Information-theoretic Classification Accuracy: A Criterion that Guides Data-driven Combination of Ambiguous Outcome Labels in Multi-class Classification

Chihao Zhang¹, Yiling Elaine Chen², Shihua Zhang¹, and Jingyi Jessica Li *²,³

¹NCMIS, CEMS, RCSDS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and School of Mathematics Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
²Department of Statistics, University of California, Los Angeles, CA 90095, USA
³Interdepartmental Program in Bioinformatics, Department of Biostatistics, Department of Computational Medicine, and Department of Human Genetics, University of California, Los Angeles, CA 90095, USA

Abstract

Outcome labeling ambiguity and subjectivity are ubiquitous in real-world datasets. While practitioners commonly combine ambiguous outcome labels in an ad hoc way to improve the accuracy of multi-class classification, there lacks a principled approach to guide label combination by any optimality criterion. To address this problem, we propose the information-theoretic classification accuracy (ITCA), a criterion of outcome “information” conditional on outcome prediction, to guide practitioners on how to combine ambiguous outcome labels. ITCA indicates a balance in the trade-off between prediction accuracy (how well do predicted labels agree

*To whom correspondence should be addressed. Email: jli@stat.ucla.edu
with actual labels) and prediction resolution (how many labels are predictable). To find the optimal label combination indicated by ITCA, we develop two search strategies: greedy search and breadth-first search. Notably, ITCA and the two search strategies are adaptive to all machine-learning classification algorithms. Coupled with a classification algorithm and a search strategy, ITCA has two uses: to improve prediction accuracy and to identify ambiguous labels. We first verify that ITCA achieves high accuracy with both search strategies in finding the correct label combinations on synthetic and real data. Then we demonstrate the effectiveness of ITCA in diverse applications including medical prognosis, cancer survival prediction, user demographics prediction, and cell type classification.

*Keywords:* multi-class classification; information theory; noisy labels; supervised learning; class label combination.
1. INTRODUCTION

Machine-learning prediction algorithms play an increasingly important role in data-driven and computer-based scientific research and industrial applications, thanks to the rapid advances in data availability, computing power, and algorithm development. Prominent examples include fraud detection based on historical transactions [Brockett et al., 2002], cardiovascular risk prediction [Wilson et al., 1998, Weng et al., 2017], and risk evaluation for multiple diseases using genomics data [Chen et al., 2016]. Accurate algorithm prediction carries great promise because powerful algorithms can extract wisdom from human experts’ numerous decisions made over the years.

However, a bottleneck in the development of reliable algorithms is the availability of high-quality data, especially in medical diagnosis/prognosis and other biomedical applications. For example, medical records are inherently noisy, containing diagnostic/prognostic outcomes that are mislabeled or labeled inconsistently by graders [Krause et al., 2018]. Further, labeling ambiguity is common in ordinal outcomes—whose ordered levels represent degrees of symptom severity or treatment effectiveness—because of graders’ subjectivity in assigning patients to levels.

Ambiguous outcome labels would inevitably deteriorate the prediction accuracy of algorithms. Nevertheless, prediction accuracy may be boosted by combining the outcome labels that are hard to distinguish in training data, at the cost of losing prediction resolution because label combination would reduce the number of predictable outcome labels. Hence, how to find a balance between prediction accuracy and resolution is a computational challenge. Although outcome labels are often combined in an ad hoc way to train algorithms in practices [Feldmann and Steudel, 2000, Hemingway et al., 2013, Riley et al., 2013, Steyerberg et al., 2013], there lacks a principled approach to guide the combination by any optimality criterion.

Besides outcome prediction, another critical application of machine learning is to refine the outcome labels that are predefined by human experts. For example, in medical informatics, an important task is to use treatment outcomes to retrospectively refine diagnosis categories [Kale and Korenstein, 2018, Lindenauer et al., 2012]. This task can be formulated as a multi-class classification problem, where the features are treatment outcomes and the response is a categorical variable indicating diagnosis categories. In this task, if patients in different diagnosis categories exhibit indistinguishable treatment outcomes, these categories should be combined. Such data-
driven prediction has been used to update existing grading systems for diagnosis, such as the Gleason score for prostatic carcinoma [Epstein et al., 2015], the NYHA functional classification for heart failure [of the New York Heart Association et al., 1994], the glomerular filtration rate (GFR) grade for chronic kidney disease, and the ACC/AHA classification for high blood pressure [Muntner et al., 2018]. For another example, in single-cell gene expression data analysis, a typical procedure is to cluster cells based on gene expression levels and subsequently annotate the cell clusters using domain knowledge [Butler et al., 2018]. This procedure is inevitably subjective because how to determine the number of clusters remains a challenging question, and some cell clusters may be hardly distinguishable by gene expression levels. Hence, a principled method is called to guide the decision of combining ambiguous labels defined by the human experts.

How to find an “optimal” class combination is not a trivial problem. The reason is that, even if the prediction is completely random, i.e., assigning data points with random labels irrespective of features, prediction accuracy would still be boosted by label combination. In such an extreme case, the increase in prediction accuracy does not outweigh the decrease in prediction resolution. Hence, our rationale is that label combination must be guided by a criterion that reasonably balances prediction accuracy and resolution. Motivated by this rationale, we propose a criterion from an information theory perspective to evaluate prediction accuracy together with prediction resolution. This data-driven criterion, called the information-theoretic classification accuracy (ITCA), guides the combination of class labels given a multi-class classification algorithm. ITCA also allows choosing a multi-class classification algorithm among the available algorithms based on their respective optimal label combinations.

There are two lines of research seemingly related to our work. The first line is classification in the presence of labeling noise [Frénay and Verleysen, 2013]. It includes three major approaches for handling labeling noise: (1) using robust losses or ensemble learning [Freund, 2001, Beigman and Klebanov, 2009]; (2) removing data points that are likely mislabeled [Zhang et al., 2006, Thongkam et al., 2008]; (3) modeling labeling noise using data generative models [Swartz et al., 2004, Kim and Ghahramani, 2008]. The second line is conformal prediction [Vovk et al., 2005], whose core idea is to generalize the prediction from a single label to a set of labels [Balasubramanian et al., 2014]. In detail, given a pre-specified confidence level (e.g., 95%), conformal prediction outputs a
set of labels for each data point so that the actual label is contained in the set with probability no less than the confidence level. In summary, both lines of research tackle the labeling noise problem at the level of individual data points; as a result, neither line can suggest how to combine labels at the global level for all data points, e.g., how to combine labels in a diagnosis or prognosis system.

Unlike these two lines of research, our proposed ITCA is a global criterion that measures prediction accuracy in relation to prediction resolution. Given a multi-class prediction algorithm, say random forest, ITCA is defined as a weighted prediction accuracy, in which each data point is weighted by the entropy attributable to its class. As a result, ITCA balances the trade-off between prediction accuracy and resolution, thus offering guidance for finding an “optimal” class label combination. As the first criterion that guides the combination of class labels, ITCA has broad applications, including medical diagnosis and prognosis, cell type classification, cancer survival prediction, and user demographics prediction. We will demonstrate these applications in Section 4.

A prominent advantage of ITCA is that it is adaptive to all classification algorithms, thus allowing practitioners to choose the most suitable classification algorithm for a specific task. As a side note, one may intuitively consider using a clustering algorithm to combine similar classes; for example, one may use the $K$-means algorithm or the hierarchical clustering algorithm to cluster the $K_0$ class centers into $K < K_0$ clusters, so that the $K_0$ observed classes are correspondingly combined into $K$ classes (see Section 3). However, this intuitive approach has a drawback: since a distance metric is required to define the class centers and their distances, a gap exists between the choices of a metric and a classification algorithm. In other words, clustering-guided class combination based on a certain metric (e.g., Euclidean distance) does not guarantee to optimize the classification accuracy of a specific algorithm (e.g., support vector machine with Gaussian kernel). In contrast, ITCA does not have this drawback because it is defined based on the classification accuracy of that specific algorithm.

The rest of this paper is structured as follows. In Section 2, we first formulate the problem, define ITCA at the sample and population levels, and explain the intuition behind the definitions. Then we introduce two search strategies—greedy search and breadth-first search—to find the optimal class combination guided by ITCA given a classification algorithm. Further, we propose alternative
criteria that may also guide class combination and compare them with ITCA. In Section 3, we use extensive simulation studies to verify the effectiveness of ITCA and the two search strategies. In Section 4, we demonstrate the broad applications of ITCA by apply it to multiple real-world datasets, including prognosis data of traumatic brain injury patients, glioblastoma cancer survival data, mobile phone user behavioral data, and single-cell RNA-seq data. In these applications, we also show the versatility of ITCA in working with various classification algorithms.

2. METHOD

2.1 Problem formulation

Let \((X, Y) \sim \mathcal{P}\) be a random pair where \(X \in \mathcal{X} \subset \mathbb{R}^d\) is a feature vector, \(Y \in [K_0] := \{1, \ldots, K_0\}\) is a class label indicating one of \(K_0\) observed classes that are potentially ambiguous, and \(\mathcal{P}\) is the joint distribution of \((X, Y)\). For a fixed positive integer \(K < K_0\), a class combination is represented by an onto mapping: \(\pi_K : [K_0] \rightarrow [K]\). For example, if \(K_0 = 4\) classes are combined into \(K = 3\) classes by merging the original classes 3 and 4, then \(\pi_3(1) = 1, \pi_3(2) = 2, \pi_3(3) = 3,\) and \(\pi_3(4) = 3\). We define \(\pi_K^{-1}\) as follows: \(\pi_K^{-1}(k) := \{k_0 \in [K_0] : \pi_K(k_0) = k\}, \forall k \in [K]\). Then in this example, \(\pi_3^{-1}(1) = \{1\}, \pi_3^{-1}(2) = \{2\}, \pi_3^{-1}(3) = \{3, 4\}\). For notation simplicity, we write \(\pi_3\) as \(\{1, 2, (3, 4)\}\). Given a class combination \(\pi_K\), a classification algorithm \(\mathcal{C}\), and a training dataset \(\mathcal{D}_t\), we denote by \(\phi_{\pi_K}^{\mathcal{C}, \mathcal{D}_t} : \mathcal{X} \rightarrow [K]\) a multi-class classifier trained by \(\mathcal{C}\) on \(\mathcal{D}_t\) to predict \(K\) combined classes. The prediction is accurate if and only if \(\phi_{\pi_K}^{\mathcal{C}, \mathcal{D}_t}(X) = \pi_K(Y)\).

Given \(K\), how to find an “optimal” \(\pi_K\) is a twofold problem. First, we need an optimality criterion of \(\pi_K\) that balances the trade-off between prediction accuracy and resolution. Mathematically, given a dataset \(\mathcal{D} := \{(X_i, Y_i)\}_{i=1}^n\) a class combination \(\pi_K\), and a classification algorithm \(\mathcal{C}\), we split \(\mathcal{D}\) into training data \(\mathcal{D}_t\) and evaluation data \(\mathcal{D}_e\), train a classifier \(\phi_{\pi_K}^{\mathcal{C}, \mathcal{D}_t}\) on \(\mathcal{D}_t\), and evaluate the prediction accuracy of \(\phi_{\pi_K}^{\mathcal{C}, \mathcal{D}_t}\) on \(\mathcal{D}_e\). Then we define an optimality criterion of \(\pi_K\) given \(\mathcal{D}_t, \mathcal{D}_e,\) and \(\mathcal{C}\), denoted by \(m(\pi_K; \mathcal{D}_t, \mathcal{D}_e, \mathcal{C})\), based on the prediction accuracy of \(\phi_{\pi_K}^{\mathcal{C}, \mathcal{D}_t}\) and the resolution of \(\pi_K\)’s \(K\) combined classes. To define the resolution, we adopt the entropy concept in information theory. The entropy of \(\pi_K\)’s \(K\) combined classes’ empirical distribution in \(\mathcal{D} = \mathcal{D}_t \cup \mathcal{D}_e\) is \(\sum_{k=1}^K \left(- \sum_{k_0 \in \pi_K^{-1}(k)} \hat{p}_{k_0}\right) \log \left(\sum_{k_0 \in \pi_K^{-1}(k)} \hat{p}_{k_0}\right),\) where \(\hat{p}_{k_0}\) is the proportion of the \(k_0\)-th original class in \(\mathcal{D}\), and \(\sum_{k_0 \in \pi_K^{-1}(k)} \hat{p}_{k_0}\) is the proportion of \(\pi_K\)’s \(k\)-th combined class in \(\mathcal{D}\). Note that this
entropy measures the uncertainty of $\pi_K(Y_1), \ldots, \pi_K(Y_n)$ and increases as $K$ increases or as the balance of the $K$ classes increases (Figure 1). Hence, we find it reasonable to use this entropy to describe the resolution of $\pi_K$. In Section 2.2, we will define $m(\pi_K; D_t, D_e, C)$ as ITCA, a criterion that can be interpreted as an entropy-weighted accuracy, to balance the trade-off between prediction accuracy and resolution.

Second, given the criterion $m(\cdot; D_t, D_e, C)$, we need a search strategy to find the optimal class combination $\pi^*_K$ that maximizes the criterion:

$$\pi^*_K = \arg \max_{\pi_K \in A} m(\pi_K; D_t, D_e, C),$$

where $A$ is the set of allowed class combinations. For example, if the outcome labels are ordinal, $A$ contains all combinations that only combine adjacent class labels. We will address this search problem in Section 2.3.

Figure 1: Entropy of the class label distribution reflects the resolution of class labels. Each pie chart indicates a class label distribution. Top: the number of balanced classes increases from left to right. Bottom: the class distributions are increasingly balanced from left to right. The resolution increases from left to right in both plots, reflected by the entropy increase.
2.2 Information-theoretic classification accuracy (ITCA)

Given \( \pi_K \), a classification algorithm \( \mathcal{C} \), and a size-\( n \) dataset \( \mathcal{D} \) randomly split into training data \( \mathcal{D}_t \) and evaluation data \( \mathcal{D}_e \), we define the ITCA of \( \pi_K \) at the sample level as

\[
\text{ITCA}(\pi_K; \mathcal{D}_t, \mathcal{D}_e, \mathcal{C}) := \sum_{k=1}^{K} \left[-\left(\sum_{k_0 \in \pi_K^{-1}(k)} \hat{p}_{k_0}\right) \log \left(\sum_{k_0 \in \pi_K^{-1}(k)} \hat{p}_{k_0}\right)\right] \cdot \frac{\sum_{(X_i, Y_i) \in \mathcal{D}_e} I(\phi_{\pi_K}^{C, \mathcal{D}_t}(X_i) = k, \pi_K(Y_i) = k)}{1 \bigg/ \sum_{(X_i, Y_i) \in \mathcal{D}_e} I(\pi_K(Y_i) = k)}, \tag{2}
\]

where \( I(\cdot) \) is the indicator function, \( \hat{p}_{k_0} := \sum_{i=1}^{n} I(Y_i = k_0)/n \) is the proportion of the \( k_0 \)-th class in \( \mathcal{D} = \mathcal{D}_t \cup \mathcal{D}_e \), and \( \phi_{\pi_K}^{C, \mathcal{D}_t} \) is the classifier trained by the algorithm \( \mathcal{C} \) on \( \mathcal{D}_t \). In this definition (2), the sum is taken over the \( K \) combined classes defined by \( \pi_K \); inside of the sum, for the \( k \)-th combined class, the first term is the contribution of the class to the entropy of \( \pi_K(Y) \)'s empirical distribution in \( \mathcal{D} \), and the second term is the conditional prediction accuracy of \( \phi_{\pi_K}^{C, \mathcal{D}_t} \) for the class on the evaluation data \( \mathcal{D}_e \). In other words, the ITCA of \( \pi_K \) is a weighted out-of-sample prediction accuracy of a classifier trained with \( \pi_K \)'s \( K \) combined classes, whose contributions to the entropy (i.e., resolution) are defined as the weights. Meanwhile, the ITCA of \( \pi_K \) can be regarded as a weighted entropy if we consider the class-conditional prediction accuracies as weights. As a result, ITCA balances the prediction accuracy and resolution.

Note that the ITCA definition in (2), if \( \hat{p}_{k_0} \) is alternatively defined as the proportion of the \( k_0 \)-th class in \( \mathcal{D}_e \), is equivalent to

\[
\text{ITCA}^{\text{alt}}(\pi_K; \mathcal{D}_t, \mathcal{D}_e, \mathcal{C}) = \frac{1}{|\mathcal{D}_e|} \sum_{(X_i, Y_i) \in \mathcal{D}_e} -\log \left(\sum_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} \hat{p}_{k_0}\right) \cdot I\left(\phi_{\pi_K}^{C, \mathcal{D}_t}(X_i) = \pi_K(Y_i)\right), \tag{3}
\]

where \( |\mathcal{D}_e| \) is the size of \( \mathcal{D}_e \). This alternative definition (3) has an intuitive interpretation. Imagine that we represent the \( K_0 \) original classes in \( \mathcal{D}_e \) as \( K_0 \) non-overlapping intervals in \([0, 1]\) such that class \( k_0 \) is represented by an interval \( I_{k_0} \) of length \( \hat{p}_{k_0} \); then \( \bigcup_{k_0=1}^{K_0} I_{k_0} = [0, 1] \). With this representation, we can introduce a random variable \( U \sim \text{Uniform}([0, 1]) \) such that \( Y = \sum_{k_0=1}^{K_0} k_0 I(U \in I_{k_0}) \); that is, the event \( U \in I_{k_0} \) is equivalent to the event \( Y = k_0 \). For a data point \( X_i \in \mathcal{D}_e \), when \( \phi_{\pi_K}^{C, \mathcal{D}_t} \) correctly predicts its combined class label \( \pi_K(Y_i) \), i.e., \( \phi_{\pi_K}^{C, \mathcal{D}_t}(X_i) = \pi_K(Y_i) \), we weight this correct prediction in (3) by difficulty, which should be higher if the combined class \( \pi_K(Y_i) \) is
smaller, i.e., the total length of $\bigcup_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} I_{k_0}$ is shorter. Hence, we define the weight as the difference between the entropy of Uniform([0, 1]) (i.e., the distribution of $U_i$ without knowledge of $Y_i$) and the entropy of Uniform($\bigcup_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} I_{k_0}$) (i.e., the distribution of $U_i$ conditional on $\pi_K(Y_i)$); this difference is bigger when the total length of $\bigcup_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} I_{k_0}$ is shorter, satisfying our requirement. It can be derived that the entropy of Uniform([0, 1]) is 0, and the entropy of Uniform($\bigcup_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} I_{k_0}$) is $\log(\sum_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} \hat{p}_{k_0})$. Hence, the correct prediction of $\pi_K(Y_i)$ has the weight $-\log(\sum_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} \hat{p}_{k_0})$ as in (3).

To address the issue that ITCA is defined based on one random split on $D$ (each data point is used only once for either training or evaluation) in (2), we further define the $R$-fold cross-validated (CV) ITCA as

$$\text{ITCA}^{CV}(\pi_K; D, C) := \frac{1}{R} \sum_{r=1}^{R} \text{ITCA}(\pi_K; D^r_t, D^r_e, C),$$

(4)

where the dataset $D$ is randomly split into $R$ equal-sized folds, with the $r$-th fold $D^r_e$ serving as the evaluation data and the union of the remaining $R - 1$ folds $D^r_t$ serving as the training data. In the following text, we refer to $\text{ITCA}^{CV}$ as the ITCA criterion if without specification.

By definition, ITCA is non-negative and becomes equal to zero when $K = 1$, i.e., the degenerate case when all classes are combined as one and classification becomes meaningless. This gives ITCA a nice property: unless all predictions are wrong for all $K \geq 2$—an unrealistic scenario, the class combination that maximizes ITCA would not be the degenerate $\pi_1$.

To investigate the theoretical properties of ITCA, we define it at the population level as

$$\text{p-ITCA}(\pi_K; D_t, C) := \sum_{k=1}^{K} \left[-\mathbb{P}(\pi_K(Y) = k) \log \mathbb{P}(\pi_K(Y) = k)\right] \cdot \mathbb{P}(\phi_{\pi_K, D_t}(X) = \pi_K(Y) | \pi_K(Y) = k),$$

(5)

where $\phi_{\pi_K, D_t}$ is the classifier trained by the algorithm $C$ on a finite training dataset $D_t$. The population is used to evaluate the entropy contributions of $\pi_K$'s $K$ combined classes and the class-conditional prediction accuracies of $\phi_{\pi_K, D_t}$.

ITCA is aligned with the principle of maximum entropy [Pal et al., 2003], an application of the Occam’s razor. For a fixed $K$, if the classifier $\phi_{\pi_K, D_t}$ has the same prediction accuracy for each class, i.e., $\mathbb{P}(\phi_{\pi_K, D_t}(X) = \pi_K(Y) | \pi_K(Y) = k)$ is a constant for all $k \in [K]$, then ITCA in (2) is proportional to the entropy of $\pi_K(Y)$ and is maximized by the $\pi_K$ that results in the $K$ combined
classes with the most balanced class probabilities. In the special case when the classifier performs
perfect prediction for all \( K \)'s, i.e., \( P(\phi^{C,D}_\pi(X) = \pi_K(Y)|\pi_K(Y) = k) = 1 \) for all \( k \in [K] \) and all \( K \in [K_0] \), ITCA becomes monotone increasing in \( K \), i.e., the higher the class resolution, the larger
the entropy.

An advantage of ITCA is that it is adaptive to all machine-learning classification algorithms and
its values are comparable for different algorithms. Hence, ITCA allows users to choose the most
suitable algorithm for a specific classification task. In a task where prediction accuracy is of primary
interest, users may compare algorithms by their optimal ITCA values (whose corresponding class
combinations may differ for different algorithms) and choose the algorithm (along with the class
combination) that gives the largest optimal ITCA value. Granted, if a classification algorithm
has a sufficiently high accuracy for predicting the original \( K_0 \) classes, ITCA would not suggest any
classes to be combined. Hence, in exploratory data analysis where the goal is to find similar classes,
users may use ITCA with a weak classification algorithm (e.g., LDA) so that the class combination
found by ITCA can reveal similar classes.

2.3 Search strategy

Given the dataset \( D = \{(X_i, Y_i)\}_{i=1}^n \), we aim at finding the optimal class combination that maxi-
mizes the ITCA in (4). A naïve strategy is the exhaustive search, i.e., computing the ITCA of all
allowed class combinations \( \pi_K \)'s for \( 2 \leq K \leq K_0 \), whose set is denoted by \( \mathcal{A} \); however, \( |\mathcal{A}| \) can be
huge even when \( K_0 \), the number of original classes, is moderate (Table 1). Specifically, if the class
labels are nominal, the number of allowed combinations, \( |\mathcal{A}| \), is known as the Bell number [Bell,
1938]. If the class labels are ordinal, only adjacent classes should be combined; then, the number
of allowed combinations is \( 2^{K_0-1} - 1 \) [Feller, 2008]. Since a classification algorithm \( C \) needs to be
trained for every \( \pi_K \) to calculate the ITCA of \( \pi_K \), a large \( |\mathcal{A}| \) would make the exhaustive search
strategy computationally infeasible.

| Table 1: The number of allowed class combinations given \( K_0 \) |
|-----------------|---------|-------|---------|---------|---------|---------|---------|
| Label type      | \( K_0 \) | 2  | 4  | 6  | 8  | 12 | 16 |
| \( |\mathcal{A}| \) Nominal | 1 | 14 | 202 | 4,139 | 4,213,596 | \( \sim10^{10} \) |
| \( |\mathcal{A}| \) Ordinal  | 1 | 7  | 31  | 127   | 2,047 | 32,767 |
To design an efficient search strategy, we adopt the greedy search algorithm. Specifically, we start from $\pi^*_{K_0} = \pi_{K_0}$, which does not combine any classes. In the $k$-th round ($1 \leq k \leq K_0 - 2$), we find the best combination that maximizes the ITCA among the allowed combinations $\pi^*_{K-k}$’s, which are defined based on the chosen $\pi^*_{K-k+1}$. Next, we start from the chosen $\pi^*_{K-k}$ and repeat this procedure until the ITCA cannot be improved or $K_0 - 2$ rounds are finished. This greedy search strategy reduces the search space significantly and is summarized in Algorithm 1. For ordinal class labels, the worst-case complexity of this greedy search strategy is $O(K_0^2)$.

**Algorithm 1** Greedy search algorithm

1: Input data $D = \{(X_i, Y_i)\}_{i=1}^n$ and a classification algorithm $C$.
2: Set $K \leftarrow K_0$ and $\pi^*_K \leftarrow \pi_{K_0}$.
3: Compute $\text{ITCA}_{\text{CV}}(\pi^*_K; D, C)$.
4: while $K > 2$ do
5: Determine the set of allowed class combinations $A_{K-1}$ based on $\pi^*_K$.
6: for each allowed class combination $\pi_{K-1} \in A_{K-1}$ do
7: Compute $\text{ITCA}_{\text{CV}}(\pi_{K-1}; D, C)$.
8: if there exists no $\pi_{K-1}$ that achieves $\text{ITCA}_{\text{CV}}(\pi_{K-1}; D, C) > \text{ITCA}_{\text{CV}}(\pi^*_K; D, C)$ then
9: Break.
10: else
11: $K \leftarrow K - 1$;
12: $\pi^*_K \leftarrow \arg\max_{\pi_{K-1} \in A_{K-1}} \text{ITCA}_{\text{CV}}(\pi_{K-1}; D, C)$.
13: Return $\pi^*_K$.

It is well known that the greedy search strategy may not lead to the globally optimal class combination. Besides the greedy search, another commonly used search strategy is the breadth-first search (BFS) summarized in Algorithm 2, which uses a queue to store class combinations that may be further combined. Specifically, given $\pi_K$ (with $K \geq 3$) removed from the front of queue, we use $\mathcal{N}(\pi_K)$ to denote the set of allowed combinations $\pi_{K-1}$’s that combine any two class defined by $\pi_K$. For $\pi_{K-1} \in \mathcal{N}(\pi_K)$ that has not been visited and improves the ITCA of $\pi_K$, the BFS strategy adds $\pi_{K-1}$ to the end of the queue and considers $\pi_{K-1}$ as a candidate optimal class combination.
The BFS strategy stops when the queue is empty.

The search space of BFS is much larger than that of the greedy search but usually smaller than that of the exhaustive search. For ordinal class labels, BFS may enumerate all allowed class combinations in the worst case; hence, its worst-case complexity is \( O(2^{K_0}) \).

Although the greedy search and BFS are heuristic strategies, we will show that they achieve comparable performance to that of the exhaustive search in our experimental results in Section 3.

**Algorithm 2** Breadth-first search algorithm

1: Input data \( D = \{(X_i, Y_i)\}_{i=1}^n \) and a classification algorithm \( C \).
2: Initialize an empty set of combinations \( A \) and an empty queue \( Q \).
3: Compute \( \text{ITCA}^{CV}(\pi_{K_0}; D, C) \).
4: \( A \leftarrow A \cup \{\pi_{K_0}\} \). \( \triangleright \) Add \( \pi_{K_0} \) to \( A \).
5: \( Q \text{.enqueue}(\pi_{K_0}) \). \( \triangleright \) Add \( \pi_{K_0} \) to the back of \( Q \).
6: while \( Q \) is not empty do
7: \( \pi_K \leftarrow Q\text{.dequeue()} \). \( \triangleright \) Remove the front element of \( Q \) as \( \pi_K \).
8: Determine the set of allowed class combinations \( N(\pi_K) \) based on \( \pi_K \).
9: for each allowed class combination \( \pi_{K-1} \in N(\pi_K) \) do
10: if \( \pi_{K-1} \) is not visited then
11: Compute \( \text{ITCA}^{CV}(\pi_{K-1}; D, C) \).
12: if \( \text{ITCA}^{CV}(\pi_{K-1}; D, C) > \text{ITCA}^{CV}(\pi_K; D, C) \) then
13: \( A \leftarrow A \cup \{\pi_{K-1}\} \). \( \triangleright \) Add \( \pi_{K-1} \) to \( A \).
14: \( Q\text{.enqueue}(\pi_{K-1}) \). \( \triangleright \) Add \( \pi_{K-1} \) to the back of \( Q \).
15: Mark \( \pi_{K-1} \) as visited.
16: Return \( \pi^*_K \leftarrow \arg \max_{\pi \in A} \text{ITCA}^{CV}(\pi; D, C) \).

2.4 Alternative criteria that may guide class combination

In addition to ITCA, we consider five alternative criteria that may guide class combination. The first two are commonly used criteria: classification accuracy and mutual information. The last three are our newly proposed criteria to balance the trade-off between classification accuracy and class resolution from three other perspectives.

**Accuracy.** It is the most commonly used criterion to evaluate the performance of a classification algorithm. For a class combination \( \pi_K \), given a classification algorithm \( C \) and a size-\( n \) dataset \( D \), the \emph{R-fold CV accuracy (ACC)} is

\[
\text{ACC}^{CV}(\pi_K; D, C) := \frac{1}{R} \sum_{r=1}^R \frac{1}{|D_r^p|} \sum_{(X_i, Y_i) \in D_r^p} \mathbb{I} \left( \phi^C_{\pi_K}(X_i) = \pi_K(Y_i) \right),
\] (6)

12
where the dataset $\mathcal{D}$ is randomly split into $R$ equal-sized folds, with the $r$-th fold $\mathcal{D}_r^e$ serving as the evaluation data and the union of the remaining $R - 1$ folds $\mathcal{D}_r^t$ serving as the training data, and $\phi_{\pi_K}^{, \mathcal{D}_r^t}$ is the classifier trained by the algorithm $\mathcal{C}$ on $\mathcal{D}_r^t$. Typically, ACC$^\text{CV}$ is used without class combination because it is maximized as 1 when all classes are combined into one. Hence, intuitively, it is not an appropriate criterion for guiding class combination. In the following text, we refer to ACC$^\text{CV}$ as the ACC criterion.

**Mutual information.** It measures the dependence between two random variables, which, in the context of classification, can be the observed class label and the predicted class label. In this sense, the mutual information can be used as a criterion of classification accuracy. Following the definition of the mutual information of two jointly discrete random variables \cite{Cover, 1999}, we define the $R$-fold CV mutual information (MI) of $\pi_K$ given $\mathcal{C}$ and $\mathcal{D}$ as

$$\text{MI}^\text{CV}(\pi_K; \mathcal{D}, \mathcal{C}) := \frac{1}{R} \sum_{r=1}^{R} \sum_{k_0=1}^{K} \sum_{k=1}^{K} \frac{\sum_{(X_i,Y_i) \in \mathcal{D}_r^e} \mathbb{I}(Y_i = k_0, \phi_{\pi_K}^{, \mathcal{D}_r^t}(X_i) = k)}{|\mathcal{D}_r^e|} \cdot \log \left( \frac{|\mathcal{D}_r^e| \sum_{(X_i,Y_i) \in \mathcal{D}_r^e} \mathbb{I}(Y_i = k_0, \phi_{\pi_K}^{, \mathcal{D}_r^t}(X_i) = k)}{\sum_{(X_i,Y_i) \in \mathcal{D}_r^e} \mathbb{I}(Y_i = k_0) \sum_{(X_i,Y_i) \in \mathcal{D}_r^e} \mathbb{I}(\phi_{\pi_K}^{, \mathcal{D}_r^t}(X_i) = k)} \right) \right) \right)$$

where in the $r$-th fold, the mutual information is calculated for $\left\{ \left( \phi_{\pi_K}^{, \mathcal{D}_r^t}(X_i), Y_i \right) : (X_i,Y_i) \in \mathcal{D}_r^e \right\}$, i.e., between the predicted labels after class combination $\pi_K$ and the original labels. Note that the mutual information does not require $K_0 = K$ in (7). The reason why we do not use the mutual information of $\left\{ \left( \phi_{\pi_K}^{, \mathcal{D}_r^t}(X_i), \pi_K(Y_i) \right) : (X_i,Y_i) \in \mathcal{D}_r^e \right\}$ (i.e., between the predicted labels and observed labels, both after class combination $\pi_K$) is that it increases as more classes are combined—an undesirable phenomenon. In the following text, we refer to MI$^\text{CV}$ as the MI criterion.

**Adjusted accuracy.** Neither the ACC criterion nor the MI criterion directly uses the class proportions. However, intuitively, it is easier to predict a data point from a larger class. To address this issue, we propose the $R$-fold CV adjusted accuracy (AAC) to weigh each correctly predicted
data point by the inverse proportion of the combined class the data point belongs to:

\[
AAC^{CV}(\pi_K; D, C) := \frac{1}{R} \sum_{r=1}^{R} \sum_{|D_r|} \frac{1}{|D_e|} \left( \frac{\mathbb{I}\left(\phi^{C,D_r}_{\pi_K}(X_i) = \pi_K(Y_i)\right)}{\hat{p}_{k_0}} \right),
\]

where \(\hat{p}_{k_0}\) is the proportion of the \(k_0\)-th original class in \(D\), same as in (2), so the denominator \(\sum_{k_0 \in \pi_K^{-1}(\pi_K(Y_i))} \hat{p}_{k_0}\) is the proportion of \(Y_i\)’s combined class \(\pi_K(Y_i)\) in \(D\). The idea is straightforward: assigning smaller weights to the classes that take up larger proportions. In the following text, we refer to \(AAC^{CV}\) in (8) as the AAC criterion. An alternative proposal of the adjusted accuracy is to assign smaller weights to the classes that are combined from more original classes; however, this proposal does not work as well as the AAC criterion.

**Combined Kullback–Leibler divergence.** To balance the trade-off between the prediction accuracy and resolution, we also propose a combined Kullback–Leibler (CKL) divergence criterion that adds up (1) the divergence of the joint feature distribution estimated using combined class labels on the evaluation data \(D_e\) (denoted by \(\hat{F}_{\pi_K,D_e}: X \rightarrow [0,1]\)) from that estimated using original class labels (denoted by \(\hat{F}_{\pi_K,D_e}: X \rightarrow [0,1]\)) and (2) the divergence of the joint feature distribution estimated using predicted combined class labels (denoted by \(\hat{F}_{\phi^{C,D_t}_{\pi_K},D_e}: X \rightarrow [0,1]\), where the classifier \(\phi^{C,D_t}_{\pi_K}\) is trained on training data \(D_t\) from that estimated using combined class labels, i.e., \(\hat{F}_{\pi_K,D_e}\). Accordingly, the \(R\)-fold CV CKL is defined as

\[
CKL^{CV}(\pi_K; D, C) := \frac{1}{R} \sum_{r=1}^{R} D_{KL}\left(\hat{F}_{\pi_K,D_e} || \hat{F}_{\pi_K,D_e}\right) + D_{KL}\left(\hat{F}_{\phi^{C,D_t}_{\pi_K},D_e} || \hat{F}_{\pi_K,D_e}\right).
\]

In the following text, we refer to \(CKL^{CV}\) as the CKL criterion. A challenge in calculating CKL is the estimation of \(d\)-dimensional joint feature distributions. To circumvent this challenge, we only calculate CKL when all class-conditional feature distributions are approximately Gaussian; that is, \(\hat{F}_{\pi_K,D_e}, \hat{F}_{\pi_K,D_e},\) and \(\hat{F}_{\phi^{C,D_t}_{\pi_K},D_e}\) can all be approximated by Gaussian mixture models. While this is an overly restrictive assumption, we use CKL as an alternative criterion to benchmark ITCA in simulation studies where this assumption holds (Section 3).

**Prediction entropy.** The p-ITCA definition in (5) does not equate to but reminds us of the entropy of the distribution of \(\left(\phi^{C,D_t}_{\pi_K}(X), \pi_K(Y)\right)\) conditional on \(\phi^{C,D_t}_{\pi_K}(X) = \pi_K(Y)\), which we
refer to as the population-level prediction entropy (p-PE):

$$\text{p-PE}(\pi_K; D_t, C) = \sum_{k=1}^{K} -\mathbb{P}(\phi_{\pi_K}^{C,D_t}(X) = \pi_K(Y) = k) \cdot \log \mathbb{P}(\phi_{\pi_K}^{C,D_t}(X) = \pi_K(Y) = k)$$

(10)

In the following text, we refer to PE\textsubscript{CV} as the PE criterion. We argue that the definition of PE is not as intuitive as that of ITCA because PE only considers the data points that have labels correctly predicted; hence, compared to ITCA, PE does not fully capture the class resolution information.

To compute ITCA, ACC, MI, AAC, CKL, and PE, we set the number of folds in CV to $R = 5$ in all numerical analyses.

3. SIMULATION STUDIES

3.1 ITCA outperforms alternative class combination criteria on simulated data

To verify the effectiveness of ITCA, we first compare ITCA with the five alternative criteria (ACC, MI, AAC, CKL, and PE) in simulations. Specifically, given the number of observed classes $K_0$ and the true class combination $\pi^*_K: [K_0] \rightarrow [K^*]$, we first generate $K^*$ class centers by a random walk in $\mathbb{R}^d$. The random walk starts from the center of class 1: $\mu_1 = 0 \in \mathbb{R}^d$. At time $k = 2, \ldots, K^*$, the walk chooses a direction at random and takes a step with a fixed length $l$; that is, the center
of class \( k \) is \( \mu_k = \mu_{k-1} + lv \), where \( v \in \mathbb{R}^d \) is a random direction vector with the unit length, i.e., \( \|v\| = 1 \). To make the true classes distinguishable, we ensure that the minimal pairwise Euclidean distance among \( \mu_1, \ldots, \mu_{K^*} \) is greater than \( \sigma \), i.e., the standard deviation of every feature in each class. Given \( \{\mu_k\}_{k=1}^{K^*} \) and \( \sigma \), we then define the distribution of \((X, Y)\) as follows.

1. The true class \( Y^* \sim \text{discrete uniform}([K^*]) \), i.e., \( Y^* \) randomly picks a value in \([K^*]\) with probability \( 1/K^* \).

2. The observed class \( Y \sim \text{discrete uniform}(\pi_{K^*}^{-1}(Y^*)) \), i.e., \( Y \) randomly picks a value in \( \pi_{K^*}^{-1}(Y^*) = \{k_0 \in [K_0] : \pi_{K^*}(k_0) = Y^*\} \subset [K_0] \) —the observed classes that belong to the true class \( Y^* \)—with probability \( 1/|\pi_{K^*}^{-1}(Y^*)| = 1/(\sum_{k_0}^{K_0} \mathbb{1}(\pi_{K^*}(k_0) = Y^*)) \).

3. The observed feature vector \( X \sim \mathcal{N}(\mu_{Y^*}, \sigma^2 I_d) \), i.e., \( X \) follows a \( d \)-dimensional Gaussian distribution with mean \( \mu_{Y^*} \) specified by the true class \( Y^* \).

Given a dataset \( \{(X_i, Y_i)\}_{i=1}^n \), which contains independently and identically distributed (i.i.d.) observations from the above distribution, an ideal criterion of class combination is expected to be maximized at \( \pi_{K^*}^+ \).
Figure 2: Comparison of ITCA and five alternative criteria (ACC, MI, AAC, CKL, and PE) using the LDA as the classification algorithm. The dataset is generated with $K_0 = 6$, $K^* = 3$, $l = 3$, $\sigma = 1.5$, $n = 2000$, and $d = 5$. The true class combination is $\pi^*_3 = \{(1, 2), (3, 4), (5, 6)\}$. For each criterion (panel), the 31 blue points correspond to the 31 class combinations $\pi_K$’s with $K = 2, \ldots, 6$. The true class combination $\pi^*_K$, is marked with the red star, and the best value for each criterion is indicated by a horizontal dashed line. The true class combination is only found by PE and ITCA without close ties.

We generate a simulated dataset with $K_0 = 6$, $K^* = 3$, $l = 3$, $\sigma = 1.5$, $n = 2000$, and $d = 5$. We assume the true class combination is $\pi^*_3 = \{(1, 2), (3, 4), (5, 6)\}$, i.e., the 1st and 2nd observed classes belong to one true class, and so do the 3rd and 4th observed classes, as well as the 5th and 6th observed classes. We also assume that the classes are ordinal; that is, $\mathcal{A}$ contains $2^5 - 1 = 31$ class combinations (Table 1). Thanks to the moderate size of $|\mathcal{A}|$, we use the exhaustive search to enumerate all allowed class combinations.

Regarding the classification algorithm, we consider the LDA and the random forest (RF) [Breiman, 2001]. Figure 2 shows that ITCA using LDA successfully finds the true class combination $\pi^*_3$; that is, when evaluated at the 31 $\pi_K$’s, ITCA is maximized at $\pi^*_3$. As expected, ACC is maximized when the $K_0 = 6$ observed classes are combined into $K = 2$ classes; hence, it is not ap-
propriate for guiding class combination. Although better than ACC, MI and two newly proposed
criteria (AAC and CKL) still fail to find $\pi_3^*$. Among the alternative criteria, only PE correctly
identifies $\pi_3^*$ because its definition is similar to that of ITCA.

To further evaluate the performance of ITCA and the five alternative criteria, we repeat the
above simulation for 31 times, each time using one of the 31 allowed class combinations as the true
class combination $\pi_{K^*}^*$; the other simulation parameters are kept the same. In total, we generate
31 datasets, one for each true class combination. We still use the exhaustive search with LDA
or RF to find the best class combination guided by each criterion, denoted by $\pi_{K^*}^m$ for criterion
$m \in \{\text{ITCA, ACC, MI, AAC, CKL, PE}\}$. To evaluate the performance of criterion $m$, we define
the distance between $\pi_{K^*}^m$ and $\pi_{K^*}^*$ as follows. First, we encode each allowed class combination
$\pi_K : [K_0] \rightarrow [K]$ for combining $K_0$ ordinal classes as a $(K_0 - 1)$-dimensional binary vector, as
in the “stars and bars” used in combinatorics [Feller, 2008]. For example, the class combination
$\{(1, 2), 3, 4, 5, 6\}$ can be represented by 12|3|4|5|6 and thus encoded as the binary vector (0, 1, 1, 1, 1),
where the 0 indicates that there is no bar between the original classes 1 and 2, the first 1 indicates
that there is a bar between the original classes 2 and 3, etc. Second, we define the distance between
$\pi_{K^*}^m$ and $\pi_{K^*}^*$ as the Hamming distance between their binary encodings; hence, the distance takes
an integer value ranging from 0 to $K_0 - 1$, with 0 indicating that $\pi_{K^*}^m = \pi_{K^*}^*$, i.e., the criterion $m$
finds the true class combination.

We evaluate the performance of each criterion using three criteria: (1) the number of datasets
(the larger the better) on which the criterion identifies the true class combination; (2) the average
and (3) the maximum Hamming distances (the smaller the better) between the criterion’s best class
combination and the true class combination across the 31 datasets. Among the six criteria, ITCA
has the best performance under all three criteria (Table 2). PE has the second best performance
after ITCA; the other criteria fail to find the true class combination on at least 70% of the datasets.

Moreover, we escalate this simulation design by setting $K_0 = 8$; then the number of allowed
class combinations becomes $2^7 - 1 = 127$ (Table 1). Correspondingly, we generate 127 datasets,
each having one allowed class combination as the true class combination. The results of the six
criteria are summarized in Table 3. Again, ITCA has the best performance.
Table 2: The performance of six criteria on the 31 simulated datasets with $K_0 = 6$. The best result in each column is boldfaced.

| Criterion | # successes | Average | Max | # successes | Average | Max |
|-----------|-------------|---------|-----|-------------|---------|-----|
|           | # datasets  |         |     |             |         |     |
|            | Hamming     |         |     |             |         |     |
|            |             |         |     |             |         |     |
| LDA        |             |         |     |             |         |     |
| RF         |             |         |     |             |         |     |
| ACC        | 1/31        | 2.03    | 4   | 1/31        | 2.03    | 4   |
| MI         | 8/31        | 1.42    | 4   | 6/31        | 1.65    | 4   |
| AAC        | 9/31        | 1.03    | 3   | 8/31        | 1.30    | 3   |
| CKL        | 7/31        | 2.42    | 5   | 1/31        | 2.13    | 4   |
| PE         | 22/31       | 0.55    | 3   | 22/31       | 0.42    | 3   |
| ITCA       | **26/31**   | **0.23**| **2**| **26/31**   | **0.19**| **2**|

Table 3: The performance of six criteria on the 127 simulated datasets with $K_0 = 8$. The best result in each column is boldfaced.

| Criterion | # successes | Average | Max | # successes | Average | Max |
|-----------|-------------|---------|-----|-------------|---------|-----|
|           | # datasets  |         |     |             |         |     |
|            | Hamming     |         |     |             |         |     |
|            |             |         |     |             |         |     |
| LDA        |             |         |     |             |         |     |
| RF         |             |         |     |             |         |     |
| ACC        | 6/127       | 2.54    | 6   | 7/127       | 2.53    | 6   |
| MI         | 7/127       | 2.51    | 6   | 11/127      | 2.33    | 6   |
| AAC        | 15/127      | 2.02    | 6   | 15/127      | 1.98    | 6   |
| CKL        | 3/127       | 3.68    | 6   | 5/127       | 2.87    | 5   |
| PE         | 101/127     | 0.47    | 4   | 94/127      | 0.46    | 3   |
| ITCA       | **120/127** | **0.12**| **3**| **120/127** | **0.08**| **2**|

Table 4: Performance of ITCA using five search strategies and LDA on the 31 simulated datasets with $K_0 = 6$.

| Strategy    | # successes | Average | Max | Average # class combinations examined |
|-------------|-------------|---------|-----|---------------------------------------|
|             | # datasets  |         |     |                                       |
|             | Hamming     |         |     |                                       |
| Exhaustive  | 26/31       | 0.23    | 2   | 31.00                                 |
| Greedy search | 26/31     | 0.23    | 2   | 12.13                                 |
| BFS         | 26/31       | 0.19    | 2   | 19.19                                 |
| Greedy (pruned) | 26/31   | 0.19    | 2   | 5.71                                  |
| BFS (pruned)| 26/31       | 0.19    | 2   | 8.84                                  |
Table 5: Performance of ITCA using five search strategies and LDA on the 127 simulated datasets with $K_0 = 8$.

| Strategy         | # successes | Average # datasets | Average Hamming | Max Hamming | Average # class combinations examined |
|------------------|-------------|--------------------|-----------------|-------------|---------------------------------------|
| Exhaustive       | 120/127     | 0.13               | 3               | 127.00      |
| Greedy search    | 120/127     | 0.12               | 3               | 22.52       |
| BFS              | 120/127     | 0.10               | 2               | 53.61       |
| Greedy (pruned)  | 120/127     | 0.09               | 2               | 11.91       |
| BFS (pruned)     | 120/127     | 0.09               | 3               | 27.20       |

The above results verify the effectiveness of ITCA in finding the true class combination. In the following, we compare the two proposed search strategies, the greedy search and BFS, with the exhaustive search. Specifically, we use the aforementioned 31 and 127 simulated datasets (corresponding to $K_0 = 6$ and 8, respectively), and we apply ITCA under the three search strategies, with LDA as the classification algorithm. Tables 4 and 5 (the top three rows in each table) show that the greedy search and BFS are almost as effective as the exhaustive search in finding the true class combination. Compared with the exhaustive search, the greedy search and BFS examine fewer class combinations and thus greatly reduce the computational time because each class combination, if examined, needs a separate classifier training.

We note that the search space of the greedy search and BFS can be further pruned if the classification algorithm satisfies a non-stringent property. As a preview, the bottom two rows of Tables 4 and 5 show that pruning reduces the search spaces of the greedy search and BFS while maintaining the performance.

When $K_0$ is large, it is unrealistic to use the exhaustive search. Here we use $K_0 = 20$ ordinal classes as an example. Out of the $2^{19} - 1 \approx 5.24 \times 10^5$ allowed class combinations (Table 1), we randomly select 50 class combinations as the true class combination $\pi_{K^*}$, whose $K^*$ ranges from 7 to 16. From each $\pi_{K^*}$, we generate a dataset with $n = 10,000$ data points (the other parameters are the same as in the aforementioned simulations). The results show that the greedy search works as well as the BFS (Table 6): both successfully find the $\pi_{K^*}$ of each dataset. On average, the greedy search only needs to evaluate ITCA on 150.08 class combinations ($87.70$ combinations with the pruned search space) out of the $\sim 5.24 \times 10^5$ allowed class combinations. In contrast, the BFS
has a much larger search space ($\sim 10^4$ class combinations).

Table 6: Performance of ITCA using five search strategies and LDA on the 50 simulated datasets with $K_0 = 20$.

| Strategy            | # successes | Average # datasets | Average Hamming | Max Hamming | Average # class combinations examined |
|---------------------|-------------|--------------------|------------------|-------------|---------------------------------------|
| Greedy              | 50/50       | 0.00               | 0                |             | 150.08                                |
| BFS                 | 50/50       | 0.00               | 0                |             | 27226.84                              |
| Greedy (pruned)     | 50/50       | 0.00               | 0                |             | 87.70                                 |
| BFS (pruned)        | 50/50       | 0.00               | 0                |             | 17155.82                              |

Notably, ITCA has a higher probability of success when $K^*$ is larger.

3.2 ITCA outperforms alternative class combination criteria on the Iris data

We also compare ITCA with the five alternative class combination criteria on the famous Iris dataset\(^1\), which contains $K^* = 3$ classes (corresponding to three types of irises: setosa, versicolor, and virginica) with 50 data points in each class. The setosa class is linearly separable from the versicolor and virginica classes, while versicolor and virginica are not linearly separable from each other. To prepare the dataset for class combination, we randomly split the setosa class into two equal-sized classes, making the number of observed classes $K_0 = 4$. Since the four classes are nominal, there are 14 allowed class combinations (Table 1).

\(^1\)http://archive.ics.uci.edu/ml/datasets/Iris/
Figure 3: Comparison of ITCA and five alternative criteria (ACC, MI, AAC, CKL, and PE) using LDA as the classification algorithm on the Iris data. The number of observed classes is $K_0 = 4$. The number of true classes is $K^* = 3$ (with the true class combination $\pi_{K^*}$ marked by the arrow and the red vertical dashed line in every panel). For every allowed class combination $\pi_K$, each criterion has its value (calculated by 5-fold CV) marked by a red circle for $K = 4$, a blue square for $K = 3$, and a green diamond for $K = 2$); each error bar has half its length corresponding to the standard error of the criterion value (i.e., the standard deviation of the 5 criterion values in the 5-fold CV, divided by $\sqrt{5}$). The horizontal line marks each criterion’s best value. Among the six criteria, only AAC, CKL, PE, and ITCA are maximized at $\pi_{K^*}$, and only ITCA has a clear gap between $\pi_{K^*}$ and all other class combinations.

For each allowed class combination, we compute the six class combination criteria with LDA as the classification algorithm (Figure 3). Among the six criteria, AAC, CKL, PE, and ITCA successfully find the true class combination $\pi_{K^*}$. However, only ITCA leads to a clear gap between $\pi_{K^*}$ and the other 13 allowed class combinations. Note that CKL has large error bars because its computation involves the inverses and determinants of sample covariance matrices, whose accurate
estimation requires a large sample size. Particularly, ACC has an undesirable result: its maximal value 1 is obtained at the class combination $\pi_2$ where the *versicolor* and *virginica* classes are combined. These results confirm the unsuitability of ACC for guiding class combination, and they demonstrate the advantage ITCA has over the five alternative criteria.

3.3 ITCA outperforms clustering-based class combination

While ITCA provides a powerful data-driven approach for combining ambiguous classes, one may intuitively consider using a clustering algorithm to achieve the same goal. We consider three clustering-based class combination approaches, which are summarized below, and we compare them with ITCA on simulated data under four settings.

**K-means-based class combination.** For the $k_0$-th class ($k_0 = 1, \ldots, K_0$), we first compute the $k_0$-th class center as $(\sum_{i=1}^{n} 1(Y_i = k_0) X_i) / (\sum_{i=1}^{n} 1(Y_i = k_0))$. We then use the K-means clustering to cluster the $K_0$ class centers into $K^*$ clusters so that the $K_0$ observed classes are correspondingly combined into $K^*$ classes.

**Spectral-clustering-based class combination.** We first compute the $K^*$-dimensional spectral embeddings of $X_1, \ldots, X_n$ [Ng et al., 2001]. Then we apply the above K-means-based class combination approach to the $n$ spectral embeddings to combine the $K_0$ observed classes into $K^*$ combined classes.

**Hierarchical-clustering-based class combination.** We first compute the $K_0$ class centers as in the K-means-based class combination approach. Then we use the hierarchical clustering with the single, complete, or average linkage to cluster the $K_0$ class centers into $K^*$ clusters so that the $K_0$ observed classes are correspondingly combined into $K^*$ classes.

Note that all these clustering-based class combination approaches require that $K^*$ (the true number of classes) is known or estimated by an external approach (e.g., an approach for determining the number of clusters [Tibshirani et al., 2001, Sugar and James, 2003, Pham et al., 2005]), which is by itself a difficult problem in real-world applications. In contrast, ITCA does not require $K^*$ to be known beforehand; instead, its optimal class combination determines $K^*$ in a data-driven way.
Figure 4: Comparison of clustering-based class combination approaches and ITCA (using Gaussian kernel SVM as the classification algorithm). Each row corresponds to one simulated dataset. From top to bottom, the number of observed classes is $K_0 = 4$, 5, 5, and 4, and the number of true classes is $K^* = 3$, 3, 3, and 3. In the leftmost column, colors mark the observed classes; in the other columns, the three colors indicate the three combined classes found by each combination approach. Check marks indicate the cases where the true class combinations are found. Only ITCA finds the true class combination on every dataset.

To benchmark ITCA against the clustering-based class combination approaches, we generate four datasets with two-dimensional features ($X_1, \ldots, X_n \in \mathbb{R}^2$), which are shown in the four rows of Figure 4 (see Supplementary Material for the details of data generation). Then we apply the above five clustering-based class combination approaches (hierarchical clustering has three linkages) and ITCA to the synthetic data; for ITCA, we use the Gaussian kernel support vector machine (SVM) as the classification algorithm.

Figure 4 shows the results of ITCA and the clustering-based class combination approaches with $K^* = 3$ known: only ITCA successfully finds the true class combination on every dataset. We
conclude that ITCA is advantageous over the clustering-based approaches even with a known $K^*$. The major reason is that the clustering-based approaches only use the $K_0$ class centers (whose definition depends on a distance metric) and do not fully use the information in individual data points, which play a central role in the definition of ITCA.

4. APPLICATIONS

4.1 Prognosis of rehabilitation outcomes of traumatic brain injury patients

According to the Centers for Disease Control and Prevention, traumatic brain injury (TBI) affects an estimated 1.5 million Americans every year. For example, in the year 2010, the nationwide financial cost of TBI, including both direct and indirect medical costs, is estimated to be approximately $76.5 billion$\(^2\). Millions of TBI survivors have permanent TBI-related disability and thus sustain enormous financial burden and personal costs (e.g., motor impairment, cognitive impairment, and relation disruptions) [Ricker, 1998]. The inpatient TBI rehabilitation care alone costs each patient tens of thousands of dollars per month. However, TBI rehabilitation is proven to be effective for only some but not all patients [Turner-Stokes, 2008]. Therefore, there is a great demand to have an automatic prognosis algorithm that can accurately predict rehabilitation outcomes for individual patients and assist patients’ decisions in seeking rehabilitation care.

We have access to the Casa Colina dataset of $n = 3078$ TBI patients who received inpatient rehabilitation care. Patients’ disability severity was evaluated and recorded by physical therapists in the form of the Functional Independence Measure (FIM) of 17 activities at admission and discharge. The FIM has seven scales ranging from 1 (patient requires total assistance to perform an activity) to 7 (patient can perform the activity with complete independence) [Linacre et al., 1994]. In addition, patients’ characteristics are recorded, including demographics (gender and age) and admission status. This dataset allows the development of an algorithm to predict the efficacy of rehabilitation care for individual patients.

We formulate the task of predicting patients’ rehabilitation outcomes as a multi-class classification problem, where the discharge FIM of each activity is a seven-level ($K_0 = 7$) ordinal outcome and the features are patients’ characteristics and admission FIM. We are motivated to combine

\[^2\]https://www.cdc.gov/traumaticbraininjury/get_the_facts.html
outcome levels because the RF algorithm, though being the best-performing algorithm, has low accuracy for predicting the $K_0$ levels of many activities. Hence, we consult the physical therapists who graded the activities and obtain their suggested combination \{1, (2, 3, 4), 5, 6, 7\}, which has levels 2–4 combined. However, this expert-suggested class combination is subjective and not activity-specific. Intuitively, we reason that different activities may have different resolutions and thus different outcome level combinations. Hence, we apply ITCA as a data-driven approach to guide the combination of outcome levels for each activity.

Powered with ITCA, we can construct a multilayer prediction framework (Figure 5), whose $K_0$ layers (from top to bottom) correspond to the numbers of combined classes $K = 1, \ldots, K_0$, with the bottommost layer indicating no class combination. There are two ways to construct a framework: greedy-search-based and exhaustive-search-based. In a greedy-search-based framework, layers are constructed by the greedy search (Section 2.3) in a sequential way from bottom up: classes in each layer except the bottommost layer are combined by ITCA from the classes in the layer right below, so the layers would follow a nested structure. In contrast, in an exhaustive-search-based framework, layers are constructed separately from the bottommost layer: each layer has its optimal class combination defined by ITCA and found by the exhaustive search (Section 2.3) given its $K$; thus, there is no nested constraint. Note that $K_0$ is often not too large in medical diagnosis and prognosis, making the exhaustive search computationally feasible.

Greedy-search-based and exhaustive-search-based frameworks have complementary advantages. On the one hand, the former outputs a data-driven hierarchy of class combinations and is thus more interpretable if the prediction would be conducted for all layers. On the other hand, the latter outputs the optimal class combination for each layer and allows choosing the optimal layer (that maximizes the ITCA); hence, it is more desirable if the prediction would only be conducted for the optimal layer. For example, if healthcare providers would like to predict outcomes at a multilayer resolution, the greedy-search-based framework is more suitable. In contrast, if the priority is to combine outcome levels into the optimal resolution for prediction, the exhaustive-search-based framework would be a better fit.

We use the RF as the classification algorithm for its better accuracy than other popular algorithms’ on the Casa Colina dataset. Leveraging the RF algorithm, we apply the greedy-search-based
and exhaustive-search-based multilayer frameworks to predict the rehabilitation outcome of each of the 17 activities. For example, Figure 5 shows the results for predicting the Toileting activity: ITCA indicates the same optimal class combination in both frameworks: \(\{1, 2, (3, 4), 5, (6, 7)\}\) in layer 5 \((K = 5)\), where levels 3 and 4 are combined, and so are levels 6 and 7.

![Greedy-search-based and exhaustive-search-based multilayer frameworks for predicting the Toileting activity in the Casa Colina dataset. In either framework, layer \(K\) has a class combination \(\pi_K\) chosen by ITCA. In the greedy-search-based framework, the layers have a tree structure. Layer 5 has the same class combination in both frameworks and is found optimal by ITCA.](image)

Figure 5: Greedy-search-based and exhaustive-search-based multilayer frameworks for predicting the Toileting activity in the Casa Colina dataset. In either framework, layer \(K\) has a class combination \(\pi_K\) chosen by ITCA. In the greedy-search-based framework, the layers have a tree structure. Layer 5 has the same class combination in both frameworks and is found optimal by ITCA.

To evaluate the ITCA results for the 17 activities, we calculate the prediction accuracy by the RF for our ITCA-guided combinations, the expert-suggested combination, and the hierarchical-clustering-based combinations (defined in Section 3.3; the results based on the average linkage are shown). Note that the prediction accuracy for different class combinations may differ even if the best guess algorithm (i.e., the naïve algorithm that assigns data points to the largest class—a baseline control) is used. Hence, for a fair comparison, we evaluate each class combination by the 5-fold CV prediction accuracy improvement of the RF algorithm from the best guess algorithm.

Figure 6 shows the comparison results for eight activities, for which ITCA selects class combinations with \(K\)’s closest to 5 (i.e., the number of combined levels suggested by experts). We find that, compared with the expert-suggested combination and the hierarchical-clustering-based combinations, the ITCA-guided combinations consistently lead to more balanced classes (which are more difficult to predict, as indicated by the lower prediction accuracy of the best guess algorithm) and more significant improvement in prediction accuracy.
Figure 6: Prediction accuracy for the optimal level combination indicated by ITCA, the expert-suggested combination, and the hierarchical-clustering-based level combinations for the rehabilitation outcomes of eight activities in the Casa Colina dataset. The horizontal axis shows the 5-fold CV prediction accuracy of the best guess algorithm, and the vertical axis shows the accuracy of the RF algorithm. The accuracy improvement of the RF from the best guess is marked on vertical dashed lines. Note that in the “Comprehension” panel, ITCA finds the expert-suggested combination; the two accuracy improvement values (0.298 and 0.291) should be equal in theory, but they are different due to the randomness of data splitting in the 5-fold CV.

As a side note, we argue that it is inappropriate to use the prediction accuracy improvement (from the best guess) as a class combination criterion. The reason is that the prediction accuracy of the best guess, though serving as a baseline accuracy, does not necessarily reflect the class resolution. For example, an outcome encoded as two classes with equal probabilities has a lower resolution than another outcome encoded as three classes with probabilities 0.5, 0.25, and 0.25; however, the best guess algorithm has the same prediction accuracy 0.5 for the two outcomes.
4.2 Prediction of glioblastoma cancer patients’ survival time

Glioblastoma cancer, also known as glioblastoma multiforme (GBM), is the most aggressive type of cancer that begins within the brain. Hence, it is of critical importance to predict GBM patients’ survival time so that appropriate treatments can be provided. In this prediction task, patients’ survival time would be predicted from their clinical measurements.

We download the TCGA GBM dataset from the cBio cancer genomics portal [Cerami et al., 2012]. The dataset contains $n = 541$ patients’ demographics, gene expression subtypes, therapy, and other clinical information. We remove nine features irrelevant to survival prediction and keep $d = 36$ features (see Supplementary Material for the data processing details).

We formulate this survival prediction task as a classification problem instead of a regression problem to demonstrate the use of ITCA, and we compare the performance of the classifier guided by ITCA with the Cox regression model. Concretely, we first discretize patients’ survival time into $K_0 = 12$ intervals: $[0, 3)$, $[3, 6)$, $[6, 9)$, $[9, 12)$, $[12, 15)$, $[15, 18)$, $[18, 21)$, $[21, 24)$, $[24, 27)$, $[27, 30)$, $[30, 33)$, $[33, +\infty)$, where $[0, 3)$ and $[33, +\infty)$ indicate that the survival time is less than 3 months and at least 33 months, respectively.

We then use ITCA to optimize the survival time intervals for prediction. For the classification algorithm, we use a three-layer neural network (NN) with the ReLU activation function and a modified cross entropy as the loss function for handling censored survival time (see Supplementary Material). Starting from the observed $K_0 = 12$ classes, we use ITCA with greedy search to select a class combination for each $K = 11, \ldots, 2$. For each $K$ and its selected class combination, we train an NN classifier and show the ITCA and ACC in Figure 7. As expected, as $K$ decreases, ACC increases, confirming our motivation that ACC cannot be used to guide class combination. In contrast, as $K$ decreases, ITCA first increases until $K = 7$ and then decreases, confirming that ITCA balances the trade-off between classification accuracy and class resolution.

Following the tradition in survival analysis, we use the Kendall’s tau coefficient (calculated between the predicted outcome and the observed survival time, not the discretized survival time interval, in 5-fold CV) to evaluate the prediction accuracy. Note that the Kendall’s tau is a reasonable accuracy measure for survival prediction because it is an ordinal association measure.

---

3 The dataset is available at https://www.cbioportal.org/
Figure 7: Results of GBM survival prediction. Three criteria, ACC, ITCA and Kendall’s tau coefficient, are shown for $K$ from $K_0 = 12$ to 2 (for each $K$, a class combination is found by ITCA, and an NN classifier is trained). Each criterion is calculated by the 5-fold CV, and its mean and standard error (i.e., standard deviation of the 5 criterion values in the 5-fold CV, divided by $\sqrt{5}$) are shown. The best class combination $\pi^*_K$ found by ITCA is indicated by the red vertical dashed line ($K = 7$), where the Kendall’s tau is also maximized.

that allows the predicted outcome to be either discrete (as in our classification formulation) or continuous (as in a regression formulation). Figure 7 shows that the optimal class combination found by ITCA leads to the best Kendall’s tau, verifying that ITCA optimizes the survival time intervals for classifier construction.

We also compare the NN algorithm with two commonly used survival prediction algorithms: the Cox regression [Cox, 1972], a regression algorithm that predicts patients’ risk scores, and the logistic regression (LR), a multi-class classification algorithm that uses the same modified cross entropy loss to predict survival time intervals as the NN algorithm does. We use the Kendall’s tau to evaluate five prediction models: NN and LR classifiers for predicting the original $K_0$ survival time intervals, NN and LR classifiers for predicting their respective combined intervals guided by ITCA, and a Cox regression model for predicting risk scores (the Kendall’s tau is calculated between negative predicted risk scores and observed survival time). Table 7 shows that the NN classifier trained for ITCA-guided combined intervals has the best prediction accuracy in terms of the Kendall’s tau; moreover, it has the highest ITCA value among the four classifiers. This result again verifies that ITCA is a meaningful accuracy measure.
Note that the Kendall’s tau is not an appropriate measure to replace ITCA because it requires the response to be a numerical or ordinal variable. Hence, the Kendall’s tau cannot guide the combination of nominal class labels.

Table 7: Performance of survival prediction algorithms on the GBM dataset.

| Model                                    | ITCA         | Kendall’s tau | Average p-value |
|------------------------------------------|--------------|---------------|-----------------|
| NN (\(K_0\) survival time intervals)    | 0.8565 ± 0.0410 | 0.6547 ± 0.0181 | 2.11e-14        |
| LR (\(K_0\) survival time intervals)    | 0.6354 ± 0.0620 | 0.6024 ± 0.0244 | 1.64e-11        |
| NN (ITCA-guided combined intervals)      | 0.9623 ± 0.0464 | 0.6855 ± 0.0178 | 1.27e-15        |
| LR (ITCA-guided combined intervals)      | 0.8196 ± 0.0222 | 0.6236 ± 0.0240 | 5.34e-10        |
| Cox regression (risk scores)             | -            | 0.6303 ± 0.0542 | 2.04e-13        |

Each criterion is computed by 5-fold cross validation; its mean and standard error (i.e., standard deviation of the 5 criterion values in the 5-fold CV, divided by \(\sqrt{5}\)) are listed; the average of the 5 p-values corresponding to the Kendall’s tau coefficients in the 5-fold CV is also listed. The NN algorithm trained with ITCA-guided \(K = 7\) combined intervals achieves the best ITCA value, the best Kendall’s tau, and the most significant average p-value.

4.3 Prediction of user demographics using mobile phone behavioral data

One of the essential tasks in personalized advertising is to predict users’ demographics (gender and age) using behavioral data. A good predictive model is necessary for data-driven marketing decisions. To simplify the prediction of user ages, data scientists often first discretize ages into groups and then construct a multi-class classifier to predict age groups instead of exact ages [Kaggle, 2016]. However, the discretization step is heuristic and unjustified. Here we use ITCA to determine age groups in a principled, data-driven way.

We apply ITCA to the TalkingData mobile user demographics\(^4\), a public dataset of mobile phone users’ behavioral data in China. Specifically, the dataset contains users’ app usage, mobile device properties, genders, and ages. Our goal is to predict a user’s gender and age from mobile device and app usage.

In detail, we discretize male users’ ages into 17 ordinal groups encoded as M20−, M20–21, . . . , M48–49, and M50+ (where M20−, M20–21, and M50+ indicate male users whose ages are < 20, \(\geq 20 \& < 21\), and \(\geq 50\), respectively). Similarly, we divide female users into 17 ordinal age groups: F20−, F20–21, . . . , F48–49, and F50+. Together, we have \(K_0 = 34\) classes to start with. We

\(^4\)The data is available at https://www.kaggle.com/c/talkingdata-mobile-user-demographics
use one-hot encoding to convert users’ mobile devices and app usage data into 818 features; after deleting the users with zero values in all features, we retain \( n = 23,556 \) users. Then we use XGBoost [Chen and Guestrin, 2016] as the classification algorithm for its successes on similar prediction tasks in Kaggle competitions. We refer readers to the Supplementary Material for a detailed description of our data processing and classifier training procedures.

Figure 8: Results on TalkingData mobile user demographics dataset using XGBoost. Left panel: ITCA and ACC versus the number of combined classes \( K \), which ranges from \( K_0 = 34 \) to 2. The criteria are estimated by 5-fold CV, and their standard errors are calculated by the standard deviations in the 5-fold CV divided by \( \sqrt{5} \). The best class combination \( \pi^*_K \) (with \( K = 6 \)) is indicated by the vertical dashed line. Right upper panel: the histogram of the ages of male users. \( \pi^*_K \) suggests three age groups: M26−, M26–33, and M34+. Right lower panel: the histogram of the ages of female users. \( \pi^*_K \) suggests three age groups: F24−, F24–49, and F50+.

Since 30 out of the 34 classes are exact ages, it is intuitively too challenging to accurately predict the 34 classes simply from users’ phone devices and app usage. This is indeed the case, reflected by the low accuracy (< 0.35) of XGBoost (Figure 8). Hence, we use ITCA with the greedy search strategy to combine the 34 ordinal classes into coarser, more meaningful age groups from the prediction perspective. Note that we add the constraint for not combining a male class with a female class.

Interestingly, ITCA suggests three different age groups for the male and female users (male: M26−, M26–33, and M34+; female: F24−, F24–49, and F50+) (Figure 8, right panel). This result
reveals a gender difference in the mobile phone behavioral data: female users have a wider middle age group (24–49 vs. male users’ 26–33). A possible explanation of this gender difference is the well-known gender disparity in career development in China: more males undergo promotion into senior positions in middle 30s compared with females, who tend to slow down career development at young ages for reasons such as marriage and child birth [Wei, 2011]. This explanation is reasonable in that users’ career development and mobile phone use behaviors are likely correlated. Hence, if we interpret the male and female age groups from the perspective of career development, we find a possible explanation of why the age of 34 is a change point for males but not for females, whose middle-to-senior change point is the age of 49, close to the retirement age of most females in China. In summary, ITCA provides a data-driven approach to defining user age groups based on mobile phone behaviors, making it a potentially useful tool for social science research.

4.4 Detection of biologically similar cell types inferred from single-cell RNA-seq data

Recent advances in single-cell sequencing technologies provide unprecedented opportunities for scientists to decipher the mysteries of cell biology [Wang and Navin, 2015, Papalexi and Satija, 2018, Stuart and Satija, 2019]. An important topic is to discern cell types from single-cell RNA-seq data, which profile transcriptome-wide gene expression levels in individual cells.

Concretely, a single-cell RNA-seq dataset is processed into a data matrix of \( n \) cells and \( d_0 \) genes, with the \((i, j)\)-th entry as the expression level (i.e., log-transformed count of reads or unique molecular identifiers) of the \( j \)-th gene in the \( i \)-th cell. Starting from the matrix, a standard analysis pipeline involves the following steps [Stuart and Satija, 2019]. First, principal component analysis is performed on the data matrix to reduce the column dimension from \( d_0 \) to \( d \ll d_0 \), resulting in a principal component matrix of \( n \) cells and \( d \) principal components. Second, a clustering algorithm (e.g., the graph-based Louvain algorithm [Blondel et al., 2008]) is applied to the principal component matrix to cluster the \( n \) cells. Finally, experts use knowledge to manually annotate the cell clusters with cell type labels.

However, the annotated cell types might be ambiguous due to the subjectivity of setting parameter values in the above pipeline (e.g., the number of principal components \( d \) and the clustering algorithm’s parameters) and the uncertainty of the clustering step. As a result, if cells are over-clustered, some annotated cell types might be biologically similar.
This problem, the detection of biologically similar cell types, can be formulated as an application of ITCA. Here, we use a single-cell RNA-seq dataset of hydra [Siebert et al., 2019] as an example. The processed dataset contains \( n = 25,052 \) cells annotated into \( K_0 = 38 \) cells types (nominal class labels). Following the procedure in [Siebert et al., 2019], we use the first \( d = 40 \) principal components as features. We choose the LDA as the classification algorithm for two reasons. First, our goal is to discover ambiguous cell types instead of achieving high prediction accuracy, so it is reasonable to choose a weak classification algorithm. Second, the features are the principal components obtained from the log-transformed counts, and they are found to approximately follow a multivariate Gaussian distribution. Given the large \( K_0 \), we apply ITCA using the greedy search algorithm. The result suggests the combination of cell type 19 (“enEp tentacle”: endodermal epithelial cells in tentacles) and cell type 30 (“enEp tentnem(pd”: endodermal epithelial cells in tentacle nematocytes—suspected phagocytosis doublets), which indeed have similar cell type labels.

Table 8: The neighboring cell types of cell types 19 and 30 in t-SNE under six perplexity values.

| Perplexity | Neighbors of cell type 19 | Neighbors of cell type 30 |
|------------|---------------------------|---------------------------|
| 10         | 2, 10, 18, **30**         | **19**, 25, 33            |
| 20         | 8, 18, **30**             | 1, **19**, 37             |
| 30         | 1, 15, 25, 26, **30**     | 1, **19**, 25, 37         |
| 40         | 8, 21, 25, **30**, 31     | **19**, 21, 33, 34        |
| 50         | 9, 17, **30**             | 11, 16, 17, **19**       |
| 60         | 8, 11, 22, **30**         | 11, **19**, 21, 22, 31   |

To evaluate this result, we examine cell types 19 and 30 using two-dimensional t-SNE visualization across a wide range of perplexity values (perplexity is the key hyperparameter of t-SNE). The t-SNE plots (Figure 9) show that the two cell types are always direct neighbors of each other (Table 8 summarizes the direct neighbors of each cell type in each t-SNE plot). We confirm this result by hierarchical clustering: the two cell types share similar gene expression patterns and are barely distinguishable in the first 40 principal components (Figure 10, left panel). As a control, we apply hierarchical clustering to distinguishing cell type 16 (“i_neuron_en3”: neuronal cells of the interstitial lineage), whose number of cells is closest to that of cell type 19, from cell type 30. The data is available at Broad Institute’s Single Cell Portal [https://singlecell.broadinstitute.org/single_cell/study/SCP260/](https://singlecell.broadinstitute.org/single_cell/study/SCP260/).
result shows that, unlike cell types 19 and 30, cell types 16 and 30 are well separated (Figure 10, right panel). Together, these evidence verifies the similarity of cell types 19 and 30, suggesting that ITCA can serve as a useful tool for identifying similar cell types and refining cell type annotations.

Figure 9: Visualizations of the cells in the hydra single-cell RNA-seq dataset using t-SNE under six perplexity values from 10 to 60. The cell types 19 and 30 are marked by triangles and black circles.

5. CONCLUSION

We introduce ITCA, an information-theoretic criterion for combining ambiguous outcome labels in classification tasks; typical examples are in medical and social sciences where class labels are often defined subjectively. ITCA automatically balances the trade-off between the increase in classification accuracy and the loss of class resolution, providing a data-driven criterion to guide class combination. The simulation studies validate the effectiveness of ITCA and the proposed search strategies. The four real-world applications demonstrate the wide application potential of
Figure 10: Heatmaps of the first 40 principal components of the hydra single-cell RNA-seq data. Row colors indicate the cell types: green for cell type 19 (458 cells; left and right panels), purple for cell type 30 (134 cells; left panel) and pink for cell type 16 (143 cells; right panel). Hierarchical clustering can hardly distinguish cell types 19 and 30 (left panel) but can well separate cell types 16 and 19 (right panel).

ITCA.

One merit of ITCA is its universality: it can be applied with any classification algorithm without modification. While ITCA has an implicit trade-off between classification accuracy and class resolution, it is worth extending ITCA to incorporate user-specified weights for classification accuracy and class resolution. Another open question is how to incorporate users’ predefined classes’ importance to the ITCA definition; that is, users may prefer some important classes to stay as uncombined.

In addition to classification problems, ITCA may potentially serve as a model-free criterion for determining the number of clusters in clustering problems. We will investigate this direction in future work.
References

Patrick L Brockett, Richard A Derrig, Linda L Golden, Arnold Levine, and Mark Alpert. Fraud classification using principal component analysis of ridits. *Journal of Risk and Insurance*, 69(3):341–371, 2002.

Peter WF Wilson, Ralph B D’Agostino, Daniel Levy, Albert M Belanger, Halit Silbershatz, and William B Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

Stephen F Weng, Jenna Reps, Joe Kai, Jonathan M Garibaldi, and Nadeem Qureshi. Can machine-learning improve cardiovascular risk prediction using routine clinical data? *Plos One*, 12(4):e0174944, 2017.

Li Chen, Peng Jin, and Zhaohui S Qin. DIVAN: accurate identification of non-coding disease-specific risk variants using multi-omics profiles. *Genome Biology*, 17(1):252, 2016.

Jonathan Krause, Varun Gulshan, Ehsan Rahimy, Peter Karth, Kasumi Widner, Greg S Corrado, Lily Peng, and Dale R Webster. Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy. *Ophthalmology*, 125(8):1264–1272, 2018.

Uwe Feldmann and Ingo Steudel. Methods of ordinal classification applied to medical scoring systems. *Statistics in Medicine*, 19(4):575–586, 2000.

Harry Hemingway, Peter Croft, Pablo Perel, Jill A Hayden, Keith Abrams, Adam Timmis, Andrew Briggs, Ruzan Udumyan, Karel GM Moons, Ewout W Steyerberg, et al. Prognosis research strategy (progress) 1: a framework for researching clinical outcomes. *BMJ*, 346, 2013.

Richard D Riley, Jill A Hayden, Ewout W Steyerberg, Karel GM Moons, Keith Abrams, Panayiotis A Kyzas, Núria Malats, Andrew Briggs, Sara Schroter, Douglas G Altman, et al. Prognosis research strategy (progress) 2: prognostic factor research. *Plos Med*, 10(2):e1001380, 2013.

Ewout W Steyerberg, Karel GM Moons, Danielle A van der Windt, Jill A Hayden, Pablo Perel, Sara Schroter, Richard D Riley, Harry Hemingway, Douglas G Altman, PROGRESS Group, et al.
Prognosis research strategy (progress) 3: prognostic model research. *Plos Med*, 10(2):e1001381, 2013.

Minal S Kale and Deborah Korenstein. Overdiagnosis in primary care: framing the problem and finding solutions. *BMJ*, 362:k2820, 2018.

Peter K Lindenauer, Tara Lagu, Meng-Shiou Shieh, Penelope S Pekow, and Michael B Rothberg. Association of diagnostic coding with trends in hospitalizations and mortality of patients with pneumonia, 2003-2009. *JAMA*, 307(13):1405–1413, 2012.

J. I. Epstein, L. Egevad, M. B. Amin, B. Delahunt, J. R. Srigley, and P. A. Humphrey. Consensus conference on gleason grading of prostatic carcinoma: definition of grading patterns and proposal for a new grading system. *The American Journal of Surgical Pathology*, 40(2):244–252, 2015.

Criteria Committee of the New York Heart Association et al. *Nomenclature and criteria for diagnosis of diseases of the heart and great vessels*, volume 253. Boston: Little, Brown & Co, 1994.

Paul Muntner, Robert M Carey, Samuel Gidding, Daniel W Jones, Sandra J Taler, Jackson T Wright, and Paul K Whelton. Potential US population impact of the 2017 ACC/AHA high blood pressure guideline. *Journal of the American College of Cardiology*, 71(2):109–118, 2018.

Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420, 2018.

Benoit Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2013.

Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, 2001.

Eyal Beigman and Beata Beigman Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, 2009.
Wensheng Zhang, Romdhane Rekaya, and Keith Bertrand. A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. *Bioinformatics*, 22(3):317–325, 2006.

Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. Support vector machine for outlier detection in breast cancer survivability prediction. In *Asia-pacific Web Conference*, pages 99–109. Springer, 2008.

Tim B Swartz, Yoel Haitovsky, Albert Vexler, and Tae Y Yang. Bayesian identifiability and misclassification in multinomial data. *Canadian Journal of Statistics*, 32(3):285–302, 2004.

Hyun-Chul Kim and Zoubin Ghahramani. Outlier robust gaussian process classification. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, pages 896–905. Springer, 2008.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

Nikhil R Pal et al. Uncertainty, entropy and maximum entropy principle—an overview. In *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, pages 1–53. Springer, 2003.

Eric Temple Bell. The iterated exponential integers. *Annals of Mathematics*, pages 539–557, 1938.

Willliam Feller. *An introduction to probability theory and its applications, vol 2*. John Wiley & Sons, 2008.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14:849–856, 2001.
Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.

Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

Joseph H Ricker. Traumatic brain injury rehabilitation: Is it worth the cost? *Applied Neuropsychology*, 5(4):184–193, 1998.

Lynne Turner-Stokes. Evidence for the effectiveness of multi-disciplinary rehabilitation following acquired brain injury: a synthesis of two systematic approaches. *Journal of Rehabilitation Medicine*, 40(9):691–701, 2008.

John Michael Linacre, Allen W Heinemann, Benjamin D Wright, Carl V Granger, and Byron B Hamilton. The structure and stability of the functional independence measure. *Archives of Physical Medicine and Rehabilitation*, 75(2):127–132, 1994.

Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E. Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J. Byrne, Michael L. Heuer, Erik Larsson, Yevgeniy Antipin, Boris Reva, Arthur P. Goldberg, Chris Sander, and Nikolaus Schultz. The cBio cancer genomics portal: An open platform for exploring multidimensional cancer genomics data. *Cancer Discovery*, 2(5):401–404, 2012.

David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (methodological)*, 34(2):187–202, 1972.

Kaggle. Talkingdata mobile user demographics, 2016. URL https://www.kaggle.com/c/talkingdata-mobile-user-demographics.
Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

Guoying Wei. Gender comparison of employment and career development in China. *Asian Women*, 27(1):95–114, 2011.

Yong Wang and Nicholas E Navin. Advances and applications of single-cell sequencing technologies. *Molecular Cell*, 58(4):598–609, 2015.

Efthymia Papalexi and Rahul Satija. Single-cell rna sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology*, 18(1):35, 2018.

Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, 2019.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008 (10):P10008, 2008.

Stefan Siebert, Jeffrey A Farrell, Jack F Cazet, Yashodara Abeykoon, Abby S Primack, Christine E Schnitzler, and Celina E Juliano. Stem cell differentiation trajectories in hydra resolved at single-cell resolution. *Science*, 365(6451):eaav9314, 2019.