Contextualized Machine Learning

Benjamin Lengerich[*†1,2], Caleb N. Ellington[*‡3], Andrea Rubbi[5], Manolis Kellis[1,2], and Eric P. Xing[3,4]

[1]Massachusetts Institute of Technology
[2]Broad Institute of MIT and Harvard
[3]Carnegie Mellon University
[4]Mohamed Bin Zayed University of Artificial Intelligence
[5]University of Cambridge

October 18, 2023

We examine Contextualized Machine Learning (ML), a paradigm for learning heterogeneous and context-dependent effects. Contextualized ML estimates heterogeneous functions by applying deep learning to the meta-relationship between contextual information and context-specific parametric models. This is a form of varying-coefficient modeling that unifies existing frameworks including cluster analysis and cohort modeling by introducing two reusable concepts: *a context encoder* which translates sample context into model parameters, and *sample-specific model* which operates on sample predictors. We review the process of developing contextualized models, nonparametric inference from contextualized models, and identifiability conditions of contextualized models. Finally, we present the open-source `PyTorch` package ContextualizedML.

1 Introduction

Contextualized ML (Figure 1) aims to learn the meta-effects of contextual information on parametric context-specific models by estimating a context encoder which translates sample context into sample-specific models. By embracing the heterogeneity and context-dependence of natural phenomena, contextualized ML provides representational capacity while retaining the glass-box nature of statistical modeling. Contextualized models can be learned by simple end-to-end backpropagation because they are composed of differentiable building blocks. In the following, we study this paradigm, analyze identifiability and nonparametric inference through Contextualized ML, and provide a Python toolkit.

---

[*]Equal contribution
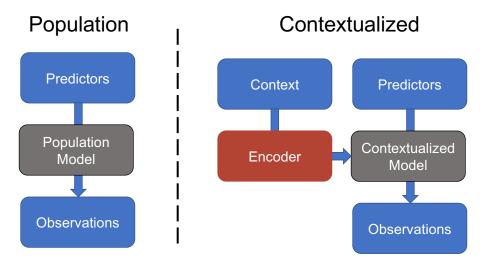[†]`blengeri@mit.edu`
[‡]`cellingt@cs.cmu.edu`

Figure 1: Contextualized paradigm. Rather than using a single population model which operates identically for all contexts, Contextualized ML estimates a locally-optimal model for each context. The heterogeneity of the population is captured by an encoder that translates sample context into sample-specific parameter values. The contextualized model has the advantage of sharing information across the population while reducing model bias.

1.1 Motivation

Modern applications of artificial intelligence are often characterized by training unconstrained ML models on large datasets. These datasets are composed of overlapping groups of samples, either explicitly (e.g. the large dataset is created by combining multiple datasets) or implicitly (e.g. the samples belong to latent sub-populations). To be generalizable, population models tend to prefer global patterns over localized effects, a problem when localized effects are critical to understanding complex processes such as in applications to computational biology (e.g. samples comprise latent cell types) and precision medicine (e.g. patients comprise latent disease subtypes). When faced with a localized effect, population-level models can either ignore the effect or encode the localized effect as an interaction of input variables. Neither of these solutions are attractive: ignoring the effect is high-bias while fitting unrestricted interactions is high-variance.

Thus, we propose to use meta-models to generate context-specific parametric models. In this way, we can reason about the context-specific parameters and summarize meta-phenomena as explicit meta-models. This strategy often allows one to tackle ML challenges with more interpretable models, rather than trying to improve results by gathering more data or by opting for more complex models.

Towards Precision Medicine Precision medicine seeks to understand the patterns of differentiation between patients such that appropriate care can be provided for each individual. However, cohort-level models estimate the same effects for all patients in a cohort, ignoring sub-cohort heterogeneity. Since patients have different histories, environments, and disease sub-types, cohort-level models cannot appropriately model the patient journeys. As [1] found in clinical evaluation of a predictive model: "Some [doctors] voiced strong concerns that using [a ML model] was the same as applying 'populational statistics' to individual patient decision making. They felt this was unethical." Thus, we seek to estimate models which adapt to patient context and drive personalized understanding. By estimating model parameters as functions of sample context, we can make principled sample-specific inferences.

Towards Intelligible Artificial Intelligence Some applications of AI are limited due to strict requirements for intelligible and transparent decisions. Large population-level models with many implicit interaction effects can be difficult to interpret, and while post-hoc procedures to approximate the large model with locally-interpretable models [2] can provide approximations of the model, such approximations do not guarantee capturing the exact behavior of the population model. We propose to approach the same endpoint more directly: by learning contextualized models from the beginning, we achieve direct interpretability without requiring post-hoc interpretation of a black-box model.

Motivating Example Let us review the motivating example of [3]: understanding election outcomes at the local level. Given candidate representations, we wish to predict and understand the factors driving the candidate's vote proportion in a particular locality (e.g. county, township, district, etc.). One approach would be to partition the dataset into similar localities and then estimate cohort-specific models for each partition. Unfortunately, by building independent models for each county, we would fail to share information between related counties, forcing us to pool together some localities with fewer samples even though they may have distinct characteristics. This simultaneous loss of power and predictive accuracy is typical of modeling large, heterogeneous datasets with homogeneous models.

Alternatively, instead of seeing these localities as discrete groups, we may embrace the data heterogeneity by modeling the ith county using a regression model $f(X_i; \Phi(C_i))$, where $\Phi(\cdot)$ is a parameter-generating function. This *contextualized* modeling allows us to train accurate models using only a single sample from each county—this is useful in settings where collecting more data may be expensive (e.g. biology and medicine) or impossible (e.g. elections and marketing). By allowing the context to be sample-specific, f no longer needs to be complex, and simple linear and logistic regression models will suffice, providing useful and interpretable models for each sample.

2 Contextualized Machine Learning

Contextualized ML estimates heterogeneous effects as distributions that adapt to context:

$$Y|X \sim \mathbb{P}_{\Phi(C)}. \tag{1}$$

That is, contextual data C is transformed into a conditional distribution by a learnable function Φ. For example, in this notation, the linear varying-coefficient model $Y|X \sim \mathrm{N}(X\beta C^T, \sigma^2)$ [4] becomes the contextualized regression model:

$$\Phi(C) := \beta C^T, \qquad \mathbb{P}_\Phi = \mathrm{N}(X\Phi(C), \sigma^2),$$

with $\beta \in \mathbb{R}^{p \times m}$ transforms context $C^T \in \mathbb{R}^{m \times 1}$ into sample-specific parameters. The free parameters are β and σ^2, values of which can be estimated by backpropagation. This can be extended to accommodate heteroskedastic noise, e.g. by modeling noise as a separate function of context

$$\Phi(C) := (\beta C^T, \phi C^T), \qquad P_\Phi = \mathrm{N}(X\Phi(C)_1, \Phi(C)_2),$$

or uncertainty in the sample-specific parameters, e.g. by a simple mixture

$$\Phi(C) := (\beta_1 C^T, \ldots, \beta_m C^T), \qquad P_\Phi = \sum_{i=1}^m \mathrm{N}(X\Phi(C)_i, \sigma^2).$$

In this way, Contextualized ML amplifies the varying-coefficient paradigm by applying the power of deep learning and auto-differentiation.

2.1 Contextualizing Models

The general approach to designing contextualized versions of cohort-based estimators is summarized in Figure 2. There are two potentially difficult steps in this process: defining a differentiable objective function, and designing a context encoder which operates on a tractable model solution space. While differentiable objective functions are problem-specific, there are a few general tricks which can often be used to improve the learnability of the deep context encoder $\Phi(C)$.

How to Contextualize Models with Deep Learning

1. Define a differentiable objective for your model of interest

$$\hat{\theta} = argmin_\theta \, \ell(X, \theta)$$

$X \in \mathbb{R}^{n \times p}$

2. Replace model parameters with a differentiable context encoder

$$\widehat{\Phi} = argmin_\Phi \sum_i^n \ell(X_i, \Phi(C_i))$$
$$[\hat{\theta}_1, \ldots, \hat{\theta}_n] = \widehat{\Phi}(C)$$

$C \in \mathbb{R}^{n \times c}$
$\Phi(c) : \mathbb{R}^c \to \mathbb{R}^{|\theta|}$

3. (Optional) Re-parameterize the context encoder with archetypes to reduce the model solution space

$$\Phi(c; \phi, A) := \sum_{k=1}^K \phi(c)_k A_k$$
$$\hat{\phi}, \hat{A} = argmin_{\phi, A} \sum_i^n \ell(X_i, \Phi(C_i; \phi, A))$$

$K \ll |\theta|$
$A \in \mathbb{R}^{K \times |\theta|}$
$\phi(c) : \mathbb{R}^c \to \mathbb{R}^K$

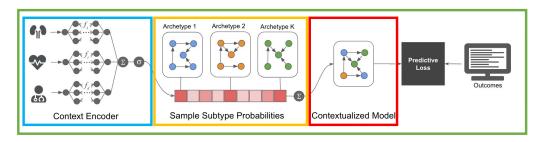4. Optimize with your favorite auto-differentiation library

Figure 2: How to contextualize a cohort-based estimator. (1) Define a differentiable objective function for each sample-specific model (red). (2) Define a differentiable context encoder to generate sample-specific parameters (blue). (3) Re-parameterize the context encoder to reduce solution space (yellow). (4) Optimize end-to-end (green).

Restrictive Context Encoders   Using a smaller class of context encoders can improve estimation. In practice, surprisingly simple forms of models can often be effectively used. For example, neural additive models [5], which are differentiable forms of additive models, can be used as context encoders to eliminate interaction effects between contextual features and enable feature-specific interpretability of context–parameter links.

Archetype-Based Modeling   Archetype-based modeling can reduce the dimensionality of the output of Φ. By representing sample-specific models as weightings of k archetypes the context encoder only needs to output a vector of size k, rather than the full model parameterization.

$$\Phi(C) := \sum_{k=1}^K \phi(C)_k A_k$$

4

Furthermore, by restricting the archetype weightings to be non-negative and sum to 1 (e.g. by applying a softmax to ϕ), the sample-specific models are a convex combination of the archetypes and can be interpreted as subtype probabilities with archetypes corresponding to subtype extrema.

Regularizing Toward Population Models By simultaneously modeling all contexts, contextualized models can be encouraged to stay closer to the population model. Let $t \in T$ index a task-specific distribution $\mathbb{P}_t(Y|X)$. Multitask learning [6, 7] seeks to improve the estimation of each $\mathbb{P}_t(Y|X)$ by sharing power between distinct tasks t. Theorem 2 of [8] shows that the task-specific distribution is the sum of the overall distribution and a task-specific pure interaction effect:

$$Y|X, t = Y|X + \rho(Y|X, t) \tag{2}$$

where $\rho(Y|X, t)$ is a pure interaction effect [9]. Thus, as sample context is an implicit task specifier, we see that context-specific estimators $Y|X, C$ also provide estimates of $Y|X$ and $\rho(Y|X, C)$ and so by regularizing against interactions between X and C (e.g. with Dropout [10]), we can encourage similarity in the context-specific distributions and encourage them to be closer to the population model. Finally, we can use the purification [9] to recover the task-specific interaction and the main effects from $Y|X, C$.

2.2 Related Work

One of the earliest ways to model sample-specific parameters as the output of a learnable function was the linear varying-coefficients (VC) model [4] in which regression parameters are produced by a linear function of covariate values, e.g. $f(x; z) = \langle x, \theta z \rangle$, with $\theta \in \mathbb{R}^{P \times K}$ for $x \in \mathbb{R}^P$, $z \in \mathbb{R}^K$. In short, contextualized ML combines the adaptability of the VC model with the power of modern ML architectures by using deep neural network as context-encoders. This combined approach was first proposed to improve interpretability of deep learning models [11] and has achieved good performance on varied tasks including survival prediction [12] and language modeling [13]. There have also been attempts to provide a nonparametric parameter-generating function by distance-matching regularization [3, 14] which proposes that there is a distance metric on contextual information which approximates Euclidean distance between sample parameters (i.e. $\|\theta_i - \theta_j\| \approx d_C(C_i, C_j)$ for samples i, j). While this regularization-based scheme provides extra flexibility by obviating the requirement of a parameteric context encoder, it also precludes end-to-end training due to the lack of a differentiable context encoder.

2.2.1 Alternative Approaches

Sample-Specific Models as Deviations Recent work has also developed sample-specific estimators as independent deviations from a population model [15–18]. This is particularly useful for structured models in which prior knowledge of the graph structure can enable efficient testing of sample-specific deviations. However, estimating sample-specific models as deviations requires $\mathcal{O}(n)$ estimation procedures for n samples and does not share power between the estimators. As a result, these approaches are more applicable to domains with fewer samples and less informative contextual data.

Heterogeneous Samples Statistical tests [19–21] can identify whether a cohort contains heterogeneous samples, enabling the identification of partitions that induce accurate group-based models. This perspective is well-suited to situations with a small numbers of groups or pre-defined partitions. However, if there are many groups relative to the number of samples, group-based modeling becomes high-variance and if samples arise from a continuous combination rather than discrete partitions, group-based models become high-bias. In such situations, higher resolution via multitask learning is required.

Post-Hoc Model Interpretations   Methods of post-hoc interpretation often seek to explain complex models by estimating local approximations [22–25]. For example, Local-Interpretable Model-Agnostic Explanation (LIME) [2] constructs local interpretations for each sample by training a linear model to approximate the outputs of a black-box model in a particular neighborhood. These local models are interpretable and approximate the output of any model, but are constrained to explain only a fixed black-box population model. In contrast, contextualized regression directly estimates local models, enabling dynamic collections of models that retain local interpretability.

2.3 Benefits of Contextualized ML

In the following, we demonstrate a few benefits of contextualized ML. More details and reproducible demos for these perspectives are available in this Jupyter notebook.
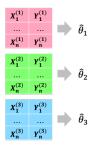
Contextualized ML Enables High-Resolution Heterogeneity   By sharing information between all contexts, contextualized learning is able to estimate heterogeneity at fine-grained resolution (Figure 3). Cluster or cohort-based models treat every partition independently, limiting heterogeneity to coarse-grained resolution where there are large enough cohorts for independent estimation. For example, this ability was exploited for context-specific Bayesian networks [26] (while cohorts models would require $\mathcal{O}(p^2)$ samples in each cohort) to reconstruct patient-specific gene expression networks.
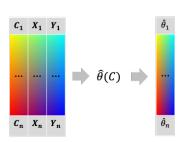

Figure 3: By sharing power between samples, contextualized ML recovers heterogeneous effects at resolutions which are finer-grained than can be done by partition-based cohort models.

Contextualized ML Interpolates Between Observed Contexts   By learning to translate contextual information into model parameters, contextualized models learn about the meta-distribution of contexts (Figure 4). As a result, contextualized models can adapt to contexts which were never observed in the training data, either interpolating between observed contexts or extrapolating to new contexts for which the meta-relationship between context and local parameters holds the same as in the training data.

Contextualized ML Enables Analysis of Latent Processes   Cluster or cohort models which are inferred by partitioning samples into groups make assumptions of IID data within each group. This approach works well when contexts are discrete, low-dimensional, and every context-specific population is well observed, but in many complex processes, contexts are continuous, high dimensional, and

Generalizes between observed contexts

Figure 4: By learning the meta-relationship between context and model parameters, contextualized ML enables interpolation between observed contexts.

sparsely observed. When cluster or cohort approaches are applied in these circumstances, downstream modeling tasks are distorted by mis-specification, where many non-IID samples are funneled into a single model. Consequently, theoretical guarantees about how well a cluster or cohort model can represent IID populations often do not apply in light of real-world heterogeneity. In contrast, contextualized learning provides a way to estimate latent, non-IID models for all samples with minimal assumptions about the grouping or clustering of these samples (Figure 5). Samples can then be grouped on the basis of model parameters and distributional differences to produce clusters in the latent model space underlying each sample. Contextualized ML intuitively recovers latent structures underlying data generation in a way *a priori* clustering cannot. Allowing downstream models to determine the grouping of samples rather than upstream contexts replaces traditional cluster analysis with contextualized analysis clusters.
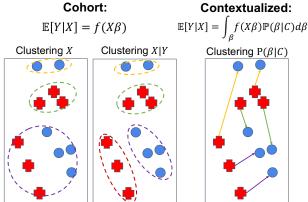
Uncovers Latent Processes

Figure 5: By estimating a contextualized model for each sample, contextualized ML uncovers important factors and latent processes in heterogeneous populations.

2.4 Python Package

Contextualized GLMs are implemented in ContextualizedML with easy interfaces. These GLMs take the form

$$\mathbb{E}[Y|X, C] = f\left(X\Phi(C)\right), \tag{3}$$

where $\Phi(C)$ is a deep context encoder, For example, contextualized linear regression:

$$\mathbb{E}[Y|X, C] = X\Phi(C), \tag{4}$$

is available by the `ContextualizedRegressor` class:

```
from contextualized.easy import ContextualizedRegressor
model = ContextualizedRegressor()
model.fit(C_train, X_train, Y_train)
```

Similarly, contextualized logistic regression:

$$\mathbb{E}[\Pr(Y = 1)|X, C] = \sigma(X\Phi(C)) \tag{5}$$

is available by the `ContextualizedClassifier` class:

```
from contextualized.easy import ContextualizedClassifier
model = ContextualizedClassifier()
model.fit(C_train, X_train, Y_train)
```

Common constructor keywords include:

- `n_bootstraps`: Number of bootstrap resampling trajectories to use.

- `encoder_type`: `mlp`, `ngam`, or `linear`, which type of model to make as context encoder. Alternatively, users may pass in their own encoder as a `PyTorch` module.

- `loss_fn`: A function to calculate loss.

- `alpha`: non-negative float, regularization strength.

- `mu_ratio`: float in range (0.0, 1.0), governs how much the regularization applies to context-specific parameters or context-specific offsets.

- `l1_ratio`: float in range (0.0, 1.0), governs how much the regularization penalizes ℓ_1 vs ℓ_2 parameter norms.

Common fitting keywords include:

- `max_epochs`: positive number, the maximum number of epochs to fit. Early stopping is turned on by default.

- `learning_rate`: positive float, default is 1e-3.

- `val_split`: float in range (0.0, 1.0), how much of the data to use for validation (early stopping).

3 Nonparametric Inference from Contextualized Models

Contextualized ML provides a framework to estimate nonparametric densities by viewing the composite densities as combinations of local parametric distributions. Let us consider a regression $Y|X \sim p(f(X))$. This regression may be considered nonparametric in two non-exclusive respects:

- the transmission function f may not be well-represented by a parametric family, or

- the distribution p may not be well-represented by a parametric family.

Contextualized linear models can be used to recover either of these forms of nonparametric models.

Contextualized Linear Models Represent Nonparametric Transmission Functions   First, contextualized linear models can represent nonparametric transmission functions by allowing coefficients to vary with context. As $\Phi(C) = \mathbb{E}_{X|C}[\frac{\partial \mathbb{E}[Y|X,C]}{\partial X}]$, we can view Φ as a differential expression describing $\frac{\partial \mathbb{E}[Y|X,C]}{\partial X}$ and reconstruct smooth, differentiable transmission functions by stitching together context-specific linear transmission functions (Figure 6). This approach approximates a conditional mixing distribution $\Gamma(x) = \sum_{k=1}^{K} \lambda_k(x)\gamma_k(x)$ of K true mixtures by fitting an overfitted mixture of $L \gg K$ atoms and then clustering these L atoms into K groups such that each group approximates γ_k. Based on this clustering, we can define a new mixing measure whose atoms are close to some γ_k for each k. This mixing measure will converge to Γ as $L \to \infty$, allowing us to approximate Γ to arbitrary precision. This framework is illustrated in Figure 6 and retains identifiability of the nonparametric transmission functions under reasonable assumptions of component separation [27].
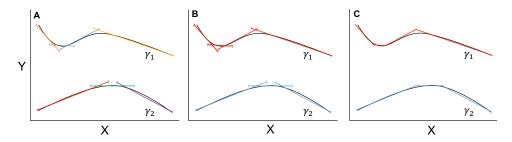

Figure 6: **(A)** We observe data arising from a mixture of nonparametric context-specific densities γ_1 and γ_2, then fit context-specific regression functions. **(B)** These context-specific atoms can be clustered and **(C)** smoothed into component to produce a nonparametric mixture model. The clustering recovers γ_1, γ_2 if the components are well-separated.

Contextualized Models Represent Non-Gaussian Outcomes   Second, contextualized models can represent non-Gaussian outcomes by summing context-specific Gaussian distributions. As locally-Gaussian distributions are universal approximators [27], any outcome distribution can be constructed by combining context-specific Gaussian distributions. If $Y|X,C$ is not well-approximated as a Gaussian distribution, we can *pseudo-sample* extra noise variables Z which localize the distribution such that $Y|X,C,Z$ is well-approximated as a Gaussian (Figure 7). In an extreme case, each value of Z can identify an individual training sample with corresponding locally-Gaussian outcome distributions that sum to form a meaningful composite distribution. As with many latent variable problems, in test samples we cannot identify which value of Z would be most correct; by integrating over all pseudo-sampled values of Z we can reconstruct the nonparametric uncertainty.
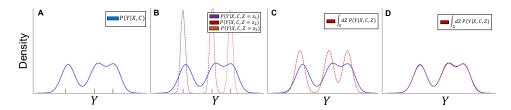
Figure 7: Pseudo-sampling procedure for representing nonparametric distributions. **(A)** The density $Y|X, C$ may not be well-approximated by a Gaussian distribution. **(B)** To overcome this, we can extend context by introducing noise variable Z to psuedo-sample localized overfitted distributions centered at each sample observation (red vertical tick marks along the horizontal axis). **(C)** By integrating over the introduced noise variable, we approximate the nonparametric distribution. **(D)** Approximation improves as the number of observations increases.

4 Identifiability of Contextualized Models

When seeking to understand contextualized models, we are interested in questions of identifiablity: how many sets of sample-specific models could equivalently recapitulate the observed data? For example, we know that both population and group-level linear models are identifiable in common conditions [28–30], but sample-specific models without covariates or constraints are not identifiable. Does the process of generating contextualized models from a shared context encoder induce identifiability? Here, we present an informal, graphical view of identifiability of contextualized models that suggests that identifiability is influenced by the flexibility of *both* the context encoder and the sample-specific models.

Notation   Let us consider a sample-specific model class parameterized by $\theta \in \mathcal{H} \subset \mathbb{R}^p$ which induces solution space $s(x, y) = \{\theta \in \mathcal{H} : h(x; \theta) = y\}$ for sample x, y. A dataset $\mathcal{D} = C, X, Y = [(C_1, X_1, Y_1), \ldots, (C_n, X_n, Y_n)]$ induces a list of solution spaces $S(\mathcal{D}) = [s(X_1, Y_1), \ldots, s(X_n, Y_n)]$. For context encoders parameterized by $\phi \in \mathcal{G} \subset R^m$, let $G(\mathcal{D}) = \{\phi \in \mathcal{G} : g(C_i; \phi) \in s(X_i, Y_i) \, \forall \, i \in [1, \ldots, n]\}$ be the set of allowable context encoders for this dataset. When $|G(\mathcal{D})| \leq 1$, there is at most one context encoder which maps each sample's context observation to its corresponding solution space, and we can say that the contextualized models are identifiable for this dataset.

Identifiability of Population Models   As a comparison, let us first consider population models from this perspective. Population models, which share θ for all samples, can be seen as constant context encoders: $g(c; \phi) = \phi$. Thus, the set of allowable context encoders for a dataset is $G_{\text{pop}}(\mathcal{D}) = \{\theta \in \mathcal{H} : h(X_i; \theta) = Y_i \, \forall \, i \in [1, \ldots, n]\} = \bigcap_{i=1}^{n} s(X_i, Y_i)$, i.e. identifiability of a population model is defined by the size of the intersection of the sample-specific solution spaces. For example, identifiability of a linear regression model is determined by how many sample-specific solution spaces (hyperplanes) coincide and how many intersect: if p solution spaces intersect, the linear model of p variables is identifiable. For linear regression with $p = 2$, $n \geq 2$ is sufficient to provide identifiability (Figure 8A). For linear regression with $p = 3$, the sample-specific solution spaces have 2 degrees of freedom and hence 2 samples can only constrain $G_{\text{pop}}(\mathcal{D})$ to a 1-dimensional subspace (Figure 8B).

Identifiability of Contextualized Models   For contextualized models, we are interested in the set of allowable context encoders for each sample: $\phi^*(c, s) = \{\phi \in \mathcal{G} : g(c; \phi) \in s\}$. The intersection of these sample-specific sets of allowable context encoders determines the allowable context encoders for the data: $G_{\text{contextualized}}(\mathcal{D}) = \bigcap_{i=1}^{n} \phi^*(C_i, s(X_i, Y_i))$. The dimension of $\phi^*(c, s)$ is upper-bounded by the product of the dimension of s and a measure of the redundancy in the context encoder (how many

10

Figure 8: Graphical depiction of identifiablity. **(A-B)** Population models are defined by the intersection of sample-specific solution spaces. In each pane, we have two solution spaces $s(X_1, Y_1)$ and $s(X_2, Y_2)$ with their intersection marked in yellow. If each sample-specific solution space has p dimensions of freedom, then identifiability requires p intersecting solution spaces. **(C-D)** Contextualized models are defined by the intersection of the allowable context encoder spaces, and hence can lose identifiability for either of two reasons: excess flexibility in the context encoder ϕ or excess flexbility in the sample-specific solution spaces s.

ways can each solution be generated). Ignoring pathological collinearity, this suggests a simple heuristic for contextualized identifiability: $n > d_g d_s$, where d_g is the degree of redundancy in the context encoder class and d_s is the number of degrees of freedom in each solution space s.

A few examples may make this heuristic more concrete. For population models, $d_g = 1$ (only a constant function can return the same value for all inputs), and hence identifiability of population models are determined by the number of degrees of freedom in the solution space. For contextualized linear models, $d_s = p-1$, suggesting that $n > d_g(p-1)$ is a useful criterion for identifiability of contextualized linear models. For linear varying-coefficients models, this criterion becomes $n > m(p - 1)$, which can be compared to traditional identifiability criteria for linear varying-coefficients models [31–35]. With $m = 1$ and $p = 2$ (Figure 8C), 2 samples are sufficient for identifiability. Note that $m = 1$ means that the context encoder operates on a single contextual variable; this single contextual variable is typically a vector of ones to accommodate offsets. For either $m = p = 2$ (Figure 8D, left) and $m = 1, p = 3$ (Figure 8D, right), at least 3 samples are required for identifiability.

5 Discussion

We have examined Contextualized ML, a paradigm for context-specific inference of differentiable models. This framework provides a principled method for sample-specific inference and analysis of heterogeneous effects, and we have presented the package ContextualizedML to make standard tasks of

context-specific regression and context-specific network inference accessible to Python users.

Several research directions remain open. While deep learning-based context encoders and auto-differentiation libraries are useful to circumvent requirements of parametric assumptions and analytical solutions, there is no guarantee that this learning scheme is optimal. In addition, these methods rely on contextual data to accurately represent latent phenomena; extending methods to generate sample representations from more diverse data sources (e.g. foundation models) could improve the learned models. Beyond questions of estimation procedures, there are also open questions regarding the analysis of estimated sample-specific models. Once we have estimated sample-specific model parameters, what is the best way to summarize these new representations: should we cluster the estimated parameters, or is it best to present these models to users as sample-specific models? These questions scratch the surface of the wide potential that contextualized ML unlocks for improved methods of data analysis.

Acknowledgements

References

[1] Qian Yang, Aaron Steinfeld, and John Zimmerman. Unremarkable ai: Fitting intelligent decision support into critical, clinical decision-making processes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[2] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[3] Benjamin J. Lengerich, Bryon Aragam, and Eric P Xing. Learning sample-specific models with low-rank personalized regression. In *Advances in Neural Information Processing Systems (In Press)*, 2019.

[4] Trevor Hastie and Robert Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(4):757–779, 1993.

[5] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

[6] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[7] Leo Breiman and Jerome H Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):3–54, 1997.

[8] Benjamin Lengerich. *Sample-Specific Models for Precision Medicine*. PhD thesis, Carnegie Mellon University, 2020.

[9] Benjamin Lengerich, Sarah Tan, Chun-Hao Chang, Giles Hooker, and Rich Caruana. Purifying interaction effects with the functional anova: An efficient algorithm for recovering identifiable additive models. In *International Conference on Artificial Intelligence and Statistics*, pages 2402–2412, 2020.

[10] Benjamin J Lengerich, Eric Xing, and Rich Caruana. Dropout as a regularizer of interaction effects. In *International Conference on Artificial Intelligence and Statistics*, pages 7550–7564. PMLR, 2022.

[11] Maruan Al-Shedivat, Avinava Dubey, and Eric P Xing. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.

[12] Maruan Al-Shedivat, Avinava Dubey, and Eric P Xing. Personalized survival prediction with contextual explanation networks. *arXiv preprint arXiv:1801.09810*, 2018.

[13] Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. Contextual parameter generation for universal neural machine translation. *arXiv preprint arXiv:1808.08493*, 2018.

[14] Benjamin J Lengerich, Bryon Aragam, and Eric P Xing. Personalized regression enables sample-specific pan-cancer analysis. *Bioinformatics*, 34(13):i178–i186, 2018.

[15] Fattaneh Jabbari, Shyam Visweswaran, and Gregory F. Cooper. Instance-specific bayesian network structure learning. In Václav Kratochvíl and Milan Studený, editors, *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, volume 72 of *Proceedings of Machine Learning Research*, pages 169–180, Prague, Czech Republic, 11–14 Sep 2018. PMLR.

[16] Marieke Lydia Kuijjer, Matthew George Tung, GuoCheng Yuan, John Quackenbush, and Kimberly Glass. Estimating sample-specific regulatory networks. *iScience*, 14:226–240, 2019.

[17] Xiaoping Liu, Yuetong Wang, Hongbin Ji, Kazuyuki Aihara, and Luonan Chen. Personalized characterization of diseases using sample-specific networks. *Nucleic acids research*, 44(22):e164–e164, 2016.

[18] Xiang Li, Shanghong Xie, Peter McColgan, Sarah J Tabrizi, Rachael I Scahill, Donglin Zeng, and Yuanjia Wang. Learning subject-specific directed acyclic graphs with mixed effects structural equation models from observational data. *Frontiers in genetics*, 9, 2018.

[19] Jiaying Gu, Roger Koenker, and Stanislav Volgushev. Testing for homogeneity in mixture models. *Econometric Theory*, 34(4):850–895, 2018.

[20] Xin Liu, Cristian Pasarica, and Yongzhao Shao. Testing homogeneity in gamma mixture models. *Scandinavian Journal of Statistics*, 30(1):227–239, 2003.

[21] Richard Charnigo and Jiayang Sun. Testing homogeneity in a mixture distribution via the l 2 distance between competing models. *Journal of the American Statistical Association*, 99(466):488–498, 2004.

[22] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

[23] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[25] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.

[26] Ben Lengerich, Caleb Ellington, Bryon Aragam, Eric P Xing, and Manolis Kellis. Notmad: Estimating bayesian networks with sample-specific structures and parameters. *arXiv preprint arXiv:2111.01104*, 2021.

[27] Bryon Aragam, Chen Dan, Eric P Xing, and Pradeep Ravikumar. Identifiability of nonparametric mixture models and bayes optimal clustering. *arXiv preprint arXiv:1802.04397*, 2018.

[28] Olav Reiersøl. Identifiability of a linear relation between variables which are subject to error. *Econometrica: Journal of the Econometric Society*, pages 375–389, 1950.

[29] Lennart Ljung and Torkel Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265–276, 1994.

[30] Christian Hennig. Identifiablity of models for clusterwise linear regression. *Journal of Classification*, 17(2):273–296, 2000.

[31] Zongwu Cai, Jianqing Fan, and Runze Li. Efficient estimation and inferences for varying-coefficient models. *Journal of the American Statistical Association*, 95(451):888–902, 2000.

[32] CN Kuruwita, KB Kulasekera, and CM Gallagher. Generalized varying coefficient models with unknown link function. *Biometrika*, 98(3):701–710, 2011.

[33] Lili Yue, Gaorong Li, and Heng Lian. Identification and estimation in quantile varying-coefficient models with unknown link function. *Test*, 28:1251–1275, 2019.

[34] Wenyang Zhang, Degui Li, and Yingcun Xia. Estimation in generalised varying-coefficient models with unspecified link functions. *Journal of Econometrics*, 187(1):238–255, 2015.

[35] Lixia Hu, Tao Huang, and Jinhong You. Estimation and identification of a varying-coefficient additive model for locally stationary processes. *Journal of the American Statistical Association*, 114(527):1191–1204, 2019.