learned?

To answer these questions we provide representation theorems such that if the data generated by algorithms satisfy properties or conditions, then it is "as if" they are behaving according to models of human cognition that specify a particular process of optimal learning and choice.

All of the models we explore involve treating the algorithm as an optimizing agent. This generates implications on counterfactuals involving feasible yet unchosen alternatives. Within decision problem, we observe the actions that are chosen; across decision problem, we observe other actions that are feasible but (depending on the model) may be differentially costly. In the capacity-constrained model, what is revealed to be feasible cannot be preferred to what is chosen. In the costly learning model, it can be preferred up to an information-dependent cost, which imposes a cyclical consistency condition on the payoffs of chosen alternatives.[8] Given that the logic is based on simple comparisons of chosen with feasible alternatives, our characterizations are transparent and facilitate simple statistical testing of these models. In particular, they reveal that it is possible to test these models by varying the machine's loss function. This is analogous to testing models of costly learning with humans by varying the incentives provided to them in decision problems, as in Dean and Neligh (2017).

As will be seen, two of our model class tests are passed, while one, despite its intuitive appeal, is rejected. We see this rejection in a positive light. It shows the power of our revealed preference approach. More generally, rejection improves understanding of what an algorithm does (or rather does not do), which also has relevance for the machine learning literature on cost-sensitive learning (Elkan, 2001) and class imbalance (e.g., Provost, 2000).

2.2. SDSC data in machine learning

The key qualitative difference between classical revealed preference theory and the version that we apply to machine learning is the need to allow for mistaken decisions that result from imperfect learning. This moves us into the branch of the theory that makes allowance for limited cognition by changing the ideal data from classical choice data to state dependent stochastic choice data (SDSC), starting with Caplin and Martin (2015). Key to this data is that it records not only what is chosen, but also what is true. This provides rich insights into mistakes, which is why the data set is so important in cognitive psychology and cognitive economics. What is important for current purposes is that SDSC is precisely the data that is generated when gauging the performance of standard supervised machine learning algorithms as now specified.

The key components of data are a finite set of *outcomes* Y (e.g., image types, disease severity levels, etc.) that the algorithm can learn about. There is also a set of *predictions* A that the algorithm can make. For example, many classification algorithms output a numeric *confidence score* for each possible outcome. In turn, confidence scores can be translated into discrete outcome predictions using a downstream classification rule, such as predicting an outcome if its confidence score exceeds a given threshold or the confidence scores of the other possible outcomes.

[7] We thank Tom Cunningham of OpenAI for pointing out this additional challenge to understanding why machine learning-based systems, particularly recommendation systems, make the predictions that they do.

[8] These conditions are respectively analogous to the weak (Samuelson, 1938) and generalized (Varian, 1982, Afriat, 1967) axioms of revealed preference. However, an important difference in our context is that the machine's losses (i.e., utility) are observed and manipulable, and play a role analogous to observed expenditure in classical revealed preference theory.

In our experiment, the set of possible outcomes $Y = \{0, 1\}$ is an indicator for the presence of pneumonia, and the set of possible predictions $A = [0, 1]$ is a continuous measure of the algorithm's "confidence" about the presence of pneumonia. We test numeric confidence scores instead of discrete downstream outcome predictions because confidence scores are more closely tied to machine incentives and yield stronger tests and sharper identification. However, our framework also accommodates situations where the analyst only has data on predicted outcomes or wishes to model the scoring algorithm jointly with a downstream classification rule.

An important input to the training of an algorithm is a *loss function* $L : A \times Y \to \mathbb{R}$, which indicates the value of a particular prediction given the outcome. One standard loss function is squared error, $L(a, y) = (a - y)^2$, for which failure to align the prediction with the outcome produces increasingly larger losses. Loss functions typically differ in how they value different types of mispredictions. For example, squared error puts a higher penalty on larger mispredictions than absolute error, $L(a, y) = |a - y|$.

For a given algorithm, using loss function L in training generates a series of predictions $a_1, ..., a_N$, one for each observation in a data set of size N. The performance of the trained model is assessed on how well its predictions align with actual outcomes in a data set. To perform this assessment, the analyst has access to the actual outcomes for observations in the data, which is a series of outcomes $y_1, ..., y_N$. Specifically, we use the labeling of Wang et al. (2017) for whether each chest X-ray indicates pneumonia or not. As is standard when evaluating algorithms, we assume this *ground truth* is correct, but our approach could be extended to include uncertainty about the ground truth.

We follow the standard practice of evaluating how well an algorithm performs on aggregate for each outcome. For example, chest X-rays that indicate pneumonia can vary in a multitude of ways, so the focus in Rajpurkar et al. (2017) is on how often the algorithm correctly predicts pneumonia instead of whether it correctly predicts pneumonia for a particular X-ray.[9] Formally, aggregate level performance for each outcome is summarized by *performance data* $P^L : A \times Y \to [0, 1]$, which is the joint distribution of predictions and outcomes for the trained model,

$$P^L(a, y) = \frac{1}{N} \sum_{n \in \{1,...,N\}} \mathbf{1}_{a_n = a \,\&\, y_n = y}$$

Because the data set is finite, the support of P^L over A, given by $\text{supp}(P^L_A)$, is also finite. As is standard practice in the machine learning literature, we study algorithmic predictions over a test sample that is independently drawn from the same population as the data on which the algorithm is trained.[10]

An adjustment is needed if the action space is sufficiently rich (say, the continuous unit interval) that each action is observed only once because this makes all probabilities in the performance data either 0 or 1. In such cases our approach can instead be applied to bins of actions, with binning achieved for example by rounding real numbers or aggregating them into empirical quantiles. Such binning introduces its own fine points that are not central to our theoretical approach, and so we do not introduce these additional formalities in our framework. We do, however, discuss them where empirically relevant in what follows.

2.3. Experimental design

We test economic models of machine learning by revisiting CheXNeXt, an influential deep convolutional neural network for predicting thoracic diseases from chest X-ray images (Rajpurkar et al., 2018). Our training models are generated using the ChestX-ray14 data set, which consists of 112,120 frontal chest X-rays which were synthetically labeled with the presence of fourteen thoracic diseases (Wang et al., 2017). The main modification we make to the CheXNeXt training procedure is that we isolate the task of pneumonia detection as in the earlier implementation of Rajpurkar et al. (2017).

When training deep learning neural networks to predict to which of two classes an observation belongs, it is standard practice to use cross-entropy $L(a, y) = -y \log a - (1 - y) \log(1 - a)$ over confidence scores and outcomes. In addition, it is also standard practice to re-weight the loss function to make losses higher or lower for a particular outcome; such class weighting is often employed when one outcome is less common (Thai-Nghe et al., 2010) or with the hope of achieving some external objective (Zadrozny et al., 2003). For example, to give more weight to cross-entropy when a chest X-ray indicated pneumonia ($y = 1$), the loss function in Rajpurkar et al. (2017) was approximately $-.99 \log(a)$ when $y = 1$ and $-.01 \log(1 - a)$ when $y = 0$. For that reason, we train the algorithm across various β-weighted cross-entropy loss functions:

$$L^\beta(a, y) = -\beta y \log(a) - (1 - \beta)(1 - y) \log(1 - a). \tag{1}$$

Specifically, we vary the loss function by considering $\beta = 0.7, 0.9, 0.99$.[11] In addition, we employ ensemble (model-averaging) methods to isolate the substantive effects of what the machine learns from random noise inherent to the stochastic training procedure. That is, for observation n, a_n is the average prediction across training runs. Using nested cross-validation methods, this yields an ensemble model prediction at each β for each of the 112,120 X-ray images in the original data. Further technical details of our training procedure are relegated to Appendix A. We return to our experiment in Subsection 3.3 after introducing our fundamental representation of machines as Bayesian expected loss minimizers.

[9] Our model and results can be readily extended to account for characteristics of instances by conditioning performance data on these characteristics.

[10] Our approach can be applied to any data set, but since algorithmic performance is typically evaluated on a hold-out or test set of instances, a natural interpretation is that this data set is the test set.

[11] For reference, the class weight used in the analysis of Rajpurkar et al. (2017) is approximately 0.99 because the probability of positive pneumonia cases in the data set is 0.0127.

3. Machines as Bayesian expected loss minimizers

In this section we present the information-theoretic foundation of the learning models we consider, the testable implications of this foundation, and positive evidence of this foundation in our experiment.

In this foundation we follow the Blackwell (1953) model of experimentation, signal processing, and choice. For both capacity-constrained and costly learning, we model an algorithm as an optimizing agent that i) starts with a prior $\mu \in \Delta(Y)$ over outcomes, ii) gets signal realizations that provide information about the outcome, iii) forms posterior beliefs $\gamma \in \Delta(Y)$ via Bayesian updating, and iv) chooses predictions based on these posteriors to minimize expected losses. As in Kamenica and Gentzkow (2011) we define Q as those distributions of posteriors with finite support that satisfy Bayes' rule,

$$Q \equiv \{Q \in \Delta(\Delta(Y)) | \sum_{\gamma \in \text{supp}(Q)} \gamma Q(\gamma) = \mu\}.$$

Posteriors are translated into probabilistic predictions through a prediction function $q : \text{supp}(Q) \to \Delta(A)$. For a given loss function L and distribution of posteriors $Q \in \mathcal{Q}$, the set of optimal prediction functions is defined as,

$$\hat{q}(L, Q) \equiv \underset{q}{\text{argmin}} \sum_{\gamma \in \text{supp}(Q)} Q(\gamma) \sum_{a \in A} q(a|\gamma) \sum_{y \in Y} \gamma(y) L(a, y).$$

Note that any pair (Q, q) produces a joint distribution of predictions and outcomes given by $P_{(Q,q)} : A \times Y \to [0, 1]$ where,

$$P_{(Q,q)}(a, y) \equiv \sum_{\gamma \in \text{supp}(Q)} Q(\gamma) q(a|\gamma) \gamma(y).$$

With these elements in place we can define the foundation of our subsequent learning models.

Definition 1. For a given loss function L, P^L has a **signal-based representation** (SBR) if there exists a prior $\mu \in \Delta(Y)$, a Bayes consistent distribution of posteriors $Q \in \mathcal{Q}$, and a prediction function $q : \text{supp}(Q) \to \Delta(A)$ such that:

1. The prior is correct: $\mu(y) = \sum_{a \in \text{supp}(P^L_A)} P^L(a, y)$.
2. Predictions are optimal at all possible posteriors: $q \in \hat{q}(L, Q)$.
3. Predictions are generated by the model: $P^L(a, y) = P_{(Q,q)}(a, y)$.

If P^L has an SBR, then it is *as if* the algorithm makes predictions to minimize the loss function given the Bayesian posterior beliefs induced by its signal structure.

3.1. Testable condition: loss calibration

The SBR foundation is characterized by a simple condition, called *loss calibration*, which requires that switching every prediction a to any alternative prediction a' — what we term a "wholesale" switch from a to a' — would never strictly reduce losses. To the best of our knowledge, no such condition has been proposed in the machine learning literature. Loss calibration is a restatement of the *No Improving Action Switches (NIAS)* condition of Caplin and Martin (2015) into the machine learning setting.

Definition 2. Performance data P^L is **loss calibrated** to loss function L if a wholesale switch of predictions does not reduce losses according to L:

$$a \in \underset{a' \in A}{\text{argmin}} \sum_{y \in Y} P^L(a, y) L(a', y) \text{ for all } a \in \text{supp}(P^L_A).$$

Applying the theoretical results of Caplin and Martin (2015) and Bergemann and Morris (2016), it is straightforward to show that loss calibration is necessary and sufficient for an SBR. For completeness we include a formal statement of the equivalence below.

Proposition 1 *(Caplin and Martin (2015); Bergemann and Morris (2016)). P^L has an SBR if and only if predictions are loss calibrated to L.*

As noted above, the characterization of optimizing models in SDSC rests on the idea that each model rules out improving changes in behavior. The key counterfactual in the SBR case is to imagine switching predictions. Whatever signals cause any prediction a to be chosen, the machine learner can in principle switch wholesale to any alternative prediction a'. The testable condition is that no switch of this type can be loss-reducing. Note that the wholesale nature of the switches is critical here. There is no way to know from the data alone what more sophisticated switches might have been possible based on the algorithm's actual signal structure. That makes it clear why loss calibration is necessary for an SBR. That it is sufficient is straightforward also: absent improving switches, one can construct an SBR straightforwardly using the probabilities of each state associated with any given prediction in the data.

Note that this axiom has "face credibility" in that it rules out clear errors in prediction. Any algorithm that fails this condition makes predictions that are not suitable for the loss function and that are thus inconsistent with an SBR and Bayesian expected loss minimization.[12]

Under weighted cross-entropy loss with binary outcomes (1), it is straightforward to show that loss calibration takes a unique closed form as a function of posterior probabilities. Let $a^\beta(\gamma)$ be the optimal prediction when the class weight is β and the posterior probability that the outcome is $y = 1$ is given by γ. In the case of binary outcomes, posterior probabilities are summarized by the scalar probability that the outcome is $y = 1$, and we associate this scalar with the posterior in what follows.

Observation 1. *Consider weighted cross-entropy loss (1). For any weight $\beta \in (0,1)$ and all posterior probabilities $\gamma \in supp(Q)$ that the outcome is $y = 1$, the unique loss calibrated confidence score is given by:*

$$a^\beta(\gamma) = \frac{\beta\gamma}{1-\beta-\gamma+2\beta\gamma}. \tag{2}$$

In the case of unweighted cross-entropy loss ($\beta = 0.5$), the loss calibrated scoring function (2) collapses to the optimal prediction being the posterior probability itself, $\alpha^{0.5}(\gamma) = \gamma$. Thus, as is well-known, unweighted cross-entropy incentivizes truthful revelation of beliefs. Such a loss function is said to be *proper* (Brier, 1950; McCarthy, 1956).

When $\beta > 0.5$, there is an incentive to overscore $\alpha^\beta(\gamma) > \gamma$, whereas when $\beta < 0.5$ there is an incentive to underscore $\alpha^\beta(\gamma) > \gamma$ all interior posteriors $\gamma \in (0,1)$. The impact of β on optimal scores can be quite strong, especially since class weights are frequently used in settings where class imbalance is large. For example, at posterior belief $\gamma = 0.5$, the optimal prediction for a given β is $\alpha^\beta(0.5) = \beta$.

3.2. Calibration in machine learning: theory and practice

In general, any proper loss function incentivizes truthful reporting under an SBR. Thus, an observable implication of SBR is that if the loss function is proper, an algorithm should be (unconditionally) *calibrated* to ground truth probabilities. That is, each confidence score should equal the true probability of the outcome given that score. This form of calibration is an important and much-studied property in machine learning because calibrated predictions correctly reflect their uncertainty. While there is no inherent tension between the aims of accuracy and calibration in our *as if* model of the machine, the relationship is more nuanced in the *as is* practice of machine learning.

Deep learning convolutional neural networks have been shown to suffer from miscalibration, specifically overconfidence, with the severity of miscalibration increasing in model size (Guo et al., 2017). However, the calibration of deep neural nets is improved with regularization procedures such as weight decay (Guo et al., 2017) as well as model ensembling (Lakshminarayanan et al., 2017). This is consistent with the intuition that both of these procedures reduce overconfidence. More recently, Minderer et al. (2021) conduct a comprehensive comparison of 180 image classification models and find that the most accurate current models, such as non-convolutional MLP-Mixers (Tolstikhin et al., 2021) and Vision Transformers (Dosovitskiy et al., 2021), are not only well-calibrated compared to earlier models, but also that their calibration is more robust to distributions that differ from training. We now turn to the experimental evidence for our generalized implication of *loss* calibration.

3.3. Experimental test: loss calibration

In the deep learning algorithm we consider — which regularizes through early stopping and aggregates over an ensemble of trained neural nets — we find that confidence scores are loss calibrated as in (2) across various weights in weighted cross entropy loss (1). Recall that this includes unconditional calibration for unweighted cross-entropy loss ($\beta = 0.5$), which is a proper loss function.

Graphical evidence of calibration and loss calibration is provided in the left and right panels of Fig. 1, respectively. In each plot, the horizontal axis represents the confidence score, and the vertical axis the corresponding pneumonia rate in the data (both on a log scale). The shapes in the figure provide the empirical decile-binned calibration curves (DeGroot and Fienberg, 1983; Niculescu-Mizil and Caruana, 2005). That is, for a given loss function, each point represents a decile bin of confidence scores, with the mean confidence score within bin on the horizontal axis, and the mean pneumonia rate within bin on the vertical axis. The solid lines represent the situation where confidence scores are calibrated. Thus, the algorithm appears effectively calibrated for the unweighted loss function $\beta = 0.5$, and miscalibrated otherwise. The dashed lines in the right plot show the theoretical relationship (2) between scores and pneumonia rates for the relative positive class weights $\beta = 0.7, 0.9, 0.99$ if an algorithm is loss calibrated (note that the loss calibrated and calibrated lines coincide on the left when $\beta = 0.5$). As β increases, the algorithm is increasingly incentivized to provide a score that is higher than the machine's actual "belief" about the probability of pneumonia, which causes the lines to bow out. The alignment of theoretical predictions and empirical estimates suggests that the algorithm is generally very close to being loss calibrated, and very close to being calibrated when the loss function is unweighted.

Finally, note that our finding of calibration with an unweighted loss function is consistent with previously documented calibration for deep learning convolutional neural networks that use regularization (i.e., weight decay in Guo et al. (2017)) and deep ensembling (Lakshminarayanan et al., 2017). With a viable SBR in hand, we now turn to the models of what the machine learns.

[12] Nevertheless, this is easy to rectify. Whenever this loss function is input into the algorithm, a single line of code at the end of the computer program making a wholesale switch to predicting a whenever it would have predicted a' would make this algorithm loss calibrated for this loss function. We say that an algorithm has been *loss re-calibrated* when it has been transformed through wholesale switches to become loss calibrated.

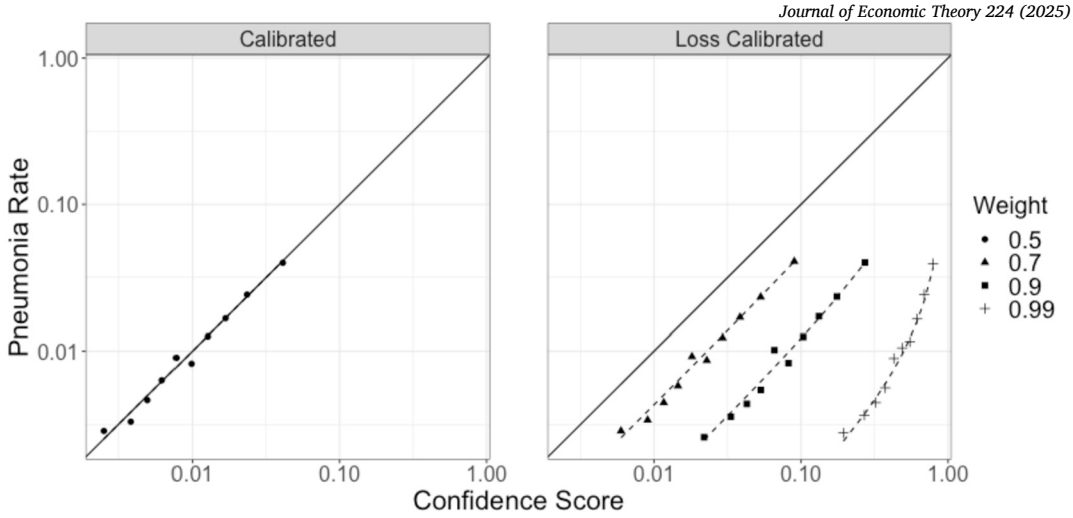

Fig. 1. Theoretical relationship (2) between confidence score and pneumonia rate for a loss calibrated algorithm with calibration target (solid lines), loss calibration targets varying class weights (dashed lines), and empirical decile-binned calibration curves (shapes) for the pneumonia-detection algorithm presented in Subsection 2.3. All objects are displayed on a log-10 scale to improve readability. This figure provides visual evidence that the algorithm is calibrated for $\beta = 0.5$ (left panel) and loss calibrated generally (left and right panels). This ensures an SBR representation and simplifies computation of the objects introduced in Section 4.

4. Models of machine learning

SBR leaves open the question of how a machine learning algorithm arrives at its signal structure – that is, what the machine learns based on the incentives provided by the loss function. We now provide two nested alternatives: choosing among a set of feasible signal structures or choosing among signal structures of different costs. Note that the latter class of models nests the former because one possible cost function is a simple indicator of feasibility, which takes a constant value for feasible signal structures and an infinite value otherwise.

4.1. Capacity-constrained learning

The first model assumes the algorithm chooses among a set of feasible signal structures to best match the incentives provided by the loss function.

To translate this into the SBR framework of Section 3, we define a feasible set of experiments $\mathcal{Q}^* \subset \mathcal{Q}$. This feasible set depends only on the algorithm's capability and is not specific to the loss function provided. We define the algorithm's strategy space Λ to include both Q and q:

$$\Lambda = \{(Q,q) | Q \in \mathcal{Q}, q : \text{supp}(Q) \to \Delta(A)\}.$$

For a given loss function L and feasible set $\mathcal{Q}^*$, the set of optimal strategies $\tilde{\Lambda}(L,\mathcal{Q}^*)$ is,

$$\tilde{\Lambda}(L,\mathcal{Q}^*) \equiv \underset{(Q,q)\in\Lambda, Q\in\mathcal{Q}^*}{\text{argmin}} \sum_{\gamma\in\text{supp}(Q)} Q(\gamma) \sum_{a\in A} q(a|\gamma) \sum_{y\in Y} \gamma(y) L(a,y).$$

With this we can define all performance data sets that are consistent with optimality for a given feasible set $\mathcal{Q}^*$ as,

$$\tilde{P}(L,\mathcal{Q}^*) \equiv \{P_{(Q,q)} | (Q,q) \in \tilde{\Lambda}(L,\mathcal{Q}^*)\}.$$

Capacity-constrained learning requires that there exists a feasible set $\mathcal{Q}^*$ such that the performance data produced by an algorithm are optimal given that feasible set for all $L \in \mathcal{L}$.

Definition 3. An algorithm is consistent with **capacity-constrained learning** if there exists a feasible set $\mathcal{Q}^* \subset \mathcal{Q}$ such that $P^L \in \tilde{P}(L,\mathcal{Q}^*)$ for all $L \in \mathcal{L}$.

4.2. Testing capacity-constrained learning

In order to simplify the characterizations of capacity-constrained learning, we restrict consideration to performance data sets with an SBR representation (or its empirically verifiable counterpart, loss calibration). Because indexing is useful in what follows, we take as given a finite set of M loss functions, indexed by $1 \le m \le M$. For notational simplicity, we denote the performance data set from training the algorithm with the m-th loss function as $P^m = P^{L^m}$.

Our characterization is taken directly from Caplin et al. (2024), who characterize capacity-constrained learning for human decision makers. In the context of machine learning, the key idea is to ensure that losses cannot be lowered by counterfactually switching to the predictions from training with a different loss function. As shown in Caplin et al. (2023), all such comparisons are visible in the *value of learning (VoL) matrix* G with generic element G^{mn} in row m and column n that specifies the minimized expected losses when the loss function is L^m and the performance data is P^n:

$$G^{mn} \equiv \sum_{a \in \text{supp}(P^n_A)} \min_{a' \in A} \sum_{y \in Y} L^m(a', y) P^n(a, y).$$

The operation on the right hand side of the equation takes any prediction $a \in \text{supp}(P^n_A)$, picks some alternative prediction $a' \in A$ to replace it wholesale, computes the corresponding expected losses for L^m, and then minimizes.

A capacity-constrained learning representation requires that no such switch of performance data can lower losses. To formalize we define the $M \times M$ *direct value difference matrix* D_0 by,

$$D_0^{mn} \equiv G^{mn} - G^{mm}. \tag{3}$$

An algorithm with an SBR is *strongly loss adapted* if for all $1 \leq m, n \leq M$,[13]

$$D_0^{mn} \geq 0, \text{ or equivalently } G^{mn} \geq G^{mm}.$$

This condition has a natural interpretation in our setting. For a capacity-constrained rationalization to be possible, we would expect that when the loss function is L^m, training with L^m generates lower losses than can be achieved by training with any other loss function.

The results in Caplin et al. (2024) can be used to show that, together with loss calibration, an algorithm being strongly loss adapted is necessary and sufficient for capacity-constrained learning. As indicated above, the theory implies that every signal structure chosen at one loss function is also feasible under another. For the theory of capacity-constrained learning to apply, no switches of chosen signal structures across loss functions can be improving after also accounting for changes in optimal predictions. That the theory also requires optimal predictions at all revealed posteriors establishes necessity of its being strongly loss adapted. Sufficiency is straightforward when we define the feasible set of signal structures as precisely those revealed in the data.

4.3. Costly learning

Our second model assumes the algorithm chooses among signal structures of different costs. Again, these are not necessarily related to the monetary costs incurred in running the algorithm, but rather reflect the pseudo-costs of the algorithm (the *as if* learning costs of the algorithm).

To formalize this, we define a pseudo-cost function $K : \mathcal{Q} \to \mathbb{R} \cup \infty$ and denote the set of all possible pseudo-cost functions as $\mathcal{K}$. An algorithm's pseudo-cost function depends only on its capabilities and is not specific to the loss function provided. Given loss function $L \in \mathcal{L}$ and pseudo-cost function $K \in \mathcal{K}$, the *cost-adjusted loss* $\hat{L}$ of strategy (Q, q) is,

$$\hat{L}((Q, q)|L, K) \equiv \sum_{\gamma \in \text{supp}(Q)} Q(\gamma) \sum_{a \in A} q(a|\gamma) \sum_{y \in Y} \gamma(y) L(a, y) + K(Q).$$

The corresponding set of optimal strategies $\hat{\Lambda}(L, K)$ is then defined as,

$$\hat{\Lambda}(L, K) \equiv \underset{(Q,q) \in \Lambda}{\text{argmin}} \ \hat{L}((Q, q)|L, K).$$

This optimization problem formalizes the way in which the algorithm trades off losses with pseudo-costs. Given any $L \in \mathcal{L}$ the set of all performance data sets that are consistent with optimality for a given pseudo-cost function $K \in \mathcal{K}$ are,

$$\hat{P}(L, K) \equiv \{P_{(Q,q)} | (Q, q) \in \hat{\Lambda}(L, K)\}.$$

Definition 4. An algorithm is consistent with **costly learning** if there exists a pseudo-cost function $K \in \mathcal{K}$ such that $P^L \in \hat{P}(L, K)$ for all $L \in \mathcal{L}$.

The second learning model generalizes the first model because a feasible set of posterior distributions $\mathcal{Q}^*$ is equivalently specified as a pseudo-cost function K^* for which the cost is zero for every feasible posterior distribution $Q \in \mathcal{Q}^*$ and infinite otherwise.

4.4. Testing costly learning

The costly learning model generalizes the capacity constrained model. It is consistent with the theory to find improving switches that characterized capacity constrained learning. Such switches may be improving provided they do not lower losses more than they

[13] Strongly loss adapted is a restatement of the *No Improving (Action and Attention) Switches (NIS)* condition of Caplin et al. (2024) into the machine learning setting.

raise costs. The question is how to characterize conditions for a costly learning representation without doing a precise cost-benefit comparison. This question has a definitive answer in the No Improving Attention Cycles (NIAC) conditions of Caplin and Dean (2015). Define $H(m,n)$ as all sequences of indices $\vec{h} = (h(1), h(2), \ldots, h(J(\vec{h})+1))$ of edge length $J(\vec{h})$ with $h(1) = m$ and $h(J(\vec{h})+1) = n$ in which the first $J(\vec{h})$ entries are distinct. The *indirect value difference matrix* D collects minimal summed loss differences across such paths

$$D^{mn} \equiv \min_{\{\vec{h} \in H(m,n)\}} \sum_{j=1}^{J(\vec{h})} D_0^{h(j)h(j+1)}. \tag{4}$$

Formally, an algorithm with an SBR is *loss adapted* if for all $1 \leq m \leq M$,[14]

$$D^{mm} \geq 0. \tag{5}$$

This condition requires that no cycle of switches could lower losses. In fact, the inequality of loss adaptedness can be replaced with an equality, since the identity cycle with $h(m) = m$ for all $1 \leq m \leq M$ is feasible and trivially yields a summed loss difference of zero. Applying the results in Caplin and Dean (2015), an algorithm with an SBR has a costly learning explanation if and only if it is loss adapted.

Intuitively, the construction and argument follow in two steps. The first step, as in the capacity-constrained model, is to construct the value of learning matrix G and thereby the direct value difference matrix D_0, with element $D_0^{mn} \equiv G^{mn} - G^{mm}$ (defined in (3)) summarizing the difference in loss between chosen and feasible performance data m and n for loss function m. Recall that if performance data n is strictly preferred to chosen performance data m under loss function m — that is, if $D_0^{mn} < 0$ — then this violates capacity-constrained learning; however, it may still be rationalizable by costly learning if the information underlying performance data m is more costly.

The second step determines whether such a cost-based rationalization exists. Letting K^m denote a candidate information cost associated with performance data P^m, a cost-based rationalization requires:

$$G^{mm} + K^m \leq G^{mn} + K^n, \text{ equivalently}$$

$$K^m - K^n \leq D_0^{mn}$$

for all decision problems m, n. The key is to consider summing such direct value difference inequalities across *cycles* of performance data because information cost differences $K^m - K^n$ across cycles sum to zero, regardless of the information cost function (which is to be inferred). This yields necessity of loss adaptedness by its definition (5); conversely, it can be constructively shown that loss adaptedness is sufficient for a costly learning representation (Caplin and Dean, 2015; Caplin et al., 2023).

For example, consider a simple hypothetical value of learning matrix and corresponding direct and indirect value difference matrices in a two-loss case:

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 0.5 \end{pmatrix}, \quad D_0 = \begin{pmatrix} 0 & -1 \\ 0.5 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} -0.5 & -1 \\ 0.5 & -0.5 \end{pmatrix}$$

The performance data under the second loss function yields lower losses under either the first or second loss function (comparing entries within row of G). Therefore, using performance data 2 under loss function 1 reduces losses under loss function 1 by $D_0^{12} = -1 < 0$, and so the capacity-constrained model is violated. This does not necessarily imply a violation of the costly learning model since the performance data under loss function 2 may be more costly. However, the net gain from using data 2 under loss function 1 (reducing loss by $D_0^{12} = -1$) exceeds the net cost of using data 1 under loss function 2 (increasing loss by $D_0^{21} = 0.5$); consequently, cycling performance data across the decision problems reduces aggregate losses by $D^{11} = D^{22} = D_0^{12} + D_0^{21} = -0.5$, which violates loss adaptedness and thus the costly learning model.

With more than two loss functions, the matter of computing aggregate losses across cycles — in turn embedded in the indirect loss difference matrix D — becomes slightly more complex. However, Caplin et al. (2023) show that if an algorithm is loss adapted, then the indirect value difference matrix can be computed by applying the polynomial-time algorithm of Floyd (1962) and Warshall (1962) (in economics, see also Varian, 1982) to the complete weighted directed graph with weight D_0^{mn} on the directed edge from node $1 \leq m \leq M$ to node $1 \leq n \leq M$. The Floyd-Warshall algorithm initializes with the direct value difference matrix $W \leftarrow D_0$ and computes the minimal aggregate loss as one adds the possibility of passing through intermediate nodes $\{1, \ldots, k\}$ for $1 \leq k \leq M$. At step k, there are two possibilities for the minimal aggregate loss W^{mn} across the subset of paths $H(m,n)$ passing through elements of $\{1, \ldots, k\}$: either aggregate losses are strictly reduced by allowing paths to pass through node k, in which case $W^{mn} \leftarrow W^{mk} + W^{kn}$, or not, in which case W^{mn} is left unchanged. At its conclusion $k = M$, the Floyd-Warshall algorithm recovers the indirect value difference matrix $W = D$ if the data is loss adapted, and a matrix W with some strictly negative entries on its diagonal otherwise. This provides an easy test of the costly learning model: an algorithm is loss adapted if and only if the candidate matrix thus computed has a zero diagonal.

[14] Loss adapted is a restatement of the *No Improving Attention Cycles (NIAC)* condition of Caplin and Dean (2015) into the machine learning setting.

4.5. Experimental results

The main products of our experiment are estimates of the VoL matrix G and indirect value difference matrix D introduced in the previous subsection. To facilitate their computation, we rely on the strong evidence for an SBR representation provided in Subsection 3.3. This strong evidence of loss calibration allows us to compute losses in the G matrix, as given below, by analytically recalibrating confidence scores inverting (2) to recover optimal confidence scores across weights.[15]

$$G = 0.01 \begin{array}{c} \begin{matrix} P^{0.7} & P^{0.9} & P^{0.99} \end{matrix} \\ \begin{pmatrix} 3.750 & 3.752 & 3.762 \\ 3.349 & 3.352 & 3.362 \\ 1.365 & 1.366 & 1.370 \end{pmatrix} \end{array} \begin{matrix} L^{0.7} \\ L^{0.9} \\ L^{0.99} \end{matrix}$$

Recall that a necessary and sufficient condition for capacity-constrained learning is that the algorithm is strongly loss adapted:

$$\mathcal{H}_0 : D_0^{mn} \equiv G^{mn} - G^{mm} \geq 0 \quad \text{for all } 1 \leq m, n \leq M.$$

In order to statistically test this multivariate one-sided hypothesis, we first estimate a 9×9 covariance matrix for the VoL matrix G via 10,000 bootstrap samples from the data set of ensemble predictions. We then compute p-values for the constituent univariate one-sided Wald tests and apply a Bonferroni correction. Even using this conservative approach to bounding the family-wise error rate, we reject the null hypothesis at standard levels of significance with $p = 0.001$. Further inspection of G reveals a systematic reason for why we reject the null hypothesis: loss functions have a common preference for the performance data from training with lower β. Thus, while the predictions for the loss function with weight $\beta = 0.7$ are consistent with the algorithm being strongly loss adapted, the predictions for the loss functions with weight $\beta = 0.9, 0.99$ are not.

Our second question is whether the algorithm is loss adapted, and thereby consistent with costly learning:

$$\mathcal{H}_0 : D^{mm} \geq 0 \quad \text{for all } 1 \leq m \leq M.$$

The estimated D matrix, given below, satisfies this null hypothesis.

$$D = 0.1^5 \begin{array}{c} \begin{matrix} P^{0.7} & P^{0.9} & P^{0.99} \end{matrix} \\ \begin{pmatrix} 0 & 2.548 & 12.467 \\ -2.529 & 0 & 9.938 \\ -6.993 & -4.463 & 0 \end{pmatrix} \end{array} \begin{matrix} L^{0.7} \\ L^{0.9} \\ L^{0.99} \end{matrix}$$

We therefore fail to reject that the algorithm is consistent with costly learning at any level of significance.[16] Intuitively, a necessary condition for this is that, even though all loss functions are minimized by switching to lower-β performance data, the gains from switching are lower for higher-β loss functions.

We conclude the section with a brief observation on the power of our test for costly learning: *any* reordering of chosen information structures would have resulted in a pointwise rejection of loss adaptedness. Thus, we failed to reject the null hypothesis in spite of — rather than in the absence of — a powerful test.

5. Conclusion

5.1. Discussion

Machine learning is increasingly central both to the modern economy and to the field of economics itself, where it has yielded improvements in policy-relevant predictions (Kleinberg et al., 2015), causal inference with high-dimensional data (Belloni et al., 2014, Athey, 2017), and the analysis of rich new sources of data (Gentzkow et al., 2019).[17] Furthermore, machine learning has been usefully applied in microeconomic theory, for example as a complement to economic theory (Fudenberg and Liang, 2019) and as a benchmark of model completeness (Fudenberg et al., 2022).

However, state-of-the art machine learning algorithms lack what Lipton (2018) refers to as *algorithmic transparency*: an understanding of why an algorithm chooses the prediction model that it does. For example, standard OLS has high algorithmic transparency because the resulting model is the unique solution to a convex optimization problem and has a closed-form expression in terms of the training data. Because modern machine learning algorithms lack such transparency, we propose two parsimonious *as if* representations of them. As with human decision-making, having a parsimonious representation that reasonably approximates machine learning behavior could enhance the theoretical and empirical analysis of modern machine learning, and more generally, would open the door to applying many tools of economics to better understand the latest approaches in machine learning.

[15] In turn, this analytical mapping circumvents the need to bin data to recover posterior beliefs, avoiding the finite sample issues associated therewith.

[16] Pointwise consistency with loss adaptedness is satisfied in 53% of our bootstrap samples.

[17] See Varian (2014), Mullainathan and Spiess (2017), Athey (2018), and Athey and Imbens (2019).

This paper introduces an economic approach to algorithmic transparency through modeling machine learning. Combining the methods of Bayesian revealed preference theory with the unique feature that algorithm training uses a known and manipulable loss function, we provide methods for testing and recovery of our models using data on machine learning predictions. Applying the theory to a state-of-the-art deep neural network for pneumonia detection in chest X-rays, we find empirical support for the information-theoretic, signal-based model foundations and for a costly model of machine learning. We statistically reject a stronger capacity-constrained learning model, in which machines optimally choose what to learn from a feasible set of information structures determined by the training data and algorithmic technology.

5.2. Future directions

We expound two classes of extensions and directions for future work. The first class expands the applications of the existing modeling approach. One such possibility is to explore whether attention costs are a proxy of the actual resource usage of the machine. For instance, the amount of computational resources spent could be driven by the algorithm's choice of a particular hyperparameter, which can be influenced by the loss function through hyperparameter optimization (e.g., Bergstra and Bengio, 2012). In our experiment, for example, an algorithm might perform more or fewer epochs if the losses under a particular loss function have not converged sufficiently (e.g., as implemented by the "ReduceLROnPlateau" function in Keras). Another possibility expands the set of experiments conducted on the machine. While this paper has focused on varying the algorithm's loss function, in future work it would also be interesting to vary a classification algorithm's choice set in multi-class problems, in the tradition of revealed preference for demand analysis.

The second and more speculative class of future directions expands the scope of the modeling approach. While this paper has focused on modeling the choices of the algorithmic training procedure, the question of machine learning interpretability is broader and forms an active and important literature. Beyond algorithmic transparency, interpretability is focused on understanding the resulting prediction model and why it makes the classification decisions or predictions that it does.[18] As articulated by Athey (2018), structural economic models naturally provide interpretability for such questions, and their interpretability can easily exceed that of "simpler" models, such as a linear model that nevertheless lacks such an interpretation.

CRediT authorship contribution statement

Andrew Caplin: Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Daniel Martin:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Philip Marx:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

None.

Appendix A. Experiment: technical details

Here we summarize the technical details of the experiment introduced in Section 2.3. Our model training procedure essentially follows that of the CheXNeXt algorithm (Rajpurkar et al., 2018), in which a deep neural network was trained using the ChestX-ray14 data set of Wang et al. (2017). The ChestX-ray14 data set consists of 112,120 frontal chest X-rays that were synthetically labeled with up to fourteen thoracic diseases. Our code for model training is adapted from the publicly available CheXNeXt codebase of Rajpurkar et al. (2018). However, we follow the earlier CheXNet implementation of Rajpurkar et al. (2017) in three ways. First, we restrict to the binary classification task of pneumonia detection, where the labels of interest are pneumonia ($y = 1$) or not ($y = 0$). In addition, we trade off a higher batch rate of 16 at the expense of a slightly smaller imaging scaling size of 224 by 224 pixels (instead of a batch size of 8 and an image rescaling of 512 by 512 pixels, respectively). As in Rajpurkar et al. (2018), we adopt random horizontal flipping, and normalize based on the mean and standard deviation of images in the ImageNet data set (Deng et al., 2009). For each model, we train a 121-layer dense convolutional neural network (DenseNet, Huang et al., 2016) with network weights initialized to those pretrained on ImageNet, using Adam with standard parameters 0.9 and 0.999 (Kingma and Ba, 2014), and using batch normalization (Ioffe and Szegedy, 2015). We use an initial learning rate of 0.0001 that is decayed by a factor of 10 each time the validation loss plateaus after an epoch, and we conduct early stopping based on validation loss. Each model was trained using either an Nvidia Tesla V100 16 GB GPU or an Nvidia Tesla A100 40 GB GPU on the Louisiana State University or Northwestern University high performance computing clusters, respectively.

Given the inferential nature of our exercise, we deviate from this prior art in two ways. First, we induce variation in the cross-entropy loss function (1) across multiple positive class weights $\beta_1 = 0.7, 0.9, 0.99$, with 0.99 approximately equal to the inverse probability class weights for pneumonia detection adopted in Rajpurkar et al. (2017). In addition to varying class weights, the main

[18] We refer again to Lipton (2018) for a general overview and useful taxonomy of interpretability, and also to Montavon et al. (2018) for an overview of "post-hoc" interpretability of deep neural networks specifically.

difference in our implementation and the implementation of Rajpurkar et al. (2017) are our data splits and our recourse to additional ensemble methods to account for randomness in the training procedure. This use of ensemble methods also likely explains why our confidence scores are loss calibrated, despite recent evidence that deep neural networks and cross-entropy loss may inherently produce poor calibration because of overconfidence (Bai et al., 2021, Liu et al., 2022). Specifically, we adopt a nested cross-validation approach where we randomly split the data set into ten approximately equal folds and then iterate through 70-20-10 train-validation-test splits (the split distribution also used in Wang et al. (2017) and a secondary application of Rajpurkar et al., 2017). We train a total of 480 models, yielding an ensemble of 96 trained models for each observation in the data set where that observation was in a test fold. The final score for each observation in the data set is then obtained by averaging confidence scores across the observation's ensemble. This procedure is repeated on the same set of data splits for each class weight $\beta = 0.7, 0.9, 0.99$ we consider.

Data availability

Data will be made available on request.

References

Afriat, Sydney N., 1967. The construction of utility functions from expenditure data. Int. Econ. Rev. 8 (1), 67–77.
Agarwal, Nikhil, et al., 2023. Combining human expertise with artificial intelligence: Experimental evidence from radiology, Tech. Rep., National Bureau of Economic Research.
Almog, David, Martin, Daniel, 2023. Rational inattention in games: experimental evidence. In: Working Paper.
Almog, David, et al., 2024. AI oversight and human mistakes: evidence from centre court. In: Working Paper.
Alós-Ferrer, Carlos, Fehr, Ernst, Netzer, Nick, 2021. Time will tell: recovering preferences when choices are noisy. J. Polit. Econ. 129 (6), 1828–1877.
Alur, Rohan, Raghavan, Manish, Shah, Devavrat, 2024. Distinguishing the indistinguishable: human expertise in algorithmic prediction. ArXiv preprint arXiv:2402.00793.
Archsmith, James E., et al., 2021. The Dynamics of Inattention in the (Baseball) Field, Tech. Rep., National Bureau of Economic Research.
Aridor, Guy, Azeredo da Silveira, Rava, Woodford, Michael, 2024. Information-constrained coordination of economic behavior. Tech. rep. National Bureau of Economic Research.
Arora, Sanjeev, et al., 2019. Implicit regularization in deep matrix factorization. Adv. Neural Inf. Process. Syst. 32.
Athey, Susan, 2017. Beyond prediction: using big data for policy problems. Science 355, 483–485.
Athey, Susan, 2018. The impact of machine learning on economics. In: The Economics of Artifical Intelligence: An Agenda. University of Chicago Press, pp. 507–547.
Athey, Susan, Imbens, Guido W., 2019. Machine learning methods that economists should know about. Annu. Rev. Econ. 11, 685–725.
Bai, Yu, et al., 2021. Don't just blame over-parameterization for over-confidence: theoretical analysis of calibration in binary classification. ArXiv preprint arXiv:2102.07856.
Barrett, David G.T., Dherin, Benoit, 2021. Implicit gradient regularization. In: International Conference on Learning Representations (ICLR).
Belloni, Alexandre, Chernozhukov, Victor, Hansen, Christian, 2014. High-dimensional methods and inference on structural and treatment effects. J. Econ. Perspect. 28 (2), 29–50.
Bergemann, Dirk, Morris, Stephen, 2016. Bayes correlated equilibrium and the comparison of information structures in games. Theor. Econ. 11 (2), 487–522.
Bergemann, Dirk, Morris, Stephen, 2019. Information design: a unified perspective. J. Econ. Lit. 57 (1), 44–95.
Bergstra, James, Bengio, Yoshua, 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13 (2).
Bhattacharya, Vivek, Howard, Greg, 2021. Rational inattention in the infield. Am. Econ. J. Microecon.
Blackwell, David, 1953. Equivalent comparisons of experiments. Ann. Math. Stat., 265–272.
Brier, Glenn W., 1950. Verification of forecasts expressed in terms of probability. Mon. Weather Rev. 78 (1), 1–3.
Caplin, Andrew, 2024. Data engineering for cognitive economics. J. Econ. Lit. Forthcoming.
Caplin, Andrew, Dean, Mark, 2015. Revealed preference, rational inattention, and costly information acquisition. Am. Econ. Rev. 105 (7), 2183–2203.
Caplin, Andrew, Dean, Mark, Leahy, John, 2017. Rationally inattentive behavior: characterizing and generalizing Shannon entropy. Tech. rep. National Bureau of Economic Research.
Caplin, Andrew, Dean, Mark, Leahy, John, 2022. Rationally inattentive behavior: characterizing and generalizing Shannon entropy. J. Polit. Econ. 130 (6), 1676–1715.
Caplin, Andrew, Martin, Daniel, 2015. A testable theory of imperfect perception. Econ. J. 125 (582), 184–202.
Caplin, Andrew, Martin, Daniel, Marx, Philip, 2023. Rationalizable learning. In: NBER Working Paper Series.
Caplin, Andrew, et al., 2024. Testing capacity-constrained learning. In: Working Paper.
Cattaneo, Matias D., et al., 2020. A random attention model. J. Polit. Econ. 128 (7), 2796–2836.
Chen, Yiting, et al., 2023. The emergence of economic rationality of GPT. Proc. Natl. Acad. Sci. 120 (51), e2316205120.
Danan, Eric, Gajdos, Thibault, Tallon, Jean-Marc, 2020. Tailored recommendations. Soc. Choice Welf., 1–20.
De Oliveira, Henrique, et al., 2017. Rationally inattentive preferences and hidden information costs. Theor. Econ. 12 (2), 621–654.
Dean, Mark, Neligh, Nate Leigh, 2017. Experimental Tests of Rational Inattention.
DeGroot, Morris H., Fienberg, Stephen E., 1983. The comparison and evaluation of forecasters. J. R. Stat. Soc., Ser. D, Stat. 32.1–2, 12–22.
Deng, Jia, et al., 2009. Imagenet: a large-scale hierarchial image database. In: Computer Vision and Pattern Recognition (CVPR), pp. 248–255.
Denti, Tommaso, 2022. Posterior separable cost of information. Am. Econ. Rev. 112 (10), 3215–3259.
Dosovitskiy, Alexey, et al., 2021. An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations.
Elkan, Charles, 2001. The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 973–978.
Floyd, Robert W., 1962. Algorithm 97: shortest path. Commun. ACM 5 (6), 345.
Frank, Murray Z., Gao, Jing, Yang, Keer, 2023. Behavioral machine learning? Computer predictions of corporate earnings also overreact. ArXiv preprint arXiv:2303.16158.
Fudenberg, Drew, Liang, Annie, 2019. Predicting and understanding initial play. Am. Econ. Rev. 109 (12), 4112–4141.
Fudenberg, Drew, et al., 2022. Measuring the completeness of economic models. J. Polit. Econ. 130 (4), 956–990.
Gentzkow, Matthew, Kamenica, Emir, 2014. Costly persuasion. Am. Econ. Rev. 104 (5), 457–462.
Gentzkow, Matthew, Kelly, Bryan, Taddy, Matt, 2019. Text as data. J. Econ. Lit. 57 (3).
Gunasekar, Suriya, et al., 2017. Implicit regularization in matrix factorization. In: Advances in Neural Information Processing Systems, vol. 30.
Guo, Chuan, et al., 2017. On calibration of modern neural networks. In: International Conference on Machine Learning. PMLR, pp. 1321–1330.
Hébert, Benjamin, Woodford, Michael, 2017. Rational inattention and sequential information sampling. Tech. rep. National Bureau of Economic Research.
Huang, Gao, et al., 2016. Densely connected convolutional networks. ArXiv preprint arXiv:1608.06993.

Ioffe, Sergey, Szegedy, Christian, 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML), pp. 448–456.
Kamenica, Emir, 2019. Bayesian persuasion and information design. Annu. Rev. Econ. 11, 249–272.
Kamenica, Emir, Gentzkow, Matthew, 2011. Bayesian persuasion. Am. Econ. Rev. 101 (6), 2590–2615.
Kim, Jeongbin, et al., 2024. Learning to be homo economicus: can an LLM learn preferences from choice. ArXiv preprint arXiv:2401.07345.
Kingma, Diederik, Ba, Jimmy, 2014. Adam: a method for stochastic optimization. ArXiv preprint arXiv:1412.6980.
Kingma, Diederik P., Welling, Max, 2013. Auto-encoding variational Bayes. ArXiv preprint arXiv:1312.6114.
Kleinberg, Jon, et al., 2015. Prediction policy problems. Am. Econ. Rev. Pap. Proc. 105 (5), 491–495.
Lakshminarayanan, Balaji, Pritzel, Alexender, Blundell, Charles, 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. Adv. Neural Inf. Process. Syst. 31.
Liang, Annie, Lu, Jay, Mu, Xiaosheng, 2022. Algorithmic design: fairness versus accuracy. In: Proceedings of the 23rd ACM Conference on Economics and Computation, pp. 58–59.
Lipnowski, Elliot, Ravid, Doron, 2022. Predicting choice from information costs. ArXiv preprint arXiv:2205.10434.
Lipton, Zachary C., 2018. The mythos of model interpretability. ACM Queue 16 (3), 31–57.
Liu, Sheng, et al., 2022. Deep probability estimation. ArXiv preprint arXiv:2111.10734.
Manzini, Paola, Mariotti, Marco, 2014. Stochastic choice and consideration sets. Econometrica 82 (3), 1153–1176.
Matejka, Filip, McKay, Alisdair, 2015. Rational inattention to discrete choices: a new foundation for the multinomial logit model. Am. Econ. Rev. 105 (1), 272–298.
McCarthy, John, 1956. Measures of the value of information. Proc. Natl. Acad. Sci., 654–655.
Minderer, Matthias, et al., 2021. Revisiting the calibration of modern neural networks. In: Neural Information Processing Systems (NeurIPS).
Montavon, Grégoire, Wojciech, Samek, Müller, Klaus-Robert, 2018. Methods for interpreting and understanding deep neural networks. Digit. Signal Process. 73.
Mullainathan, Sendhil, Spiess, Jann, 2017. J. Econ. Perspect. 31 (2), 87–106.
Neyshabur, Behnam, Tomioka, Ryota, Srebro, Nathan, 2015. In search of the real inductive bias: on the role of implicit regularization in deep learning. In: International Conference on Learning Representations (ICLR).
Niculescu-Mizil, Alexandru, Caruana, Rich, 2005. Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 625–632.
Pattanayak, Kunal, Krishnamurthy, Vikram, 2021. Behavioral economics approach to interpretable deep image classification. Rationally inattentive utility maximization explains deep image classification. ArXiv preprint arXiv:2102.04594.
Provost, Foster, 2000. Machine learning from imbalanced datasets 101. In: Proceedings of the AAAI 2000 Workshop on Imbalanced Datasets, vol. 68, pp. 1–3.
Rajpurkar, Pranav, et al., 2017. CheXNet: radiologist-level pneumonia detection on chest x-rays with deep learning. ArXiv preprint arXiv:1711.05225.
Rajpurkar, Pranav, et al., 2018. Deep learning for chest radiograph diagnosis: a retrospective comparison of the CheXNeXt algorithm to practicing radiologists. PLoS Med. 15 (11), e1002686.
Rambachan, Ashesh, 2021. Identifying Prediction Mistakes in Observational Data.
Samuelson, Larry, Steiner, Jakub, 2024. Constrained Data-Fitters. Yale University and University of Zurich, CERGE-EI, and CTS.
Samuelson, Paul A., 1938. A note on the pure theory of consumer's behaviour. Economica 5 (17), 61–71.
Sims, Christopher A., 2003. Implications of rational inattention. J. Monet. Econ. 50 (3), 665–690.
Thai-Nghe, Nguyen, Gantner, Zeno, Schmidt-Thieme, Lars, 2010. Cost-sensitive learning methods for imbalanced data. In: The 2010 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.
Tolstikhin, Ilya O., et al., 2021. MLP-mixer: an all-MLP architecture for vision. Adv. Neural Inf. Process. Syst. 34.
Varian, Hal R., 1982. The nonparametric approach to demand analysis. Econometrica, 945–973.
Varian, Hal R., 2014. J. Econ. Perspect. 28 (2), 3–28.
Wang, Xiaosong, et al., 2017. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2097–2106.
Warshall, Stephen, 1962. A theorem on Boolean matrices. J. ACM 9 (1), 11–12.
Woodford, Michael, 2014. Stochastic choice: an optimizing neuroeconomic model. Am. Econ. Rev. 104 (5), 495–500.
Zadrozny, Bianca, Langford, John, Abe, Naoki, 2003. Cost-sensitive learning by cost-proportionate example weighting. In: Third IEEE International Conference on Data Mining. IEEE, pp. 435–442.
Zhao, Chen, et al., 2020. Behavioral neural networks. In: Available at SSRN 3633548.