Philosophy of Science (2026), **93**, 123–142
doi:10.1017/psa.2025.10159

PSA PHILOSOPHY OF SCIENCE ASSOCIATION

PHILOSOPHY OF SCIENCE

ARTICLE

Peirce in the Machine: How Mixture of Experts Models Perform Hypothesis Construction

Bruce Rushing

University of Virginia, Charlottesville, VA, USA
Email: ejf9db@virginia.edu

(Received 05 June 2024; revised 06 August 2025; accepted 09 August 2025; first published online 03 September 2025)

Abstract

Mixture of experts is a prediction aggregation method in machine learning that aggregates the predictions of specialized experts. This method often outperforms Bayesian methods despite the Bayesian having stronger inductive guarantees. We argue that this is due to the greater functional capacity of mixture of experts. We prove that in a limiting case, mixture of experts will have greater capacity than equivalent Bayesian methods, which we vouchsafe through experiments on non-limiting cases. Finally, we conclude that mixture of experts is a type of abductive reasoning in the Peircean sense of hypothesis construction.

1. Introduction

Machine learning has developed many powerful techniques for solving prediction problems over the past few decades. Among these are prediction aggregation methods that seek to combine multiple machine learning models to make better predictions than can be achieved by individual predictors. This has strong theoretical foundations from the theory of meta-induction (Cesa-Bianchi and Lugosi 2006; Schurz 2019) as well as empirical support for the various methods (Dietterich 2000; Rokach 2010; Masoudnia and Ebrahimpour 2014). Two important strategies for prediction aggregation are Bayesian model averages (BMAs) and mixtures of experts (MOEs). The former is a Bayesian method to make predictions by averaging predictions across a posterior of possible models learned from the data; the latter involves learning specialized machine learning models called experts and then making predictions based on which experts would make the best prediction based on observed features. BMA has been a widely preferred strategy because of strong theoretical guarantees about its inductive superiority. But surprisingly, MOE often performs better in practice, which has led to its adoption as a crucial element in many cutting-edge machine learning models like large language models. We aim to address this startling fact in this paper.

© The Author(s), 2025. Published by Cambridge University Press on behalf of Philosophy of Science Association. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

We argue that MOEs outperform BMAs because MOEs create more expressive hypotheses or models than BMAs. It can be shown that in at least one exact sense of capacity, an MOE can have greater capacity than a BMA composed of models similar to that MOE. We then validate this theoretical result with experiments that show that as the capacity required to predict a dataset grows, MOEs maintain good performance while BMAs fall off based on the capacity of their members. Both of these results suggest that the superiority of MOEs over BMAs, despite the latter's inductive guarantees, stems from the restriction of BMAs to a poorer hypothesis class than MOEs.

The philosophical upshot of this finding is that MOEs can be thought of as doing abduction, in the Peircean sense of hypothesis construction, along with induction. In comparison, BMAs and other similar schemes are fundamentally inductive alone. We argue that MOEs demonstrate a divide-and-conquer strategy for hypothesis formation, similar to a suggestion given by Peirce. This means that MOE addresses a problem that Bayesian methods fail to address, which is where the hypothesis space comes from.

Importantly, this should be of interest to philosophers for two reasons. First, it provides a novel formal model for doing abduction in the Peircean sense. Second, it ties abduction to the rich mathematical framework of statistical learning theory, which could enable further formal theories to be produced.

Here is how our argument proceeds. First, we discuss MOEs as applied to supervised learning, i.e., learning from labeled data. Second, we review BMAs, reasons for their optimality, and empirical evidence pushing against that optimality. We discuss possible explanations for this difference but find them to be wanting. Third, we propose another explanation centered on the capacity of MOEs compared to BMAs. We show that with capacity made exact, we can prove that the capacity of MOEs will exceed the capacity of BMAs. Fourth, we build on this theoretical result by demonstrating experiments that show MOE performance maintains itself on datasets that require a certain amount of capacity to learn well while BMAs drop off. Fifth, we discuss the philosophical implications of this explanation, by highlighting that MOEs are essentially a form of abduction in the sense of hypothesis construction.

2. Mixtures of experts

In supervised learning, the goal of the machine learning algorithm is to predict some target y from some vector of features $\mathbf{x}$ (variables the model observes), and this goal is realized by training the model on some dataset of n target and feature pairs $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1,\ldots,n}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the training features and targets respectively. A model can be thought of as learning a probability function $f(y|\mathbf{x}, \theta)$, where θ are the model parameters, from which predictions can be sampled or deterministically picked as the target with the highest probability, i.e., the prediction $\hat{y} = \text{argmax}\, f(y|\mathbf{x}, \theta)$. Training is accomplished by minimizing some loss or cost function $L(\theta)$, which measures the dis-utility of a prediction error, through a training algorithm to find the best set of model parameters θ for making predictions. Models are then evaluated by how well they do on some hold-out test data set $\mathcal{T} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1,\ldots,m}$ not seen during training. The best models have the lowest loss on this test set.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

For example, suppose the problem is to predict blood pressure from body mass index (BMI). We treat our target y as the continuous random variable blood pressure and our feature vector $\mathbf{x} = [1, x_1]^{\mathrm{T}}$ as the continuous random variable BMI, x_1, and identity feature, 1. A linear regression is then a normally distributed probability function $f_{\mathrm{lr}}(y|\mathbf{x}, \theta)$ with known variance σ^2 and parameter vector $\theta = [\theta_0, \theta_1]^{\mathrm{T}}$, where θ_1 is the slope and θ_0 is the intercept, that characterize the mean of our distribution:

$$f_{\mathrm{lr}}(y|\mathbf{x}, \theta) = \mathcal{N}\left(\mathbf{x}^{\mathrm{T}}\theta, \sigma^2\right), \tag{1}$$

where $\mathbf{x}^{\mathrm{T}}$ is the transpose of our feature vector. Assuming statistical independence of our model parameters, we then fit our model by maximizing the likelihood $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} f_{\mathrm{lr}}(\mathcal{Y}|\mathcal{X}, \theta)$, which corresponds to minimizing the negative log-likelihood loss function L_{NLL}:

$$L_{\mathrm{NLL}}(\theta) = -\sum_{i=1}^{n} \log f_{\mathrm{lr}}(y^{(i)}|\mathbf{x}^{(i)}, \theta). \tag{2}$$

Due to our model being normally distributed and ignoring constants that do not depend on θ, the negative log-likelihood equates to minimizing the squared error:

$$L_{\mathrm{NLL}}(\theta) = \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y^{(i)} - \mathbf{x}^{(i)\mathrm{T}}\theta)^2. \tag{3}$$

We could then train our linear regression on the dataset $\mathcal{D}$ via gradient descent—an iterative optimization algorithm—to find the parameters that minimize this loss function. To evaluate how well our regression does, we evaluate its negative log-likelihood on a hold-out set of blood pressure and BMI pairs not seen during training. This is done because we want our model to inductively generalize from its training; we don't want the model to have merely memorized the blood pressure and BMI pairs found in training. The best model is one that successfully minimizes the training loss while having the lowest test set loss.

Aggregating predictions from many machine learning models is a common technique for improving predictive performance in supervised learning. There are two primary families of aggregation: ensemble methods and MOE methods.

In ensemble methods, we train individual models independently across the set of all training features $\mathcal{X}$ and then use some aggregation scheme across all models to produce a prediction. Examples of aggregation schemes include averaging and voting. In the averaging case, the prediction of an ensemble machine learning model $g(\mathbf{x})$ is an average of n models with probabilities $f(y|\mathbf{x}, \theta_i)$ parameterized by θ_i and weighted by weights w_i for $i = 1, \ldots, n$:

$$g(\mathbf{x}) = \sum_{i=1}^{n} w_i f(y|\mathbf{x}, \theta_i). \tag{4}$$

The core intuition behind ensemble techniques is that they leverage the wisdom of the crowds: Instead of relying upon one highly specialized model for prediction, many models are consulted and then factored into the prediction. Betting markets in sports gambling can be thought of as an ensemble technique leveraging human gamblers as individual predictors, where the going price in the market corresponds to an average of all wagers. Examples of ensemble techniques include bagging, where the random

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

predictors are trained and weighted on random subsets of data, and boosting, a technique where weak predictors are combined into stronger predictors by iteratively training predictors on their peers' errors.

With an MOE (Jacobs et al. 1991; Jordan and Jacobs 1994; Yuksel et al. 2012; Masoudnia and Ebrahimpour 2014), models called experts specialize on elements of a partition of the feature set $\mathcal{X}$, and predictions on a new sample from the mixture consist of routing the sample to the relevant experts based on where in that partition the sample falls. In the most general case where we route the sample to all experts, we employ weights for the experts that are a function of the sample features $\mathbf{x}$. The MOE model $h(\mathbf{x}; n)$ with n experts then is a sum over experts with probability functions $f_i(y|\mathbf{x}, \theta_i)$ parameterized by θ_i and a gating function $G_i : \mathcal{X} \to [0, 1]$ for $i = 1, \ldots, n$:

$$h(\mathbf{x}; n) = \sum_{i=1}^{n} G_i(\mathbf{x}) f_i(y|\mathbf{x}, \theta_i). \tag{5}$$

The gating functions $G_i(\mathbf{x})$—a rule for assigning weights to the experts—are typically probabilities, such as the softmax (a function for converting numbers to probabilities) over a linear model. We say that experts are homogeneous when expert probability functions f_i have the same form but are characterized by different parameters θ_i and inhomogeneous otherwise. For example, an MOE with only linear regressions as experts has homogeneous experts. Importantly, the gating function sometimes considers more than one expert in making a prediction—it sometimes assigns positive probability to multiple experts for a given sample. In those cases, the gating function is said to "softly partition" the feature set between experts, which means that it uses an expert's predictions for more than one partition cell. The limiting case where only one expert is selected from a group of experts—where only one expert e_i receives $G_i(\mathbf{x}) = 1$ and all the rest zero—makes this partition hard: each expert specializes on one and only one cell of a partition over $\mathcal{X}$. We will call this the top-expert MOE. In that case, $h(\mathbf{x}; n)$ is simply a piecewise function that applies the experts $e_i(\mathbf{x})$[1] on $\mathbf{x}$ depending on what member of the partition $Z_i \in \{Z_1, \ldots, Z_n\}$ of $\mathcal{X}$ that $\mathbf{x}$ happens to belong in:

$$h(\mathbf{x}; n) = \begin{cases} e_1(\mathbf{x}), & \mathbf{x} \in Z_1, \\ e_2(\mathbf{x}), & \mathbf{x} \in Z_2, \\ \vdots & \vdots \\ e_n(\mathbf{x}), & \mathbf{x} \in Z_n. \end{cases} \tag{6}$$

Here, expert e_1 only provides advice on cell Z_1, e_2 on cell Z_2, and so on. The intuition behind MOE is familiar to consumers of medicine; when presented with symptoms, we will often go to the doctor we think is best specialized in diagnosing and treating illnesses that typically present those symptoms. If we happen to choose only one doctor for clinical advice, we employ the top-expert MOE, and if the doctors are all broadly similar in how they approach problems—say by receiving standard medical training instead of alternative medicine—then they are homogeneous.[2]

[1] Experts are functions mapping features $\mathbf{x}$ to targets: $e_i(\mathbf{x}) = \hat{y}$.

[2] This analogy fails for MOE training: expert selection and expert training occur jointly.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

MOEs and ensembles, while both aggregation techniques, have important differences. Those differences consist of (i) how they are trained and (ii) how they aggregate predictions.

In the training regime, ensembles train their models separately and independently from their aggregation scheme. For example, in bagging (Breiman 1996), models are trained separately on random subsets of data drawn from $\mathcal{D}$ and only combined at inference time with their aggregation scheme. Boosting (Schapire 1990; Freund and Schapire 1996) modifies this regime by training models sequentially on the whole data but weighting data samples by the errors of the previous model in the sequence to gradually scale up better predictors. In contrast, the experts and the gating function are all trained simultaneously in MOE; the gating function learns the relevant soft partition while the experts specialize in their particular region of data. This is done with either gradient descent or the expectation maximization algorithm. Unlike ensembles, this means the aggregation scheme is not done separately in inference but is crucial to the training process because of the tight connection between the gating function and the experts.

It is important to emphasize that the aggregation schemes between ensembles and MOEs differ in the relevance of the sample features for aggregating predictions. The critical difference lies in how we aggregate: In the ensemble method shown in equation 4, the weights w_i are scalars and therefore are constant for all inputs, while in the MOE given in equation 5, the weights are determined by a gating function $G_i(\mathbf{x})$, which explicitly makes the contribution of each expert *conditional* on the input sample's features $\mathbf{x}$.[3] So aggregation proceeds very differently between ensembles and MOEs.

An extremely influential ensemble scheme is BMA, where all possible predictors of a certain class are considered by requiring the weights w_i to be probabilities that reflect the posterior distribution of the model parameters given the data observed. This ensemble method has important theoretical guarantees, and we turn to discussing it now.

3. Bayesian optimality

BMA receives its name from the application of Bayes' rule to statistical and machine learning model prediction. Recall that Bayes' rule states that if $\theta_1, \theta_2, \ldots$ is a partition of model parameters, then the conditional probability of an element of that partition θ_i given some observed data $\mathcal{D}$ is the ratio of the likelihood times the prior probability and the marginal probability of the data:[4]

$$p(\theta_i|\mathcal{D}) = \frac{p(\mathcal{D}|\theta_i)p(\theta_i)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta_i)p(\theta_i)}{\sum_{j=1}^{\infty} p(\mathcal{D}|\theta_j)p\left(\theta_j\right)}. \tag{7}$$

This is called the posterior probability, and it is applied in the *posterior predictive distribution* $p(y|\mathbf{x}, \mathcal{D})$, which is used to make a probabilistic prediction. The key idea is

[3] MOE resembles Schurz's (2019, sec. 8.2) conditionalized meta-induction, which has optimality conditions.

[4] For uncountable parameter partitions: $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \theta)p(\theta|\mathcal{D})\, d\theta$.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

that predicting some target y from features $\mathbf{x}$ and some previously observed data can be had by applying the law of total probability with respect to the model partition $\theta_1, \theta_2, \ldots$ and weighting the model likelihoods by the posterior of the model given the data:

$$p(y|\mathbf{x}, \mathcal{D}) = \sum_{i=1}^{\infty} p(y|\mathbf{x}, \theta_i) p(\theta_i|\mathcal{D}). \tag{8}$$

In essence, this states that if we average over all possible models by how likely it is that those models are correct given the data we have observed, then we can compute a posterior of the target given the observed data.[5] This has several desirable properties.

First, it allows us to form predictions with a natural regularizer. Regularization is a technique common in machine learning to combat overfitting. Overfitting is where a machine learning algorithm essentially does well on training data while failing to predict hold-out test data; the model "memorizes" the training data instead of truly learning the relevant "inductive patterns in the data." An explanation for this behavior comes from the Probably, Approximately Correct (PAC) learning framework (Kearns and Vazirani 1994): models that overfit are too "complex" for the data.[6] Consequently, model complexity is penalized by an additional term in the loss function during training. This is called regularization, and the additional term in the loss function that penalizes complexity is called a regularizer. In the case of BMA, the posterior $p(\theta|\mathcal{D})$ builds in a natural regularizer with the prior $p(\theta)$; if the model is a linear or polynomial regression and the prior is the standard normal, then the zero-centered prior assigns a lower probability to models with large parameter values, effectively penalizing model complexity similar to the complexity penalties in the Akaike and Bayesian information criteria. So BMA naturally provides better generalization and leads to "simpler" models over other aggregation techniques.

Second, BMA produces models that have good decision-theoretic properties. We consider our loss function as capturing the relative dis-utility of making decisions, and a machine learning model as providing a decision rule (here we treat predictions as a type of action). Then, if we suppose there is a "true" parameter θ for generating the data $\mathcal{D}$, we say the frequentist risk of a model is the likelihood of the data given the true-parameter-weighted expected value of that model's loss. For example, when predicting the outcome of a sequence of coin tosses, the frequentist risk would be how well our model does on some loss for those tosses weighted by how likely each toss is given the bias of the coin. So supposing us to be realists about chances, the frequentist risk captures how well a machine learning model makes decisions relative to the true chances. This notion of risk has a tight connection to BMA. A BMA model is a *Bayesian estimator* in the sense that it minimizes the Bayes risk with respect to the posterior $p(\theta|\mathcal{D})$: the Bayes risk of a model with prior $\rho(\theta)$ is simply the $\rho(\theta)$-weighted expected value of the frequentist risk. Instead of assuming a "true" parameter, we propose a prior probability distribution for the parameter and average out that prior across the frequentist risk. Suppose we are unsure what the true bias of a coin is when observing

[5] The posterior predictive distribution assumes conditional independence: $\mathbf{x} \perp \mathcal{D}|\theta$ and $\theta \perp \mathbf{x}|\mathcal{D}$.

[6] Complexity is irrelevant to PAC-learning guarantees (Herrmann 2020). See Sterkenburg (2023) for further discussion.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

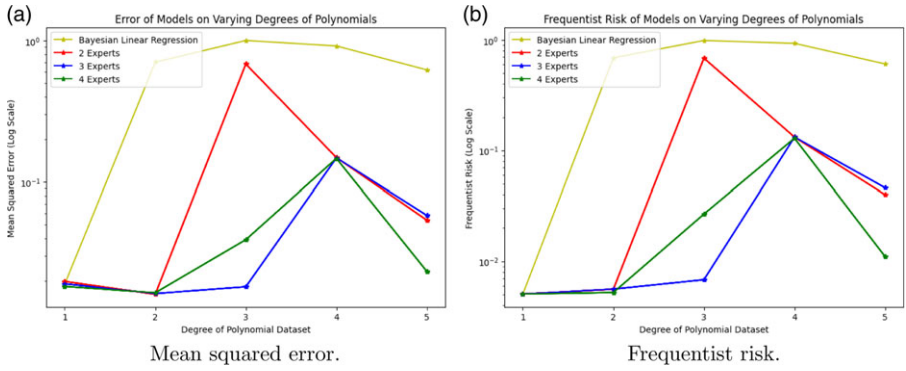


Figure 1. Plots of the mean squared error and frequentist risk for Bayesian linear regressions and MOE with [2, 3, 4] expert linear regressions on polynomial datasets of degrees [1, 2, 3, 4, 5]. Lower mean squared error and frequentist risk is better.

a sequence of coin tosses. Then we can consider the Bayes risk as putting a prior distribution over the bias and weighting the frequentist risk by that prior. If our model of the coin toss minimizes that risk, then we say it is a Bayesian estimator for the aforementioned prior. Consequently, it should be clear that a BMA model is a Bayes estimator concerning the posterior $p(\theta|\mathcal{D})$ because of how it is defined.

The so-called complete class theorems are an important property connecting the frequentist risk and the Bayes risk. These theorems state that Bayesian estimators form a complete class in the sense that any machine learning model that does better on the frequentist risk than any other model is a Bayesian estimator (Wald 1947; Robert 2007; Murphy 2022)[7] Critically, however, these optimality guarantees hold only *within a fixed class of models*. For example, a BMA composed of linear regression models can find the optimal linear predictor within the convex hull of the considered models, but it cannot generate a different type of model such as a quadratic predictor. Fixing the class appropriate class of models, the complete class theorems mean that there will be some Bayesian prior we can average over to equal the "correct" model of the data that does best on the loss; the hope is that we can then find that prior through our posterior by conditioning on the data. So, ideally, a BMA with enough data and non-perverse prior will be the Bayesian estimator that does as well as one can do on the frequentist risk. This should show up in terms of minimizing the loss on the hold-out test set.

With both of these desirable properties, it is then surprising that BMA is often outperformed by a MOE.

In figure 1, we show experiments involving a type of BMA and similar MOEs. The former is a Bayesian linear regression, which averages across a posterior $\mathcal{N}(\theta|\mu, \Sigma)$ all linear regressions given in equation 1, and the latter is an MOE involving two or more linear regressions as found in equation 1 (details can be found in the

[7] A model dominates another if its frequentist risk is lower for all θ. Admissible models aren't strictly dominated. The complete class theorem: Admissible models are Bayesian estimators for some prior (Wald 1947).

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

supplementary, appendix 2). Both models were trained on identical training data and evaluated on the same hold-out test data. The data consists of polynomials with some added normally distributed noise. Figure 1a shows the performance of these models on the mean squared error on hold-out test data, where lower is better; all mixture models match or exceed the performance of the Bayesian linear regression—with the greatest gap between the larger number of experts on higher-degree polynomials. Furthermore, since we control the data generation, we can directly calculate the frequentist risk. Again, the MOEs perform best here as seen in figure 1b, with the results closely tracking the mean squared error where lower is better. This collides with the expected hope that we should find the BMA model to do better here.

One might think that these results are due to the MOEs employing more sophisticated experts, such as polynomial regressions, or a non-probabilistic aggregation scheme, but that is not the case. Both the Bayesian linear regression and the MOEs average over linear regressions, i.e., the MOEs are homogenous: each expert is accounted for in the Bayesian linear regression, and those experts receive some weighted probability according to the posterior. If anything, the MOEs employ *fewer* experts since a BMA averages across all possible expert configurations while the most we consider in our experiments are four experts. Furthermore, the averaging scheme employed here is not all that different; both the Bayesian linear regression and the MOEs weight their subcomponent models by probabilities. In fact, the Bayesian linear regression does things the right way by weighting the models by a data-driven posterior. So the difference between the two cannot be due to better experts or a non-probabilistic approach to aggregation.

The ability of MOE to outperform ensembles like BMA has led them to be an important component of contemporary machine learning techniques. For example, Yuksel et al. (2012) cite a 2008 survey of then-important machine learning methods and argue that MOE outperforms all of them (Yuksel et al. 2012, 1178). They cite in particular the advantages an MOE model has over the popular ensemble method of boosting (Yuksel et al. 2012, 1187–88). Furthermore, Masoudnia and Ebrahimpour (2014) argue that MOE consistently outperforms popular ensemble techniques like bagging and boosting (Masoudnia and Ebrahimpour 2014, 286–87). This is despite the challenges MOE faces in learning the right partition and gating function (see Masoudnia and Ebrahimpour (2014, sec. 3.3)). Those challenges have increasingly been overcome through sparsity, i.e., only selecting the top-k experts when making predictions, and noisy gating, i.e., injecting noise during the training process to force the gating function to use all experts (Shazeer et al. 2017). This has led to MOEs being an increasingly important component of large language models, like OpenAI's GPT-4 (Betts 2024) and Mistral's Mixtral (Jiang et al. 2024).

Two explanations have been given for the superiority of MOE over ensemble methods like BMA. First, MOEs work well on data generated by oxymoronic one-to-many functions, i.e., "functions" that can map a single input to multiple outputs, whereas traditional models and ensembles cannot learn this data (Murphy 2022, 454). However, this explanation fails in the experiments described above: all of the datasets were generated by proper functions that pass the vertical line test. Second, MOEs supposedly work well due to experts learning data that is negatively correlated (Masoudnia and Ebrahimpour 2014, 287). While this is true for some data, it is not always the case, as can be seen in the aforementioned experiments on degree-three

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

polynomials where different elements of the data are either positively correlated or not correlated at all. So both of these explanations seem unnecessary for explaining why MOE does well.

Recapping, an important class of ensemble models are Bayesian ones. These models have attractive theoretical features such as regularization and accuracy considerations. But these models are often outperformed by MOEs, as demonstrated by some experiments. This has led to a general adoption of MOEs in machine learning for hard problems over ensemble schemes, such as in large language models. Two reasons for this superiority are that MOEs can fit data generated by one-to-many "functions" and that MOEs train their experts on negatively correlated data. But better performance can be found on proper functions and the partition learned need not involve negative correlations. So another explanation should be given, which we turn to now.

4. The functional capacity of models

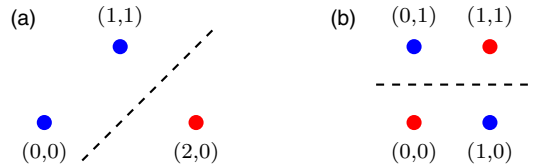
A plausible, intuitive hypothesis for why MOEs outperform ensemble techniques like BMA is that MOEs have greater functional capacity than ensembles. That is, MOEs characterize hypothesis classes that have a richer set of labeling schemes than ensembles like BMA, and so the improved performance relative to provably "better" methods like BMA is due to BMA being restricted to a more impoverished model class. This shows up in the experiments in section 3, where we see that the various MOEs continue to do well on mean squared error loss as the degree of the polynomial data scales. The Bayesian is in some sense limited by only considering linear regressions, while the MOEs seem able to stitch together multiple functions—despite the MOEs, like the BMA, only using linear regressions.

To make this hypothesis exact, we need to do two things. First, we need to specify what is a good explication of functional capacity, and second, we need to show how, in principle, at least some MOEs have a greater functional capacity than a BMA.

The intuitive idea of the functional capacity of a model is supposed to be something like the set of functions the model can learn. What counts as a relevant function depends on the problem the machine learning model solves. In supervised learning, those problems are sorted into the buckets of regression and classification, which correspond to predicting continuous targets or discrete targets respectively. With classification, the prediction problem is one of separating data into sets identified with the target, and the complexity of the problem is in some sense dependent on the data being used—predicting the label of a picture from raw pixels is harder than predicting from a higher-level feature like the presence or absence of an animal. Measuring how well we can separate data is the core intuition behind the Vapnik–Chervonenkis dimension (VC dimension). A more sophisticated model with higher functional capacity will be one that can better sort more complex data into classes; VC dimension gives us an exact measure of that capacity in terms of the cardinality of the data. This makes it a good explication of functional capacity because it specifies the type of functions a model can fit by one important sense of the complexity of the data in the classification problem at hand.

To define VC dimension, we need to define what it means to separate data, i.e., to *shatter* a dataset. Our context is binary classification, where machine learning models

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press



Shattering three points. Not shattering four points.

Figure 2. An example of shattering and failing to shatter. Figure 2a shows a set of points that can be shattered by the set of linear classifiers because any arbitrary labeling, i.e., which points are assigned blue or red, can be correctly classified by at least some linear classifier, i.e., we can draw a line separating the two labels without any mistakes. Figure 2b shows a set of points that cannot be shattered by that set since no line can separate this particular coloring without any errors.

are treated as functions that map between features $\mathcal{X}$ and the binary labels $\{0, 1\}$. For example, we might be interested in classifying whether a picture is of a cat or not. We then group our binary classifiers into sets $\mathcal{H}$ and specify that $\mathcal{H}$ shatters some dataset X if for any arbitrary labeling of X there is a classifier in $\mathcal{H}$ that correctly classifies that labeling.

Definition 4.1. *Let $\mathcal{H}$ be a set of binary classifiers $h : \mathcal{X} \to \{0, 1\}$. Given a set of points $X = \{x^{(1)}, \ldots, x^{(m)}\}$ where $x^{(i)} \in \mathcal{X}$, we say that $\mathcal{H}$ shatters X if for any labels $Y = \{y^{(1)}, \ldots, y^{(m)}\}$ where $y^{(i)} \in \{1, 0\}$, there exists $h \in \mathcal{H}$ such that $h(x^{(i)}) = y^{(i)}$ for $i = 1, \ldots, m$.*

Suppose our set of models are all linear classifiers, i.e., they classify by drawing lines to separate data. Then, as can be seen in figure 2, this class can shatter a given set of three points in $\mathbb{R}^2$ as seen in figure 2a, but there is no set of four points that it can shatter as seen in figure 2b. It shatters the former because regardless of how we label those three points, some line can separate them, which implies there is some linear classifier that can make a completely correct and error-free identification of the targets. It fails to shatter the latter because those four points have a labeling that no line can cleanly separate error-free.

The VC dimension of a class $\mathcal{H}$ is then the cardinal size of the largest set of data, with respect to some space of possible data like $\mathbb{R}$, that that class can shatter.

Definition 4.2. *The Vapnik–Chervonenkis dimension of a set of binary classifiers $VCD(\mathcal{H})$ defined over instance space $\mathcal{X}$ is the cardinality of the largest finite subset of $\mathcal{X}$ that $\mathcal{H}$ can shatter. If arbitrarily large subsets can be shattered, then $VCD(\mathcal{H}) := \infty$.*

For example, linear classifiers in the Cartesian plane can shatter any three-point labeling, but for any four points there will be a labeling that cannot be shattered. So the VC dimension of the set of linear classifiers concerning $\mathbb{R}^2$ will be exactly three.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

It should be noted that VC dimension is not correlated with parameter count: one can have a high VC dimension with a low parameter count. For example, the set of models given by $I(\sin\alpha \cdot \mathbf{x})$, where $I(x)$ is the indicator function that assumes 1 if $x > 0$ and 0 otherwise, has only one parameter $\alpha \in \mathbb{R}$ but an infinite VC dimension (Hastie et al. 2009, 237). This makes VC dimension a better characterization of model functional capacity than raw parameter count since the above model intuitively is more complex than a linear classifier—even though the linear classifier has more parameters.

Importantly, ensembles have a maximum VC dimension given by the number of ensemble members and the base VC dimension of the class of members used in the set of ensemble hypotheses. Let $L(\mathcal{H}, n)$ be the set of binary classifiers that are linear combinations of hypotheses drawn from binary classifiers $\mathcal{H}$, i.e., $L(\mathcal{H}, n) = \left\{x \mapsto sgn\left(\sum_{i=1}^{n} w_i f_i(x)\right) : \mathbf{w} \in \mathbb{R}^n, \forall i, f_i \in \mathcal{H}\right\}$, where sgn is the signum function, which returns $+1$ for positive or zero inputs and -1 for negative inputs. By definition, every ensemble will be in this set. Then it can be shown that if $VCD(\mathcal{H}) = m$ is finite for some finite $m > 3$, then $VCD(L(\mathcal{H}, n)) \leq \tilde{O}(mn)$, where $\tilde{O}$ means we ignore any constants or logarithmic terms (Shalev-Shwartz and Ben-David 2014, 109).[8] This is effectively to say that the VC dimension of ensembles with n members and whose members have a VC dimension m is no bigger than nm. While these bounds do not technically apply to BMAs, since BMAs consider infinitely many hypotheses, in practice they do because almost all BMAs are approximated through finite sampling schemes, i.e., Monte Carlo methods. Crucially, this is an upper bound: It can be shown in many cases that the VC dimension is significantly less—sometimes it is exactly the same VC dimension as the base class $\mathcal{H}$ (see Shalev-Shwartz and Ben-David (2014, 113), exercise 10.4, for examples). The upshot is that ensembles are bounded above by the number of members and the capacity of those members to separate data.

MOE, however, can be shown to have a higher VC dimension than the experts it employs. Jiang (2000) showed that for mixtures of n Bernoulli binary classifiers on $\mathbb{R}$, the VC dimension is exactly n, and for logistic regressions or Bernoulli binary classifiers on the instance space of $\mathbb{R}^d$, the VC dimension can be bounded below by the number of experts n and from above to the worst case $O(n^4 d^2)$ (Jiang 2000, 7). These bounds can be tightened on $\mathbb{R}$. First, we say a set of functions $\mathcal{F}$ whose domain is the reals is closed under translation just in case, for any function f in that set and any $a \in \mathbb{R}$, the function $f_a(x + a)$ is also in $\mathcal{F}$. Examples of these sets will be the set of all linear binary classifiers, the set of all polynomial binary classifiers of some positive degree d, and the set of all feedforward neural networks (a prominent type of artificial neural network where information flows in only one direction) whose domains are the reals.[9] Second, suppose we restrict ourselves to a limiting case involving top-expert MOEs with homogeneous experts drawn from a class closed under translation.

[8] From Shalev-Shwartz and Ben-David (2014, 109), Lemma 10.3: $VCD(L(\mathcal{H}, n)) \leq n(VCD(\mathcal{H}) + 1)\left(3\log n(VCD(\mathcal{H} + 1)) + 2\right)$.

[9] Linear and polynomial classifiers are closed under translation (modifying intercept or constant term). For feedforward networks, input translation modifies the first layer's bias: $h_1(x + a) = f\left(W_1 x + \mathbf{b}_1'\right)$ where $\mathbf{b}_1' = W_1 a + \mathbf{b}_1$.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

In that case, we can show that the VC dimension will have a lower bound that is a product of the number of experts and those experts' VC dimension.[10]

Proposition 4.3. *Let $\mathcal{H}$ be a set of top-expert mixture of experts binary classifiers of n homogeneous binary classifier experts drawn from the set $\mathcal{E}$ closed under translation with $VCD(\mathcal{E}) = m$ defined over instance space $\mathbb{R}$. Then $VCD(\mathcal{H}) nm$ for $\mathbb{R}$.*

What this means is that at least some MOEs have a higher VC dimension than BMAs and so greater functional capacity. They can fit more complex data and so involve fundamentally more complex hypotheses. The intuition behind Proposition 4.3 can be seen by observing that a top-expert MOE with homogenous experts is a piecewise function that deploys each expert on exactly one element of a partition of the feature space; such a piecewise function will be able to separate more complex data than any of its components individually. Likewise, non-top-expert MOEs will form similar piecewise functions, though of possibly less complexity depending on the soft partition learned by the gating function. The upshot is that with functional capacity understood as VC dimension, we can vouchsafe the earlier intuition that MOEs have greater "capacity" and can learn more complex functions than ensembles like BMA.

This means that comparing MOEs to BMAs is an apples-to-oranges comparison. If MOE and Bayesian models are restricted to sub-models with the same complexity, then MOE will have an inherent advantage on more complex datasets like those observed in section 3. Provide the Bayesian the right type of models and that advantage will likely disappear.

In summary, we have argued that an intuitive explanation for the superiority of MOE over ensembles like BMA—despite the theoretical reasons for the Bayesian to be advantaged—is the greater functional capacity of MOEs. We argued that VC dimension is a good explication of that capacity, and we proved that the VC dimension of some MOEs will be greater than a BMA over models with the same capacity as the MOE. However, theoretical considerations can only cut so much ice. The question is whether MOEs, such as those that are not top-experts, can do well on datasets that exactly track greater VC dimensions. We turn to that now.

5. Experiments

The discussion and results in section 4 provide a theoretical reason why MOEs may have greater functional capacity than ensembles like BMAs, but does this show up empirically? Two problems prevent an immediate acceptance of this fact. First, Proposition 4.3 only applies to top-expert MOEs but most MOEs in practice have multiple experts per partition; second, the proposition applies only if the top-expert MOE is learned when this may often prove difficult or impossible in practice.[11] Consequently, we should empirically validate our theoretical results. We address this worry here by detailing experiments involving datasets that stand as a proxy for the VC dimension of binary classifiers.

[10] Full proof in supplementary, appendix 1.

[11] Compare with universal approximation theorems (Hornik et al. 1989): they bound network expressivity but don't guarantee specific networks approximate target functions.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

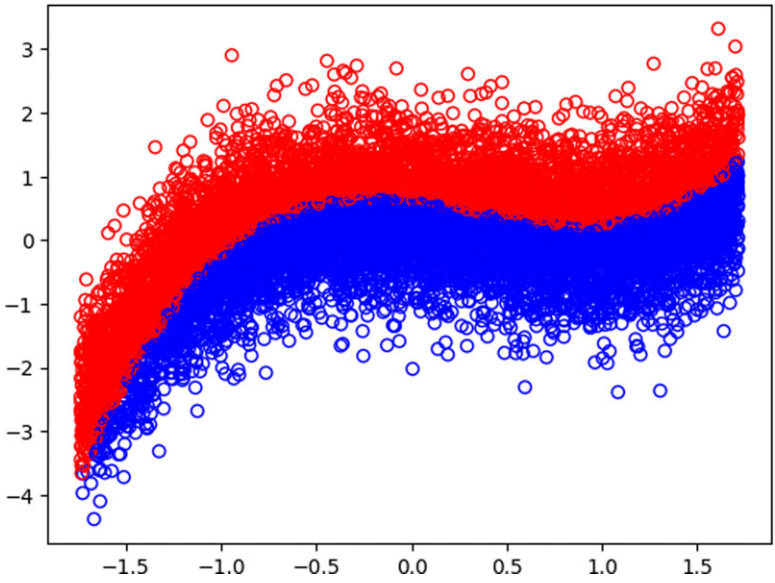


Figure 3. Example of a VC dimension polynomial dataset, degree 3. Red indicates the polynomial binary classifier assigns label 0 and blue indicates the classifier assigns label 1. The resulting correlated VC dimension is $\binom{5}{2} = 10$.

Our method for building datasets with a correlated VC dimension takes advantage of the fact that polynomial binary classifiers of degree m over instance space $\mathbb{R}^d$ have a VC dimension of $\binom{d+m}{d}$ (Shalev-Shwartz and Ben-David 2014, 57). We can then use those binary classifiers to generate datasets that in fact will be classifiable by machine learning models with the appropriate VC dimension. An example dataset can be seen in figure 3. We build a polynomial binary classifier of degree m with coefficients $[\theta_m, \ldots, \theta_0]$, and then we generate points in $\mathbb{R}^2$ around that polynomial with a normal distribution. Points that lie above the polynomial are assigned one label and those below another label. We say this dataset has a correlated VC dimension of $\binom{2+m}{2}$ since we can achieve near-perfect accuracy by fitting a logistic regression with a design matrix that is a degree-m polynomial.

Our chosen binary classifiers for the experiments are logistic regressions. A logistic regression is a binary linear classifier with likelihood $f_\sigma(y|\mathbf{x}, \theta)$ where $\theta = [\theta_0, \ldots, \theta_d]^{\mathrm{T}}$ and $\mathbf{x} = [1, x_1, \ldots, x_d]^{\mathrm{T}}$ such that

$$f_\sigma(y|\mathbf{x}, \theta) = \mathrm{Ber}(y|\sigma(\mathbf{x}^{\mathrm{T}}\theta)). \tag{9}$$

Here, Ber is the Bernoulli distribution and σ is the logistic sigmoid function $\sigma(x) = 1/[1 + \exp(-x)]$. Our MOEs consist of two to four experts that are logistic regressions that employ a top-two expert configuration, and we choose a BMA that

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

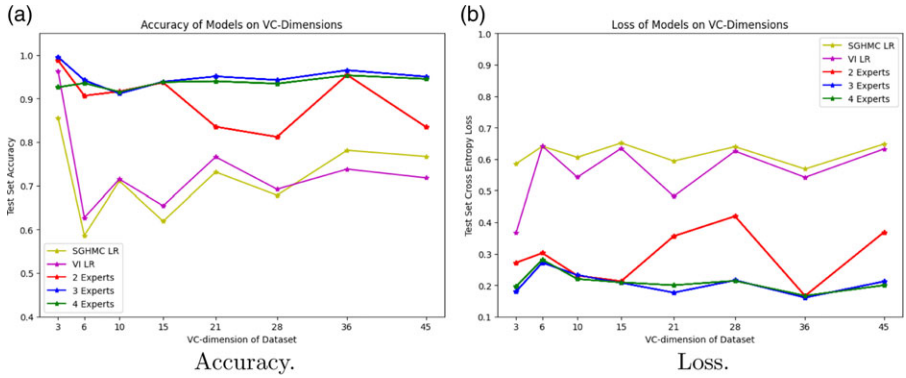


Figure 4. The hold-out test set accuracy and loss of stochastic gradient Hamiltonian Monte Carlo (SGHMC) and variational inference (VI) logistic regression (LR) BMAs and logistic regression MOEs. For accuracy in figure 4a, higher is better; for loss in figure 4b, lower is better. We see that as the VC dimension increases, the accuracy and loss of the BMAs falls off, while the accuracy of the MOEs stays relatively constant, with some degradation in the two-expert model.

averages over logistic regressions. Unfortunately, there are no analytically computable posteriors $p(\theta|\mathcal{D})$ for logistic regressions, so we have to approximate it for the BMA (see supplementary, appendix 2 for details).

All models were trained to minimize the cross-entropy loss function (a type of loss function for categorical targets) using stochastic gradient descent (a variant of gradient descent that estimates the gradient via random samples) on a training set, and we then evaluated them on a hold-out test set. The results can be seen in figure 4.

The experiments show that at a correlated VC dimension of 3, the Bayesian models perform roughly on par with the MOEs, but then there is considerable separation as the VC dimension of the dataset climbs. Figure 4a shows that BMAs consistently underperform on test-set accuracy, only doing modestly better than chance; however, the MOEs perform well as the VC dimension continues to climb, though the MOE with two experts shows degraded accuracy relative to the other models. In figure 4b, we see the same trends on test-set accuracy reflected in the cross-entropy loss: the Bayesian models approach the loss on hold-out test data as the MOEs, but then underperform as the VC dimension climbs. Again, the two-expert MOE model shows degradation more so than the higher expert counts.

These results validate empirically the theoretical results from section 4. Note that the bound proved in Proposition 4.3 is a lower bound for top-experts, so the VC dimension of the MOEs may be higher. The upshot is that we have both theoretical and empirical reasons to suggest that our explanation for the superiority MOEs over ensembles like BMA is correct: MOEs can simply learn more expressive hypotheses than averaging schemes like BMAs.

6. Mixtures of experts as abductive modeling

The argument from the previous sections has been that MOEs outperform averaging schemes like the Bayesian ones because they avail themselves of a richer hypothesis

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

space. What is very interesting about this is that they find themselves in that bigger hypothesis space by way of simpler hypotheses; the experts, as noted, have a lower functional capacity than the MOE as a whole. Through the gating function, the feature space is soft-partitioned into cells that expert models can then specialize on. Importantly, this allows a virtuous learning process that enables experts to master relatively narrow domains while ensuring that collectively the experts build a more sophisticated, broader picture of the problem at hand. We argue in this section that this process is a form of *abduction* in the Peircean sense of hypothesis construction. This explains the fundamental difference between MOE and BMA: the former combines abduction and induction while the latter only uses induction.

A big assumption of the Bayesian approach to inductive reasoning is that the hypothesis space must be given. We have an algebra (or sigma-algebra) that captures in some sense all of the questions that we want to have answered in our inquiry and evidence we could bear on those questions. Then we proceed in our inquiry by considering some partition that is a subset of that algebra that characterizes the relevant scientific hypotheses and applying Bayes' rule on new evidence as it comes in to update ourselves on our credences in the members of that partition. The assumption about the hypothesis space shows up twice here: first, in the granted algebra that characterizes inquiry, and second, in the selected hypothesis partition used when applying Bayes' rule. We see both assumptions play a crucial role in BMA. The initial algebra is the possible model parameters and the range of values the feature and target random variables may assume, and the hypothesis partition is the range of possible model parameters assumed. With this fixed, no further hypothesis or evidence may be admitted into the averaging scheme. So BMA perfectly embodies the Bayesian approach—an approach that is fundamentally an inductive one.

By an inductive approach, we mean an ampliative inference from evidence to hypotheses. For the Bayesian, this can be thought of as a type of diachronic coherence between our current partial beliefs and our future partial beliefs; we change our beliefs in such a way as to avoid sure losses when gambling on those beliefs (Zabell 2005, 132–33). In that sense, BMA is induction *par excellence*—we aim to make predictions of new data based on what we have observed from old data in a probabilistic manner that guarantees we do as best we can by a loss function that captures the modeler's subjective utility function on the prediction problem. The main assumption we make is that some class of models best parameterizes the predictive problem, and so we learn about those models based on the data we have observed in a diachronically coherent manner. Then, when we make our predictions, we factor what we have learned about those models into our predictions.

While MOE certainly does induction too, there is an additional element not present in the Bayesian approach. MOE partitions the input feature space and then through the gating function routes predictive samples to experts based on where they fall in that partition. This is termed a divide-and-conquer strategy or conditional computation since the MOE learns how best to divide a problem into sub-problems that can more easily be addressed by expert models. In contrast, BMAs learn no such partition, nor do they break down the problem into more easily addressable sub-problems. This divide-and-conquer approach is chiefly responsible for MOE's expressive advantage over ensemble methods.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

We argue that the MOE approach to the problem of prediction in machine learning is exactly an approach germane to abductive inference or hypothesis construction.

What we mean here by abductive inference or abduction is *hypothesis construction* or *generation*. The problem of hypothesis construction was articulated by C. S. Peirce as a form of conjecturing (Peirce 1901). He thinks that this is perhaps the most important element of scientific inquiry, and he discusses several principles to be applied when conjecturing hypotheses. Among those to consider is the economies of research—the resources spent finding hypotheses and testing them—which are important because we should expect our candidate hypotheses to break down and our time and resources in inquiry limited. He proposes several qualities to guide hypothesis selection (Peirce 1901, 109). One in the current context stands out. He calls this "Caution" and what he means is adopting a strategy of hypothesis formation that breaks down the problem space into regions that allow for groups of hypotheses to be quickly tested and discarded by their sub-components to converge at the best candidate in the most expeditious manner, and he proposes to illustrate the idea with a game. While there is some ambiguity about what Peirce means here, one plausible interpretation is that hypothesis construction should proceed with an eye toward the logical parts of the hypothesis and how they can be structured in a way to facilitate a divide-and-conquer search when testing the hypothesis. We select a hypothesis based on how fruitfully its components divide up the testing space. That selection strategy functions as an integral component in abductive reasoning.

The same divide-and-conquer strategy goes on in MOE at the time of inference where a complex sample is routed to a region where a smaller, simpler component can be used to make a prediction. An MOE model as a hypothesis breaks down the prediction problem into sub-problems with the learned partition, and then it deploys the relevant experts to make its predictions. Similarly, it also applies this divide-and-conquer strategy in reverse when constructing the MOE model, since the structure of the model architecture and training environment ensures that an MOE learns a good way to divide up the input space to make it tractable for the simpler, expert predictors. In both cases, a new hypothesis is constructed with an eye toward its simpler logical components and the input space—much akin to the strategy Peirce gave for picking cautionary, economical hypotheses.

How new hypotheses are constructed can be easily observed by the limiting case of top-expert MOEs. There the new hypothesis given by the MOE model is just a piecewise function of the expert models based on the partition the gating function learns. This is a more complex hypothesis, as we argued in section 4, since it can fit more complex data than the individual components. And crucially, it is not simply a weighted combination of the expert models due to the gating function: models are only applied based on the input and where that input falls in the partition learned by the gating function. A piecewise function is typically not just a replication or Boolean of its parts. So at least in the case of the top-expert MOE, something truly new is produced.

We can iterate the partitioning strategy to sequentially divide and conquer the prediction problem—leading to even more expressive hypotheses. Hierarchical mixtures of experts (Jordan and Jacobs 1994) does exactly this by applying gating functions sequentially to form a tree structure where the experts act as leaves; each gate partitions the cell selected by the previous gate, eventually terminating in an

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

application of the relevant experts. This process is akin to asking questions that rule out more and more alternatives until the best hypothesis is lighted upon.

The upshot is that MOE provides a formal model of abduction through a notion of functional capacity given by the framework of statistical learning theory. This is a novel formal model compared to existing models like Schurz (2008)—see Niiniluoto (2018) for a thorough discussion of various models. Its algorithm allows the sequential creation of "complex" hypotheses out of "simpler" hypotheses. These hypotheses are fundamentally different and new compared to their components as characterized by their relative VC dimension; consequently, we can use statistical learning theory to capture how other algorithms might provide substantially new hypotheses like how our analysis here captures MOE's ability to construct new hypotheses out of old ones. So along with providing a formal model of abduction through its divide-and-conquer strategy, MOE provides a test case for using additional formal tools to capture what it means for hypotheses in a new algebra to be fundamentally more expressive than those in an old algebra.

One thing that should be noted is that the abductive process used in MOEs has an element of induction. Namely, the gating function and experts are learned simultaneously by how well the MOE as a whole predicts the training data. This suggests that the process of hypothesis formation is intimately tied to having a positive feedback loop with how well the hypothesis can be used experimentally; forming hypotheses requires putting hypotheses to the test and using the evidence from those tests to guide in further hypothesis selection. This is a point that Peirce repeatedly hits upon, and we see that this virtuous process plays out in the case of MOEs.

In short, MOEs embody a form of hypothesis construction. That hypothesis construction can be seen in the limiting case where we have a top-expert MOE with homogenous experts; we have a new hypothesis constructed as a piecewise function of simpler hypotheses that has provably more functional capacity than those original hypotheses. Importantly, the construction of new hypotheses proceeds in combination with inductive reasoning where the gating function and experts are learned at the same time by how well they predict the training data. This enables beneficial learning due to how hypothesis construction can be guided by inductive feedback on the training data—leading to better hypotheses through the pressure of gradient descent or another learning algorithm.

7. Conclusion

We have considered the question of why MOE methods perform better than ensemble methods such as BMA—even though BMA has certain optimality guarantees. We argued that in a limiting case, MOE has a greater functional capacity, in the precise sense of VC dimension, than ensembles like BMAs. We then demonstrated with a series of experiments that this limiting case seems to track the more usual cases machine learning researchers encounter. The philosophical upshot is that MOEs seem to employ a type of abductive reasoning in the Peircean sense of hypothesis construction; they use a divide-and-conquer strategy to compose new, more complex hypotheses out of simpler hypotheses in an economical manner. This explains the discrepancy between Bayesian methods and MOEs because the former are fundamentally just inductive methods while the latter can in some sense be said

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

to construct the hypothesis spaces that the former rely upon. It is also notable because it gives philosophers a novel formal model of an abductive reasoning process, drawn from statistical learning theory. The upshot is that statistical learning theory too can provide tools for generating and evaluating hypothesis creation algorithms.

We conclude by discussing two open questions. Do these results imply that MOEs are just better than BMAs? Can the greater functional capacity of MOEs pose inductive problems?

Returning to the question about the superiority of MOE over BMA, one might conclude from this discussion that MOE is the superior algorithm in machine learning over BMA. We think this is premature. We have argued that the superiority of MOE over BMA is due to the former applying a form of abductive reasoning in combination with inductive reasoning while BMA is inductive only. However, the inductive methods in BMA are at least as good as the inductive methods used in MOE—if not better; this means that MOE could likely be improved by incorporating BMA into the training process through which gates and experts are learned. Instead of using a single or a finite subset of experts who specialize in soft partitions of the feature space, there could be a benefit of employing BMAs here. Furthermore, a BMA gating function might also improve the learning process. The upshot is that the abductive process captured in MOEs could be further improved by incorporating a superior inductive process in MOE model formation.

Lastly, we have argued that the superior performance of MOE over BMAs is fundamentally due to BMAs *underfitting* the data. The greater VC dimension of MOEs allows them to separate more complex datasets, which can be seen in our experiments involving binary classifications generated by arbitrary polynomials. But this greater expressiveness comes at a cost: it means that MOEs might be more prone to *overfitting* the training data, i.e., "memorizing" the data. This poses an inductive problem that we have not solved here. Namely, the greater functional capacity of MOEs should be more prone to overfitting. Whether they do or not depends on the data they are trained on, and the problem before them. It is an open question, which we leave to future research, whether this is less of an issue because most real-world problems require the kind of higher functional capacity found in MOEs than in ensembles. Maybe the world is just really complex—rendering the worry about overfitting an unfulfilled hypothetical. On that issue, we remain silent here.

Supplementary material. To view supplementary material for this article, please visit https://doi.org/10.1017/psa.2025.10159.

Acknowledgments. We would like to thank Daniel Herrmann, Gerard Rothfus, Ben Levinstein, Javier Gomez-Lavin, and two anonymous reviewers for feedback and useful comments.

Funding. None to declare.

Declarations. None to declare.

References

Betts, Sean. 2024. "Peering Inside GPT-4: Understanding Its Mixture of Experts (MoE) Architecture". https://medium.com/@seanbetts/peering-inside-gpt-4-understanding-its-mixture-of-experts-moe-architecture-2a42eb8bdcb3 [accessed 7 May, 2024].

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

Breiman, Leo. 1996. "Bagging Predictors". *Machine Learning* 24:123–40. https://doi.org/10.1007/BF00058655.
Cesa-Bianchi, Nicolo and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge: Cambridge University Press.
Dietterich, Thomas G. 2000. "Ensemble Methods in Machine Learning". In *Multiple Classifier Systems*, edited by Josef Kittler and Fabio Roli, 1–15. Berlin: Springer. https://doi.org/10.1007/3-540-45014-9_1.
Freund, Yoav and Robert E. Schapire. 1996. "Experiments with a New Boosting Algorithm". In *ICML'96: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, edited by Lorenza Saitta, 148–56. San Francisco, CA: Morgan Kaufmann.
Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. vol. 2. New York: Springer.
Herrmann, Daniel A. 2020. "PAC Learning and Occam's Razor: Probably Approximately Incorrect". *Philosophy of Science* 87 (4):685–703. https://doi.org/10.1086/709786.
Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. "Multilayer Feedforward Networks Are Universal Approximators". *Neural Networks* 2 (5):359–66. https://doi.org/10.1016/0893-6080(89)90020-8.
Jacobs, Robert A., Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. "Adaptive Mixtures of Local Experts". *Neural Computation* 3 (1):79–87. https://doi.org/10.1162/neco.1991.3.1.79.
Jiang, Albert Q., Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. "Mixtral of Experts". Preprint, arXiv:2401.04088.
Jiang, Wenxin. 2000. "The VC Dimension for Mixtures of Binary Classifiers". *Neural Computation* 12 (6):1293–1301. https://doi.org/10.1162/089976600300015367.
Jordan, Michael I. and Robert A. Jacobs. 1994. "Hierarchical Mixtures of Experts and the EM Algorithm". *Neural Computation* 6 (2):181–214. https://doi.org/10.1162/neco.1994.6.2.181.
Kearns, Michael J. and Umesh Vazirani. 1994. *An Introduction to Computational Learning Theory*. Cambridge, MA: MIT Press.
Masoudnia, Saeed and Reza Ebrahimpour. 2014. "Mixture of Experts: A Literature Survey". *Artificial Intelligence Review* 42:275–93. https://doi.org/10.1007/s10462-012-9338-y.
Murphy, Kevin P. 2022. *Probabilistic Machine Learning: An Introduction*. Cambridge, MA: MIT Press.
Niiniluoto, Ilkka. 2018. *Truth-Seeking by Abduction*. New York: Springer.
Peirce, Charles Sanders. 1901. "On the Logic of Drawing History from Ancient Documents, Especially from Testimonies". *The Essential Peirce, 1893–1913* 2:75–114.
Robert, Christian P. 2007. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. vol. 2. New York: Springer.
Rokach, Lior. 2010. "Ensemble-Based Classifiers". *Artificial Intelligence Review* 33:1–39. https://doi.org/10.1007/s10462-009-9124-7.
Schapire, Robert E. 1990. "The Strength of Weak Learnability". *Machine Learning* 5:197–227. https://doi.org/10.1007/BF00116037.
Schurz, Gerhard. 2008. "Patterns of Abduction". *Synthese* 164 (2):201–34. https://doi.org/10.1007/s11229-007-9223-4.
Schurz, Gerhard. 2019. *Hume's Problem Solved: The Optimality of Meta-induction*. Cambridge, MA: MIT Press.
Shalev-Shwartz, Shai and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press.
Shazeer, Noam, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". Preprint, arXiv:1701.06538.
Sterkenburg, Tom F. 2023. "Statistical Learning Theory and Occam's Razor: The Argument from Empirical Risk Minimization". Preprint, arXiv:2312.13842.
Wald, Abraham. 1947. "An Essentially Complete Class of Admissible Decision Functions". *The Annals of Mathematical Statistics* 549–55. https://doi.org/10.1214/aoms/1177730345.

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press

Yuksel, Seniha Esen, Joseph N. Wilson, and Paul D. Gader. 2012. "Twenty Years of Mixture of Experts". *IEEE Transactions on Neural Networks and Learning Systems* 23 (8):1177–93. https://doi.org/10.1109/TNNLS.2012.2200299.

Zabell, Sandy L. 2005. *Symmetry and its Discontents: Essays on the History of Inductive Probability*. Cambridge: Cambridge University Press.

Cite this article: Rushing, Bruce. 2026. "Peirce in the Machine: How Mixture of Experts Models Perform Hypothesis Construction." *Philosophy of Science* 93 (1):123–142. https://doi.org/10.1017/psa.2025.10159

https://doi.org/10.1017/psa.2025.10159 Published online by Cambridge University Press