Learnability with Partial Labels and Adaptive Nearest Neighbors

Nicolas A. Errandonea[1] **Santiago Mazuelas**[1,2] **Jose A. Lozano**[1,3] **Sanjoy Dasgupta**[4]
[1] Basque Center for Applied Mathematics, Bilbao, Spain
[2] Ikerbasque, Basque Foundation for Science
[3] University of the Basque Country, Spain
[4] University of California San Diego, USA
{nerrandonea, smazuelas, jlozano}@bcamath.org, sadasgupta@ucsd.edu

Abstract

Prior work on partial labels learning (PLL) has shown that learning is possible even when each instance is associated with a bag of labels, rather than a single accurate but costly label. However, the necessary conditions for learning with partial labels remain unclear, and existing PLL methods are effective only in specific scenarios. In this work, we mathematically characterize the settings in which PLL is feasible. In addition, we present PL A-kNN, an adaptive nearest-neighbors algorithm for PLL that is effective in general scenarios and enjoys strong performance guarantees. Experimental results corroborate that PL A-kNN can outperform state-of-the-art methods in general PLL scenarios.

1 INTRODUCTION

Partial labels learning (PLL) is a weakly supervised framework in which each training instance is associated with a bag of labels, rather than the ground-truth label (Nguyen and Caruana, 2008; Cour et al., 2011; Tian et al., 2023). The goal in PLL is to train a classifier that can accurately predict the true label for unseen instances, overcoming the ambiguity introduced by the bags. A typical example of partial labels arises in medical imaging annotation, where non-expert annotators may provide multiple labels when uncertain about the correct label of an image. Interest in PLL has continued to grow in recent years as it offers an attractive surrogate to supervised learning, since accurately labeled datasets often require costly expert annotation. PLL remains a challenging problem due to the diverse ways in which bags can be generated, often resulting in highly ambiguous training data.

Existing theoretical results for PLL have established distributional conditions that are sufficient to recover the Bayes rule while learning from bags of labels (Cour et al., 2011; Liu and Dietterich, 2014; Cabannes et al., 2020; Lv et al., 2020; Wen et al., 2021; Lv et al., 2023). Among these, Cour et al. (2011) introduced the label-aligned condition as a realistic and intuitive assumption for the bags of labels. Specifically, for label-aligned bags, the most probable label of each instance is also the most probable label to appear in its bag of labels obtained at training. The label-aligned condition encompasses most of the other proposed assumptions as special cases. Moreover, Cour et al. (2011) suggested that the label-aligned condition is also intuitively necessary for learning in partial labels settings, though no formal theoretical results were provided. Despite substantial progress identifying sufficient conditions for recovering the Bayes rule, the fundamental properties that enable learning in a partial labels setting remain unclear.

Multiple successful algorithmic approaches for PLL have been proposed in recent years. Existing methods obtain information from the training data by either trying to disambiguate the ground-truth label from each bag (Zeng et al., 2013; Liu and Dietterich, 2012; Xu et al., 2023; Jia et al., 2024), or treating all the labels in the bags as ground-truth labels (Hüllermeier and Beringer, 2006; Cour et al., 2011; Zhang and Yu, 2015; Zhou and Gu, 2018; Cabannes et al., 2020). A popular approach among the latter methods is the nearest neighbors (k-NN) algorithm (Hüllermeier and Beringer, 2006), which predicts an instance's label as the most frequent label among the k neighboring bags.

Existing PLL methods can obtain accurate classification and provide strong performance guarantees only in specific scenarios. The approaches in Hüllermeier and

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Beringer (2006); Cour et al. (2011); Zeng et al. (2013); Cabannes et al. (2020); Feng et al. (2020); Wen et al. (2021); Xu et al. (2023); Jia et al. (2024) are based on the assumption that the bags of labels always contain the ground-truth label. This assumption often breaks down in real-world scenarios, where annotators may omit the correct label, producing noisy partial labels (Cid-Sueiro, 2012). The methods in Feng et al. (2020); Wen et al. (2021); Lv et al. (2023) rely on the assumption that the bags' characteristics are the same for all the instances, thereby not considering cases where the bag generation process varies across the instance space. Moreover, the theoretical guarantees for the existing methods are established under additional restrictive assumptions, such as assuming deterministic labels, and only cover specific cases of label-aligned bags (Cour et al., 2011; Cabannes et al., 2020; Feng et al., 2020; Wen et al., 2021; Lv et al., 2023).

In this work, we mathematically characterize the conditions that can enable learning in a partial labels setting. In addition, we introduce PL A-kNN, an adaptive nearest neighbors algorithm for partial labels that is effective in general scenarios, and enjoys strong theoretical guarantees. The main contributions of the paper are listed as follows:

- We characterize the PLL settings in which it is possible to overcome the ambiguity introduced by the bags and recover the Bayes rule of the underlying distribution.
- We propose PL A-kNN, an adaptive nearest-neighbor approach for PLL that tailors the number of neighbors for each instance according to the bags of the surrounding instances.
- We show that PL A-kNN is Bayes consistent for any case of label-aligned bags, and that no other algorithm can be consistent under more general PLL scenarios than PL A-kNN.
- The experimental results demonstrate that PL A-kNN can outperform state-of-the-art methods in general scenarios. In addition, PL A-kNN can achieve accuracies on pair with the ideal nearest neighbors method that uses the optimal number of neighbors.

2 LEARNABILITY WITH PARTIAL LABELS

2.1 Preliminaries

The instance space is considered to be a finite dimensional normed space, denoted as $\mathcal{X}$. Let $\mathcal{Y} = \{1, 2, \ldots, c\}$ represent the label space and $\mathcal{S} = 2^{\mathcal{Y}}$ represent the space of the bags of labels. Data are assumed to be drawn i.i.d. from an unknown distribution P over $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$. However, only pairs of instances and bags $(x, s) \in \mathcal{X} \times \mathcal{S}$ are available for learning. The goal in PLL is to learn a classifier $h : \mathcal{X} \to \mathcal{Y}$ from a set of n training examples $\{(x_l, s_l)\}_{l=1}^{n}$.

The risk of a classification rule h is the probability with which the rule misclassifies a tuple (x, y) drawn i.i.d. from distribution P, that is:

$$\mathcal{R}(h) = \mathbb{E}_{(x,y)\sim P}\left[\mathbb{I}\{h(x) \neq y\}\right].$$

The Bayes risk $\mathcal{R}^*$ is the smallest possible risk and is achieved by the Bayes rule h^*, that predicts the most probable label for each instance. An algorithm is Bayes consistent if its associated classifier asymptotically achieves the Bayes risk as the number of training instances grow.

For each instance x, the distribution over bags is linked to the label distribution $P(y|x)$ as follows

$$P(s|x) = \sum_{y\in\mathcal{Y}} P(s|y, x)P(y|x).$$

We refer to the conditional probabilities $P(s|y, x)$ as the bag generation process, since they describe how bags are generated given instance-label pairs. A partial labels setting is determined by a bag generation process.

For each instance $x \in \mathcal{X}$, the bag generation process is characterized by the $|\mathcal{Y}|$ vectors $\mathbf{p}_{1,x}, \mathbf{p}_{2,x}, \ldots, \mathbf{p}_{|\mathcal{Y}|,x} \in \mathbb{R}^{|\mathcal{S}|}$ given by

$$\mathbf{p}_{i,x} = \begin{bmatrix} P(s_1|i, x) \\ P(s_2|i, x) \\ \vdots \\ P(s_{|\mathcal{S}|}|i, x) \end{bmatrix} \in \mathbb{R}^{|\mathcal{S}|} \text{ for } i \in \mathcal{Y}. \tag{1}$$

Each component of the vector corresponds to the probability of generating a bag in $\mathcal{S} = \{s_1, s_2, ...s_{|\mathcal{S}|}\}$ when the true label is i.

2.2 Learnability with partial labels

Learning in a partial labels setting is not always feasible, as even an infinite amount of samples may not be enough to recover the Bayes rule of the underlying distribution. Specifically, two probability distributions can share the same distribution over bags but yield different Bayes rules. Previous work has proposed several sufficient conditions for recovering the Bayes rule (Cour et al., 2011; Liu and Dietterich, 2014; Cabannes et al., 2020; Lv et al., 2023), but the conditions that enable learning with partial labels remain unclear.

As discussed above, the Bayes rule cannot be recovered in partial labels settings in which distributions with

different Bayes rules result in the same distribution over bags. Therefore, we define a bag generation process as reconstructible if any two probability distributions that abide by it and share the same distribution over bags also yield the same Bayes rule. If a bag generation process is reconstructible, it is possible to recover the Bayes rule from the distribution over bags.

Definition 2.1. A bag generation process $P(s|y,x)$ is *reconstructible* if for all $x \in \mathcal{X}$ we have that $\sum_{y\in\mathcal{Y}} P(s|y,x)Q_1(y) = \sum_{y\in\mathcal{Y}} P(s|y,x)Q_2(y)$ for $Q_1, Q_2 \in \Delta(\mathcal{Y})$ implies

$$\arg\max_{y\in\mathcal{Y}} Q_1(y) = \arg\max_{y\in\mathcal{Y}} Q_2(y).$$

The following theorem characterizes the necessary and sufficient condition that a bag generation process has to satisfy to be reconstructible.

Theorem 2.2. A bag generation process $P(s|y,x)$ is reconstructible if and only if, the vectors $\mathbf{p}_{1,x}, \mathbf{p}_{2,x}, \ldots, \mathbf{p}_{|\mathcal{Y}|,x} \in \mathbb{R}^{|\mathcal{S}|}$ in (1) are linearly independent for any x.

Proof.

We define $M(x) = [\mathbf{p}_{1,x}, \mathbf{p}_{2,x}, \ldots, \mathbf{p}_{|\mathcal{Y}|,x}] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{Y}|}$, the matrix that characterizes the bag generation process for instance x. The entries of the matrix satisfy $M_{j,i}(x) = P(s_j|i,x)$ for any $x \in \mathcal{X}, i \in \mathcal{Y}, s_j \in \mathcal{S}$.

Sufficiency: If for each $x \in \mathcal{X}$ the vectors

$$\mathbf{p}_{1,x}, \mathbf{p}_{2,x}, \ldots, \mathbf{p}_{|\mathcal{Y}|,x} \in \mathbb{R}^{|\mathcal{S}|} \tag{2}$$

are linearly independent, and we have a pair of distributions $Q_1, Q_2 \in \Delta(\mathcal{Y})$ such that

$$\sum_{y\in\mathcal{Y}} P(s|y,x)\, Q_1(y) \;=\; \sum_{y\in\mathcal{Y}} P(s|y,x)\, Q_2(y) \quad \forall s \in \mathcal{S},$$

then $M(x)(Q_1 - Q_2) = 0$. Since the vectors in (2) are linearly independent, $M(x)$ has full column rank, which implies that $Q_1 = Q_2$. In particular,

$$\arg\max_{y\in\mathcal{Y}} Q_1(y) = \arg\max_{y\in\mathcal{Y}} Q_2(y),$$

hence the bag generation process is reconstructible.

Necessity: This proof is obtained by contradiction. If the bag generation process is reconstructible, and for some $x_0 \in \mathcal{X}$, the vectors

$$\mathbf{p}_{1,x_0}, \mathbf{p}_{2,x_0}, \ldots, \mathbf{p}_{c,x_0}$$

are linearly dependent, then there would exist a nonzero vector $q \in \mathbb{R}^c$ such that $M(x_0)q = 0$.

Since the columns of $M(x_0)$ add to one, we also have that $\sum_{y\in\mathcal{Y}} q = 0$. Then, we could split q into its positive-negative parts:

$$q_+(y) = \max\{0, q(y)\},$$
$$q_-(y) = \max\{0, -q(y)\}, \quad y \in \mathcal{Y}.$$

Clearly, $q = q_+ - q_-$, and since $\sum_{y\in\mathcal{Y}} q = 0$, we have that both q_+ and q_- are nonzero.

Normalizing the vectors q_+ and q_- would give two different label distributions:

$$Q_1 = \frac{q_+}{\sum_{y\in\mathcal{Y}} q_+(y)}, \qquad Q_2 = \frac{q_-}{\sum_{y\in\mathcal{Y}} q_-(y)} \in \Delta(\mathcal{Y}),$$

in which $\sum_{y\in\mathcal{Y}} q_+(y) = \sum_{y\in\mathcal{Y}} q_-(y)$, as $\sum_{y\in\mathcal{Y}} q = 0$. Then, we would have

$$M(x_0)(Q_1 - Q_2) \;=\; \frac{1}{\sum_y q_+(y)} M(x_0)q \;=\; 0.$$

Therefore, we would have $M(x_0)Q_1 = M(x_0)Q_2$, or equivalently

$$\sum_{y\in\mathcal{Y}} P(s|y,x_0)\, Q_1(y) = \sum_{y\in\mathcal{Y}} P(s|y,x_0)\, Q_2(y) \quad \forall s \in \mathcal{S}.$$

However, since q_+, q_- are the positive-negative parts of q we would have

$$\arg\max_{y\in\mathcal{Y}} Q_1(y) \neq \arg\max_{y\in\mathcal{Y}} Q_2(y),$$

which contradicts the reconstructibility of $P(s|y,x)$.□

The theorem above presents the necessary and sufficient condition for learning in a partial labels setting, establishing that the bag generation process has to satisfy a linear independence condition. Theorem 2.2 provides the first complete characterization of learnability with partial labels. Previous work in Feng et al. (2020) already proved that the linear independence of $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{|\mathcal{Y}|} \in \mathbb{R}^{|\mathcal{S}|}$ is sufficient for learning when the bag generation process does not vary across instances. Theorem 2.2 generalizes the result from Feng et al. (2020) and shows that the linear independence assumption is in fact a necessary condition. In contrast, the other existing sufficiency results (Liu and Dietterich, 2014; Cabannes et al., 2020) are oriented to specific assumptions that enable the recovery of the Bayes rule with a given procedure.

Contrary to the suggestion of Cour et al. (2011), Theorem 2.2 shows that the label-aligned condition is not necessary for learning in a partial labels setting. Specifically, there exist bag generation processes that are reconstructible but do not correspond to label-aligned scenarios. For instance, the process defined by $P(s|y,x) = 1$ if $s = \{\pi_x(y)\}$ for some permutation π_x of the labels, and $P(s|y,x) = 0$ otherwise,

is reconstructible, since it satisfies the conditions of Theorem 2.2. Such bag generation process would only be label-aligned if π_x is the identity permutation for any instance x. However, even though such a case is reconstructible, it cannot be realistically addressed by an algorithm, since it would require knowing the permutation π_x for every instance x.

As discussed above, full access to the bag generation process $P(s|y,x)$ can enable effective learning in any reconstructible setting, but such a detailed knowledge is highly unrealistic in practice. Therefore, we seek more practical assumptions under which an algorithm can recover the Bayes rule. The following further analyses the label-aligned condition, as a simple general assumption that does not require detailed knowledge of the bag generation process.

2.3 Label-aligned assumption

The label-aligned scenarios of PLL satisfy that, for each instance, the label that appears more frequently in its bags is also the most probable label. Specifically, let

$$\mathcal{S}_y = \{s \in \mathcal{S} : y \in s\}$$

be the set of bags that include label y. The label-aligned condition requires that

$$\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x) = \arg\max_{y\in\mathcal{Y}} P(y|x) \quad \forall x \in \mathcal{X}, \qquad (3)$$
$$\text{where} \quad P(\mathcal{S}_y|x) = \sum_{s\in\mathcal{S}_y} P(s|x).$$

The following result indicates that bag generation processes that result in label-aligned scenarios are always reconstructible

Definition 2.3. A bag generation process $P(s|y,x)$ is *label-aligned* if any distribution where the bags are generated according to $P(s|y,x)$ satisfies the label-aligned condition (3).

Corollary 2.4. If a bag generation process $P(s|y,x)$ is label-aligned, then it is also reconstructible.

Proof. See Appendix C. □

The label-aligned condition is more general than most of the assumptions taken in the literature. In Appendix A we show that the assumptions taken for six partial-label models from Cour et al. (2011); Liu and Dietterich (2014); Cabannes et al. (2020); Feng et al. (2020); Wen et al. (2021); Lv et al. (2023) are specific cases of the label-aligned condition. Furthermore, Theorem 3.1 in Section 3 establishes that the label-aligned condition is a minimal algorithmic assumption.

In the following section we introduce PL A-kNN, an effective algorithm in general label-aligned scenarios. PL A-kNN predicts the most probable label of an instance by computing the empirical frequencies of the labels in the bags of the neighbors.

3 THE PL A-KNN ALGORITHM

This section first details the proposed PL A-kNN algorithm. Then, we provide a consistency result for PL A-kNN, along with rates of convergence, for any label-aligned case. We conclude the section showing the robustness of the algorithm in cases that depart from the label-aligned condition.

The PL A-kNN algorithm gradually whittles down the potential labels of an instance by comparing the frequencies of the labels among the bags of an increasing set of neighbors. The algorithm uses fewer neighbors when the most frequent label has a clear lead over the frequencies of the other labels in the bags, and uses more neighbors when the label frequencies are close.

PL A-kNN initially considers the set with all labels $\hat{s} = \mathcal{Y}$ as the potential labels of instance x. At step k, the algorithm computes the frequencies of the labels in $\hat{s}$ over the bags of the k nearest neighbors of x. Labels with frequencies that deviate from the maximum frequency by more than threshold

$$\Delta(n,k,\delta) = c_1\sqrt{\frac{\log(n)+\log(|\mathcal{Y}|/\delta)}{k}}$$

are removed from $\hat{s}$. The pseudo-code for PL A-kNN is provided in Algorithm 1.

Algorithm 1 PL A-kNN algorithm

Inputs: Instance x
Training examples $\{(x_l, s_l)\}_{l=1}^n$
Maximum number of iterations T
Confidence parameter δ
Output: Label $h(x)$

Set $A = c_1\sqrt{\log(n)+\log(|\mathcal{Y}|/\delta)}$
Initialize $\hat{s} = \mathcal{Y}$, $k = 0$, and $\tau_1, \tau_2 ..., \tau_{|\mathcal{Y}|} = 0$
while $|\hat{s}| > 1$ and $k < T$ **do**
  $k = k + 1$
  Find the k nearest neighbor of x and take l_k as its index in $l = 1, 2, ...n$
  $\Delta = \frac{A}{\sqrt{k}}$
  $\tau_y = \tau_y + \mathbb{I}\{y \in s_{l_k}\} \quad \forall y \in \mathcal{Y}$
  $m = \max_{y\in\hat{s}} \tau_y$
  for $y \in \hat{s}$ **do**
    if $\frac{m-\tau_y}{k} \geq \Delta$ **then**
      $\hat{s} = \hat{s} \setminus \{y\}$
    end if
  end for
end while
$h(x) = \hat{s}$

The algorithm uses a maximum number of iterations T, which is set to a value significantly smaller than the number of samples, since bags from very distant neighbors are poor representatives of the bag corresponding to x. If after T iterations $|\hat{s}| > 1$, we require a disambiguation criterion. In this paper we propose a simple disambiguation process in which we select the label in $\hat{s}$ that came closest to eliminating all other labels during the iterative process. The detailed pseudo code for PL A-kNN with the disambiguation criterion is provided in Appendix D.

The PL A-kNN algorithm is inspired by the adaptive k-nearest neighbor method for binary supervised classification (A-kNN) from Balsubramani et al. (2019). PLL requires to consider multiclass settings, and the proposed PL A-kNN differs fundamentally from the multiclass supervised classification extension suggested in Balsubramani et al. (2019). Such an extension expands the neighborhood size until the frequency of a label is higher than threshold Δ, and then predicts that label. In contrast, PL A-kNN gradually enlarges the neighborhood while discarding labels that deviate from the maximum frequency by more than threshold Δ, narrowing down the potential labels of the instance. Moreover, PL A-kNN is backed by strong theoretical results, whereas the extension in Balsubramani et al. (2019) is not provided with any guarantee, since the theoretical results in that work are only valid for the binary method. The difference in performance guarantees lies in how the two methods exploit label frequencies: PL A-kNN leverages the margins between label frequencies, while A-kNN only considers which is the most frequent label and disregards the rest of the information. A more detailed comparison between these two approaches is provided in Appendix B.

3.1 Consistency of PL A-kNN

The following theorem shows that the PL A-kNN algorithm is Bayes consistent. Specifically, the risk of PL A-kNN converges almost surely to the Bayes risk as the number of samples n grows. The sequence of the confidence parameters $\{\delta_n\}_{n=1}^{\infty}$ and maximum number of iterations $\{T_n\}_{n=1}^{\infty}$ has to satisfy the following asymptotic properties

$$\sum_{n=1}^{\infty} \delta_n < \infty, \quad \lim_{n\to\infty} \frac{\log(1/\delta_n)}{T_n} = 0, \qquad (4)$$
$$\exists A \in (0,1] \text{ such that } T_n \geq An \ \forall n.$$

Theorem 3.1. (Consistency) Let h_n be the classifier from the PL A-kNN Algorithm 1 with parameters that satisfy (4). If the underlying probability distribution satisfies the label-aligned condition (3), we have

$$\lim_{n\to\infty} \mathcal{R}(h_n) = \mathcal{R}^*$$

almost surely. In addition, no other algorithm can be Bayes consistent under more general PLL scenarios than PL A-kNN.

Proof. See Appendix C. □

The previous theorem shows that PL A-kNN is Bayes consistent under more general assumptions than the other existing methods in the literature. Specifically, Appendix A shows that the assumptions taken for the consistency guarantees of the algorithms from Cour et al. (2011); Cabannes et al. (2020); Feng et al. (2020); Wen et al. (2021) are specific cases of the label-aligned condition. There is one model proposed in Lv et al. (2023) that shows consistency for cases where the bags are not label-aligned. However, such work assumes that the bag generation process is identical across instances, overlooking scenarios in which the bag generation varies across the instance space, and establishes the consistency result only under deterministic labels

The previous theorem also demonstrates that PL A-kNN is consistent under minimal algorithmic assumptions. In other words, an algorithm cannot achieve Bayes consistency under strictly more general scenarios than those where PL A-kNN is consistent. To the best of our knowledge, PL A-kNN is the first algorithm for PLL that is proven consistent under minimal algorithmic assumptions.

3.2 Rates of convergence

In what follows, we first establish instance-specific convergence rates, and subsequently derive the standard convergence rates. These rates depend on the advantage of each instance x, which quantifies the margin of the most frequent labels ($\arg\max_{y\in Y} P(\mathcal{S}_y|x)$) with the other labels in the bags surrounding x. The concept of advantage we propose in this paper is an extension to PLL of the concept introduced in Balsubramani et al. (2019) for binary supervision.

The advantage of an instance x depends on the frequencies of labels in the bags for instances in balls centered at x. We denote the closed ball B centered in $x \in \mathcal{X}$ with radius r as:

$$B(x,r) = \{x' \in \mathcal{X} : \|x - x'\| \leq r\}.$$

We then denote as $r_p(x)$ the smallest radius such that ball $B(x, r_p(x))$ has probability mass of at least p, that is:

$$r_p(x) = \inf\{r \geq 0 : P(B(x,r)) \geq p\}.$$

A point $x \in \mathcal{X}$ is (p,γ)-salient if the following holds:

$$\text{for any } i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x) \text{ and } \ j \notin \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x),$$
$$P(\mathcal{S}_i|B(x,r)) > P(\mathcal{S}_j|B(x,r)) + \gamma \quad \forall r \in [0, r_p(x)].$$

A point x can satisfy this definition for a variety of (p,γ) tuples. The advantage of $x \in \mathcal{X}$ in the region $B(x, r_A(x))$ is taken to be the largest value of $p\gamma^2$ out of all the (p,γ)-salient tuples with $p \leq A$, that is

$$\mathrm{adv_A}(x) = \begin{cases} 1 & \text{if } \arg\max_{y \in Y} P(\mathcal{S}_y \mid x) = \mathcal{Y}, \\ \sup\{p\gamma^2 : x \text{ is } (p,\gamma)\text{-salient}, p \leq A\} & \text{else.} \end{cases}$$

Provided that A is not too small, the advantage within $B(x, r_A(x))$ is generally the same as in the entire space (the ball of mass 1), since instances are commonly (p,γ)-salient only within a local region of the space. Moreover, it always holds that at least $\mathrm{adv_1}(x) \leq \frac{\mathrm{adv_A}(x)}{A}$.

Large advantage values for x indicate that it is easier to distinguish the labels that are the most frequent labels in the bags of x with the bags of neighboring instances.

The following theorem shows that PL A-kNN classifies x as the Bayes rule $h^*(x)$ with probability at least $1-\delta^2$ when the number of samples is of the order of $1/\mathrm{adv_A}(x)$, with A given by (4).

Theorem 3.2. (Query-specific convergence) There is an absolute constant $C > 0$ for which the following holds. Let h be the classifier from PL A-kNN Algorithm 1 with maximum number of iterations T and confidence parameter δ satisfying (4). If the underlying probability distribution satisfies the label-aligned condition (3), for each $x \in \mathcal{X}$ such that

$$n \geq \frac{C}{\mathrm{adv_A}(x)} \max\left\{\log\frac{1}{\mathrm{adv_A}(x)}, \log\frac{|\mathcal{Y}|}{\delta}\right\}. \quad (5)$$

we have $h(x) = h^*(x)$ with probability at least $1-\delta^2$. In addition, the set of instances x for which $\mathrm{adv_A}(x) = 0$ has zero measure.

Proof. See Appendix C. □

The theorem above directly relates the advantage of an instance with the number of samples required by PL A-kNN to ensure its correct classification. The criterion (5) of the theorem assesses whether there are enough "advantageous" training examples in the neighborhood of x for PL A-kNN to predict the most most probable label for x after at most T iterations.

The query-specific rates from Theorem 3.2 are tailored to each instance, depending only on the local properties of the bags. On the other hand, the rates of convergence of other methods (Cabannes et al., 2020; Feng et al., 2020; Lv et al., 2023) are with respect to the overall risk of the classifier.

The bounds of the Theorem 3.2 hold for each instance x with high probability. We can strengthen this theorem so that the guarantee holds with high probability simultaneously for all elements in $\mathcal{X}$ by modifying the threshold for candidate elimination to

$$\Delta(n,k,\delta) = c_1\sqrt{\frac{d_0\log(n) + \log(|\mathcal{Y}|/\delta)}{k}} \quad (6)$$

where d_0 denotes the VC dimension of the set of balls in $\mathcal{X}$.

Theorem 3.3. (Uniform query-specific convergence) Let h be the classifier from PL A-kNN Algorithm 1, with maximum number of iterations T and δ satisfying (4), and Δ taken as in (6). If the underlying probability distribution satisfies the label-aligned condition (3), then, with probability at least $1-\delta^2$, we have that $h(x)$ coincides with the Bayes rule $h^*(x)$ for all $x \in \mathcal{X}$ that satisfy (5).

Proof. See Appendix C. □

Theorem 3.3 shows that the PL A-kNN algorithm with high probability can correctly classify all the instances that satisfy (5).

It is also possible to derive instance-specific rates for the elimination of each label that is not the most probable. Specifically, one can follow arguments analogous to those developed in this section by defining the advantage of the most probable label relative to each other label individually. Therefore, even for instances where the available sample size is insufficient to guarantee the identification of the optimal label, a similar per label analysis can ensure that certain suboptimal labels are eliminated from the set of possible labels.

We now provide rates of convergence to the Bayes risk for the PL A-kNN classifier that depend on the cumulative distribution of the advantage.

Theorem 3.4. (Rates of convergence) Let C be the constant from Theorem 3.2 and h be the classifier from PL A-kNN Algorithm 1 with maximum number of iterations T and confidence parameter δ satisfying (4). If the underlying probability distribution satisfies the label-aligned condition (3), with probability at least $1-\delta$, h satisfies

$$R(h) - R^* \leq \delta + P(\mathrm{adv_A}(x) \leq a_n)$$
$$\textit{where}\quad a_n = \frac{C}{n}\max\{2\log(n), \log(|\mathcal{Y}|/\delta)\}$$

Proof. See Appendix C. □

Theorem 3.4 bounds the excess risk of the PL A-kNN classifier with the Bayes risk in terms of the cumulative distribution of the advantage at a_n. Namely, the excess risk of PL A-kNN is upper bounded by the probability mass of the instances with advantage smaller than a_n.

The cumulative distribution of the advantage can be seen as a suitable notion of the "global ambiguity"

in a specific scenario. Steeper functions for this cumulative distribution mean that fewer instances have small advantage, thus providing faster rates of convergence. Therefore, Theorem 3.4 provides rates of convergence tailored to the PLL scenario, as the rates only depend on the global ambiguity of the underlying distribution. On the other hand, the rates of convergence provided for other methods (Cabannes et al., 2020; Feng et al., 2020; Lv et al., 2023) are given in terms of the Rademacher complexity of the family of classifiers considered and the maximum ambiguity in the scenario, thereby not fully adapting the bounds to the underlying distribution.

In Appendix B we provide explicit finite-sample rates of convergence by assuming two common smoothness conditions for non-parametric estimators over P.

3.3 Robustness of PL A-kNN under relaxed label-aligned conditions

The label-aligned condition (3) might not hold in some real-life scenarios. For example, the condition might not be satisfied in a small subset of instances. In other cases, the most frequent labels in the bags might not be the most probable labels but have conditional probabilities close to the maximum. Therefore, we weaken the original condition (3) and bound the behavior of the limiting risk of the PL A-kNN classifier in these cases.

For any instance x and threshold $\theta \in [0, 1]$, we define as $\mathcal{Y}^\theta(x)$ the set of labels whose conditional probability are within a margin θ of the most probable label. Specifically,

$$\mathcal{Y}^\theta(x) = \big\{i \in \mathcal{Y} : \; P(j|x) - P(i|x) \leq \theta \quad \forall j \in \mathcal{Y}\big\}.$$

We assume that bags of labels satisfy the following relaxed label-aligned condition for a subset $G \subseteq \mathcal{X}$ and $\theta \in [0, 1]$:

$$\begin{aligned} &\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x) = \arg\max_{y\in\mathcal{Y}} P(y|x) \quad && \forall x \in \mathcal{X} \setminus G \\ &\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x) \subseteq \; \mathcal{Y}^\theta(x), && \forall x \in G \end{aligned} \tag{7}$$

The following result bounds the limiting risk of the PL A-kNN algorithm when the underlying distribution satisfies the label-aligned relaxation (7).

Theorem 3.5. (Limiting risk bounds) Let h_n be a classifier from PL A-kNN Algorithm 1 with parameters that satisfy (4). If the underlying probability distribution satisfies the relaxed label-aligned condition (7) for $G \subseteq \mathcal{X}$ and θ , we have

$$\lim_{n\to\infty} \mathcal{R}(h_n) \leq \mathcal{R}^* + \theta P(G)$$

almost surely.

Proof. See Appendix C. □

The previous result shows that the PL A-kNN classifier provides reliable predictions when the underlying probability distribution satisfies a slight weakening of the label-aligned condition (3). In particular, the risk of PL A-kNN is close to the Bayes risk when only a small subset of instances fails to satisfy (3), or when the most frequent labels over the bags have conditional probabilities similar to that of the most probable label.

The results above show that the presented PL A-kNN method provides strong performance guarantees with minimal assumptions. The next section shows that the algorithm is effective in practice under general PLL scenarios.

4 EXPERIMENTAL RESULTS

In this section we evaluate the performance of PL A-kNN over a wide set of artificial partial label scenarios obtained from the MNIST, Fashion-MNIST and CIFAR10 datasets, as well as in two real-life partial labels datasets, Mirflickr Huiskes and Lew (2008) and MSRCv2 Liu and Dietterich (2012), under different levels of noise. For the vision datasets, we extract feature representations using GoogLeNet Szegedy et al. (2015) and subsequently apply principal component analysis to retain the 50 most informative components.

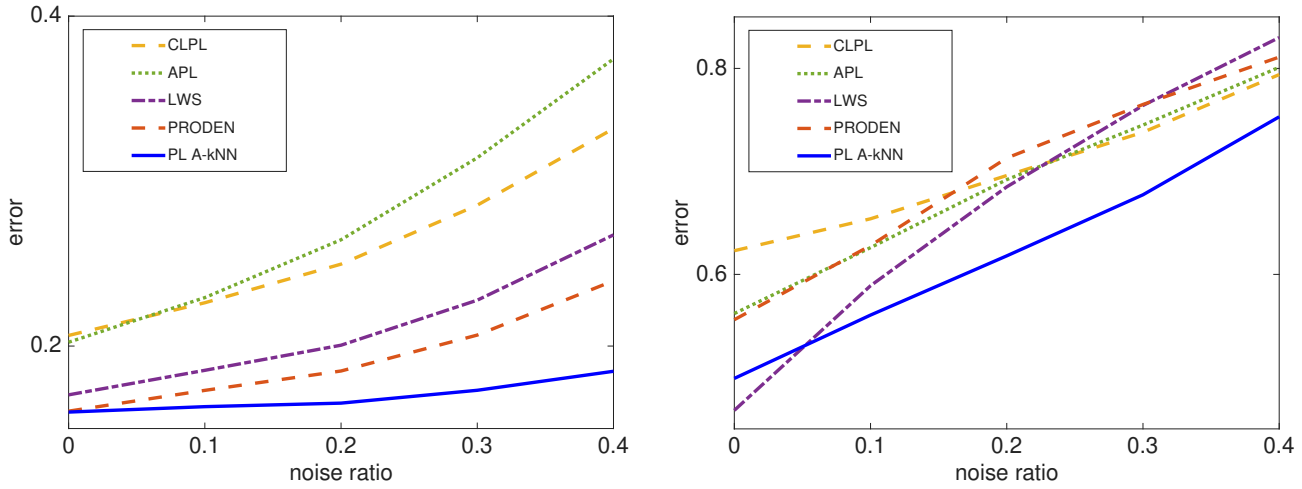
We compare PL A-kNN with four other state of the art partial labels methods with theoretical backing. In particular, we compare with the CLPL method from Cour et al. (2011), the LWS method from Wen et al. (2021), the PRODEN method from Lv et al. (2020) and the APL method from Lv et al. (2023). We use the linear variant of all models to provide a fair comparison and tune the models from previous work according to the specifications in their papers.

We compare PL A-kNN with three kNN benchmarks: the ideal kNN that uses the optimal number of neighbors; 10NN, which is the usual choice of k in previous works; and the multiclass extension of A-kNN from Balsubramani et al. (2019).

4.1 Generation of the artificial partially labeled scenarios

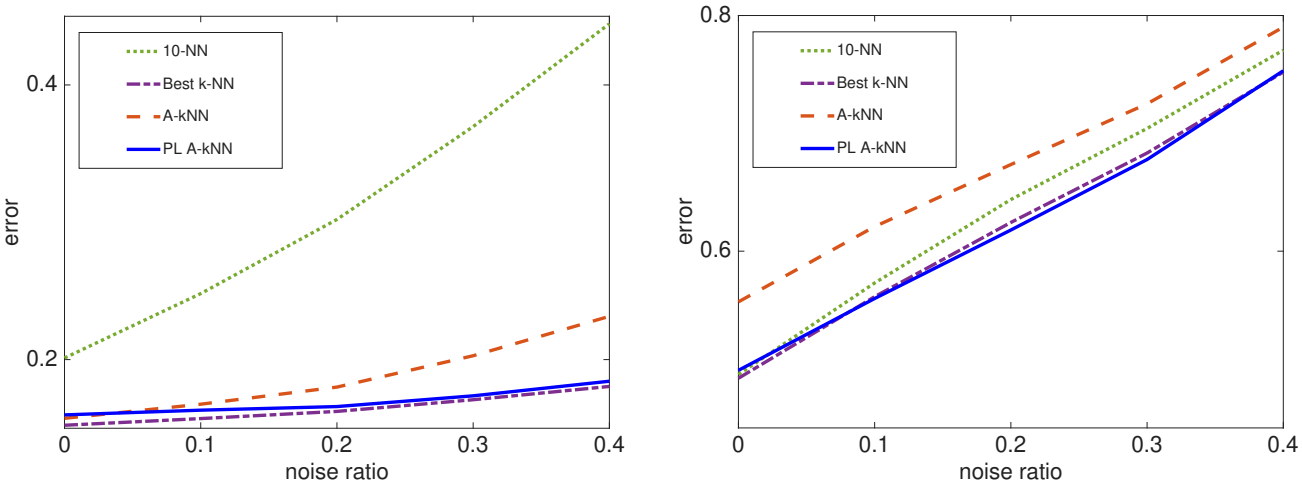
We construct partially labeled datasets from supervised datasets by generating bag generation processes that vary across instances and allow noise.

The bag generation process are generated as follows. We partition the training set into five clusters. For cluster j and label i we draw uniformly $\alpha_{i,j} \in [0, 0.8]$, which describe the probability of adding each incorrect label to the bag of instances in j with ground truth



(a) Fashion-MNIST (b) MSRCV2

Figure 1: Comparison of the error rates of PL A-kNN and state-of-the-art methods for Fashion-MNIST and MSCRv2 under an increasing noise rate. The results show that PL A-kNN consistently outperforms existing approaches across a wide range of noise levels. See Appendix D for the comparison in MNIST CIFAR10 and MirFlickr.



(a) Fashion-MNIST (b) MSRCV2

Figure 2: Comparison of the error rates of PL A-kNN and state-of-the-art methods for Fashion-MNIST and MSRCv2 under an increasing noise rate. The results show that PL A-kNN outperforms 10-NN and A-kNN across a wide range of noise levels, while having comparable performance to the best kNN. See Appendix D for the comparison in MNIST CIFAR10 and MirFlickr.

label i. The ground-truth label is, by construction, always included in the candidate set. This procedure generates bags with high ambiguity and ensures that the bag generation process varies across clusters. To add noise we adopt the confusion process proposed by Lv et al. (2023). With probability ν, the ground truth label is replaced by a uniformly sampled random label. This corrupted label is then treated as the ground truth for bag generation—meaning it is guaranteed to appear in the bag, and other labels are added according to the same cluster-specific process described above.

For the real partially labeled datasets, label noise is introduced by removing the true label from the bags.

4.2 Specifications for PL A-kNN

We fix the hyperparameters of PL A-kNN as $c_1 = 0.5$ and $\delta = 0.1$ across all experiments, with $T = 400$ for vision datasets and $T = 50$ for the substantially smaller real-life datasets. Appendix D provides a detailed

description of the preprocessing used for all nearest-neighbor methods.

4.3 Results

Figures 1 and 2 include the performance of PL A-kNN with respect to the other models over an increasing noise ratio. Specifically, Figure 1 compares PL A-kNN with four state-of-the-art methods for CIFAR-10 and MSRCv2, while Figure 2 compares it with three kNN-based benchmarks under the same datasets. We only include results for one synthetic partially labeled vision dataset and one real dataset because they are representative of the trends observed across the remaining datasets. Complete results for all experimental settings are provided in Appendix D. All experiments use an 80/20 train-test split, and results are averaged over 100 independent runs.

Figure 1 shows that PL A-kNN outperforms state-of-the-art methods in general scenarios under a wide range of noise levels. Unlike competing methods, PL A-kNN consitently delivers strong performance, either matching or exceeding the alternatives in every setting. By contrast, the other methods tend to perform well only under specific conditions but degrade significantly in others. For example, the LWS method Wen et al. (2021) and PRODEN Lv et al. (2020) achieve high accuracy when noise is low but suffers substantial drops under high noise. Conversely, the CLPL method Cour et al. (2011) performs very poorly in low-noise scenarios but improves in the highly noisy settings of the real partial labels datasets.

Figure 2 shows that PL A-kNN provides a clear improvement over the standard 10-NN baseline commonly adopted in this setting (Hüllermeier and Beringer, 2006; Zhang and Yu, 2015). Notably, PL A-kNN also outperforms the extension for A-kNN from Balsubramani et al. (2019), achieving a comparable performance to the kNN under the best choice of number of neighbors.

The experimental results show that PL A-kNN can provide accurate classification rules across a wide range of general scenarios—from cases with a high ratio of noise to cases where the true label is always included in the bags.

5 Conclusion

In this paper we mathematically characterize the settings in which PLL is feasible. In addition, we present PL A-kNN, an adaptive nearest neighbors algorithm for PLL that is effective in general scenarios. We provide theoretical guarantees for PL A-kNN under weaker assumptions than previous work. In particular, we prove the Bayes consistency of PL A-kNN for any label-aligned scenario and show that no other algorithm can achieve consistency in strictly more general scenarios than PL A-kNN. Moreover, we also provide rates of convergence to the Bayes risk. Experimental results show that PL A-kNN can outperform state-of-the-art methods in general scenarios, matching the performance of the ideal nearest neighbors with the optimal number of neighbors.

Limitations

The proposed PL A-kNN method effectiveness depends on the availability of a preprocessing pipeline that yields a metric structure compatible with nearest-neighbor classification. When the features do not adequately capture label similarity, performance may degrade. Therefore, careful feature design and preprocessing remain important for practical deployment.

Acknowledgments

Funding for this work was provided by the Spanish Ministry of Science and Innovation under grants PID2022-137063NB-I00 and PID2022-137442NB-I00, funded by MCIN/AEI/10.13039/501100011033 and European Union "NextGenerationEU"/PRTR; the BCAM Severo Ochoa accreditation CEX2021-001142-S (MICIN/AEI/10.13039/501100011033); the Basque Government BERC 2022–2025 IT2109-26 and ELKARTEK programs; and the U.S. National Science Foundation under grant CCF-2217058.

References

Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007.

Akshay Balsubramani, Sanjoy Dasgupta, Shay Moran, et al. An adaptive nearest neighbor rule for classification. *Advances in Neural Information Processing Systems*, 32, 2019.

Vivien Cabannes, Alessandro Rudi, and Francis Bach. Structured prediction with partial labelling through the infimum loss. In *International Conference on Machine Learning*, pages 1230–1239. PMLR, 2020.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. *Advances in neural information processing systems*, 23, 2010.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. *Advances in Neural Information Processing Systems*, 27, 2014.

Jesús Cid-Sueiro. Proper losses for learning from partial labels. *Advances in neural information processing systems*, 25, 2012.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *Advances in neural information processing systems*, 33:10948–10960, 2020.

Juha Heinonen. *Lectures on analysis on metric spaces*. Springer Science & Business Media, 2001.

M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval (MIR'08)*, Vancouver, British Columbia, Canada, 2008. ACM.

Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

Yuheng Jia, Fuchao Yang, and Yongqiang Dong. Partial label learning with dissimilarity propagation guided candidate label shrinkage. *Advances in Neural Information Processing Systems*, 36, 2024.

Liping Liu and Thomas Dietterich. A conditional multinomial mixture model for superset label learning. *Advances in neural information processing systems*, 25, 2012.

Liping Liu and Thomas Dietterich. Learnability of the superset label learning problem. In *International conference on machine learning*, pages 1629–1637. PMLR, 2014.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International conference on machine learning*, pages 6500–6510. PMLR, 2020.

Jiaqi Lv, Biao Liu, Lei Feng, Ning Xu, Miao Xu, Bo An, Gang Niu, Xin Geng, and Masashi Sugiyama. On the robustness of average losses for partial-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2569–2583, 2023.

Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27 (6):1808–1829, 1999.

Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–559, 2008.

Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156, 2010.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Yingjie Tian, Xiaotong Yu, and Saiji Fu. Partial label learning: Taxonomy, analysis and outlook. *Neural Networks*, 161:708–734, 2023.

Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *International conference on machine learning*, pages 11091–11100. PMLR, 2021.

Ning Xu, Biao Liu, Jiaqi Lv, Congyu Qiao, and Xin Geng. Progressive purification for instance-dependent partial label learning. In *International Conference on Machine Learning*, pages 38551–38565. PMLR, 2023.

Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 708–715, 2013.

Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, pages 4048–4054, 2015.

Yu Zhou and Hong Gu. Geometric mean metric learning for partial label data. *Neurocomputing*, 275:394–402, 2018.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]

(b) Complete proofs of all theoretical results. [Yes]

(c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A On the generality of the label-aligned condition

In this appendix, we introduce the partial labels model of Liu and Dietterich (2014), which enables the recovery of the underlying distribution via empirical risk minimization, and six other partial label models from the literature (Cour et al., 2011; Cabannes et al., 2020; Feng et al., 2020; Wen et al., 2021; Lv et al., 2023) for which previous works have proposed Bayes consistent learning methods. We prove that six of these models correspond to specific cases of label-aligned bags (3). For the last partial labels model, we show that it considers scenarios that do not lead to label-aligned bags, and that there are label-aligned cases that do not satisfy the model's assumptions, proving that neither condition is more general than the other.

A.1 Partial labels model from Liu and Dietterich (2014)

The partial labels model from Liu and Dietterich (2014) assumes deterministic labels (i.e., probability distributions that assign all the probability mass of each instance to a single label), that the ground-truth label is always included in the bags, and that no other label can always co-occur with the ground-truth label in the bags of an instance (see the beginning of Section 3 and Subsection 3.1 in Liu and Dietterich (2014)). More precisely, the model in Liu and Dietterich (2014) considers scenarios that satisfy

$$
\begin{aligned}
&P(y|x) \in \{0,1\}, && \forall\, y \in \mathcal{Y},\, x \in \mathcal{X},\\
&y \notin s \implies P(s|y,x) = 0, && \forall\, y \in \mathcal{Y},\, x \in \mathcal{X},\\
&\text{for each } x \in \mathcal{X},\ \text{if } P(y=i|x) = 1,\ \text{then } P(\mathcal{S}_j|x,i) < 1, && \forall\, j \in \mathcal{Y} \setminus \{i\}.
\end{aligned}
$$

We now prove that the PLL scenarios that satisfy the previous conditions are label-aligned. Let P be a distribution that satisfies the previous assumptions. For a given instance x, let $i = \arg\max_{y\in\mathcal{Y}} P(y|x)$. Then, as labels are deterministic, we have that $P(i|x) = 1$, and since the ground-truth label is always included in the bags, $P(\mathcal{S}_i|x) = 1$. The co-occurrence condition implies that $P(\mathcal{S}_j|x) = P(\mathcal{S}_j|i,x) < 1, \forall j \neq i$. Therefore $i = \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$, and P has label-aligned bags.

A.2 Partial labels model from Cour et al. (2011)

The partial labels model from Cour et al. (2011) assumes the label-aligned condition (see Proposition 5 in Cour et al. (2011)). Moreover, it also assumes that the bag generation process satisfies the following domination condition:

$$
\begin{aligned}
&\forall x \in \mathcal{X},\ \text{let } i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x), \quad j \notin \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x),\\
&\text{and } s \in \mathcal{S} \text{ such that } i,j \notin s,\ \text{then}\\
&\quad P(s \cup \{i\}|x) > P(s \cup \{j\}|x)\,.
\end{aligned}
$$

A.3 Partial labels model from Cabannes et al. (2020)

The partial labels model from Cabannes et al. (2020) assumes that, for each instance, the intersection of all the possible bags of that instance is exactly the most probable label (see Definition 1 and 2, Proposition 1 in Cabannes et al. (2020)). More precisely, the model in Cabannes et al. (2020) considers scenarios that satisfy

$$
\begin{aligned}
&\forall x \in \mathcal{X}, \text{let } \ i = \arg\max_{y\in\mathcal{Y}} P(y|x), \ \text{ then}\\
&\bigcap_{s\in\mathcal{S}:P(s|x)>0} s = \{i\}.
\end{aligned}
$$

We now prove that the PLL scenarios that satisfy the previous conditions are label-aligned. Let P be a distribution that satisfies the previous assumptions. For a given instance x, let $i = \arg\max_{y\in\mathcal{Y}} P(y|x)$. Then, we have that $P(\mathcal{S}_i|x) = 1$ and $P(\mathcal{S}_j|x) < 1, \forall j \neq i$, since only the most probable label is always included in the bags. Therefore, $i = \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ and P has label-aligned bags.

A.4 Partial labels model from Feng et al. (2020)

The partial labels model from Feng et al. (2020) assumes that all bags that contain the ground-truth label are equally probable (see Equation 5 in Feng et al. (2020)). More precisely, the model in Feng et al. (2020) considers scenarios that satisfy

$$P(s|y,x) = \begin{cases} \dfrac{1}{2^{|\mathcal{Y}|-1}-1} & y \in s \\ 0 & y \notin s. \end{cases}$$

We now prove that all the PLL scenarios that satisfy the previous condition have label-aligned bags. Let P be a distribution that satisfies the previous assumption.

Let labels $i, j, k, k_2 \in \mathcal{Y}$ be such that $i \neq k$ and $j \neq k_2$. Then,

$$P(\mathcal{S}_i|k,x) = \sum_{s \in \mathcal{S}_i : k \in s} P(s|k,x) = \sum_{s \in \mathcal{S}_i : k \in s} \frac{1}{2^{|\mathcal{Y}|-1}-1} = |s \in \mathcal{S}_i : k \in s| \frac{1}{2^{|\mathcal{Y}|-1}-1} =$$
$$|s \in \mathcal{S}_j : k_2 \in s| \frac{1}{2^{|\mathcal{Y}|-1}-1} = \sum_{s \in \mathcal{S}_j : k_2 \in s} \frac{1}{2^{|\mathcal{Y}|-1}-1} = \sum_{s \in \mathcal{S}_j : k_2 \in s} P(s|k_2,x) = P(\mathcal{S}_j|k_2,x).$$

We can prove analogously that $P(\mathcal{S}_i|i,x) = P(\mathcal{S}_j|j,x)$.

We denote now $\alpha(x) = P(\mathcal{S}_i|i,x) - P(\mathcal{S}_j|k,x)$, where $j \neq k$. Due to the previous equalities, the value of alpha does not depend on the labels, and is clear that $\alpha(x) > 0 \ \forall x \in \mathcal{X}$, since $|\mathcal{S}_i| > |s \in \mathcal{S}_j : k \in s|$.

We then have that $P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x) = \sum_{y \in \mathcal{Y}} (P(\mathcal{S}_i|y,x) - P(\mathcal{S}_j|y,x)) P(y|x) = \alpha(x)(P(i|x) - P(j|x))$. Therefore, for any given instance, if $P(i|x) > P(j|x)$, then $P(\mathcal{S}_i|x) > P(\mathcal{S}_j|x)$, so P has label-aligned bags.

A.5 Partial labels model from Wen et al. (2021)

The partial labels model from Wen et al. (2021) assumes deterministic labels, that the ground-truth label i is always included in the bags, and that the other labels j in the bags are independently drawn given probabilities $q_{ij} < 1$ (see Subsection 3.3.1 and Equation 11 in Wen et al. (2021)). More precisely, the model in Wen et al. (2021) considers scenarios that satisfy

$$\begin{aligned} &P(y|x) \in \{0,1\}, && \forall y \in \mathcal{Y}, x \in \mathcal{X}, \\ &P(s|i,x) = \prod_{k \in s} q_{i,k} \prod_{j \notin s} (1 - q_{i,j}), && \forall s \in \mathcal{S}, \forall y \in \mathcal{Y}, x \in \mathcal{X}, \text{ where} \\ &q_{i,i} = 1, \quad \text{and} \quad q_{i,j} < 1, \quad i \neq j. \end{aligned}$$

We prove now that all distributions that satisfy the previous conditions are label-aligned. Let P be a distribution that satisfies the previous assumptions. For a given instance x, let $i \in \mathcal{Y}$ be the label with $P(i|x) = 1$. Then $P(\mathcal{S}_i|x) = P(\mathcal{S}_i|i,x) = 1$, since for any bag s with $i \notin s$ we have that $P(s|i,x) = 0$. Moreover, $P(\{i\}|x) = P(\{i\}|i,x) > 0$, since $q_{i,j} < 1 \ \forall i \neq j$, and therefore $P(\mathcal{S}_j|x) < 1$ for any $j \neq i$. We conclude then that P is label-aligned.

Note: Wen et al. (2021) states that their model is intended for stochastic labels, in which case the bags are not always label-aligned. However, if labels are stochastic, their assumptions on the bag generation process are incompatible with the Bayes consistency of any algorithm. In fact, the proofs of their theorems assume implicitly deterministic labels. The following provides a short counterexample for the stochastic case, where we show that no algorithm can be Bayes consistent under the assumptions of the model.

Let us consider a binary label space and an instance space consisting of a single point. Let two partial label probability distributions be given by

$$P(1) = 2/3, \quad P(2) = 1/3, \quad q_{1,1} = 1, \quad q_{1,2} = 2/3, \quad q_{2,2} = 1, \quad q_{2,1} = 0$$
$$\hat{P}(1) = 1/3, \quad \hat{P}(2) = 2/3, \quad \hat{q}_{1,1} = 1, \quad \hat{q}_{1,2} = 1/3, \quad \hat{q}_{2,2} = 1, \quad \hat{q}_{2,1} = 1/2.$$

Both scenarios abide the conditions from the partial labels model in Wen et al. (2021) and generate the same distribution of bags

$$P(\{1\}) = 2/9, \quad P(\{2\}) = 3/9, \quad P(\{1,2\}) = 4/9,$$

while having the two different labels as Bayes rules. Therefore, it is not possible for an algorithm to be Bayes consistent for all scenarios considered in the model when the labels are stochastic.

A.6 Noiseless partial labels model from Lv et al. (2023)

The noiseless partial labels model from Lv et al. (2023) assumes deterministic labels, that the ground truth label is always included in the bags, that the bag generation process is the same for all instances, and that the frequencies within the bags of labels other than the ground-truth label are strictly less than one (see Section 4.2 and Theorem 1 in Lv et al. (2023)). More precisely, the noiseless model in Lv et al. (2023) considers scenarios that satisfy

$$\begin{aligned}
&P(y|x) \in \{0,1\} && \forall y \in \mathcal{Y}, \forall x \in \mathcal{X} \\
&y \notin s \implies P(s|y,x) = 0 && \forall y \in \mathcal{Y}, \forall x \in \mathcal{X} \\
&P(s|y,x_1) = P(s|y,x_2) && \forall y \in \mathcal{Y}, \forall x_1, x_2 \in \mathcal{X} \\
&P(\mathcal{S}_j|i,x) < 1 && i \neq j.
\end{aligned}$$

We now prove that all PLL scenarios that satisfy the previous assumptions have label-aligned bags . Let P be a distribution that satisfies the previous assumptions. For a given instance, let $i \in \mathcal{Y}$ be the label with $P(i|x) = 1$. We then have that $P(\mathcal{S}_i|x) = P(\mathcal{S}_i|i,x) = 1$, since for any bag s with $i \notin s$ we have that $P(s|i,x) = 0$. On the other hand $P(\mathcal{S}_j|x) = P(\mathcal{S}_j|i,x) < 1$ for any $j \neq i$. We conclude then that P has label aligned bags.

A.7 Noisy partial labels model from Lv et al. (2023)

The noisy partial labels model from Lv et al. (2023) assumes deterministic labels, that the bags generation process does not vary across instances, and that the most probable label is the one most likely to be selected when randomly choosing a label from the bags(see Section 4.2 and Theorem 5 in Lv et al. (2023)). More precisely, the noisy model in Lv et al. (2023) considers scenarios that satisfy

$$\begin{aligned}
&P(y|x) \in \{0,1\}, \quad \forall y \in \mathcal{Y}, \forall x \in \mathcal{X}, \\
&P(s|y,x_1) = P(s|y,x_2), \quad \forall y \in \mathcal{Y}, \forall x_1, x_2 \in \mathcal{X}, \\
&\sum_{s \in \mathcal{S}_i} \frac{1}{|s|} P(s|i,x) > \sum_{s \in \mathcal{S}_j} \frac{1}{|s|} P(s|i,x) \quad \forall i \in \mathcal{Y}, \forall j \neq i, \forall x \in \mathcal{X}.
\end{aligned}$$

In this case, there are PLL scenarios that satisfy the model's assumptions but do not have label-aligned bags, and there are label-aligned cases that do not satisfy the model's assumptions. For example, in a partial labels problem with three labels and only one instance, let the distribution over the labels of P_1 be

$$P_1(1) = 1 \quad P_1(2) = 0 \quad P_1(3) = 0$$

and the bag generation process when 1 is the true label be

$$\begin{aligned}
&P_1(\{1\}|1) = 0.1 \quad P_1(\{2\}|1) = 0 \quad P_1(\{3\}|1) = 0.4 \\
&P_1(\{1,2\}|1) = 0.5 \quad P_1(\{2,3\}|1) = 0 \quad P_1(\{1,3\}|1) = 0 \\
&P_1(\{1,2,3\}|1) = 0.
\end{aligned}$$

We then have that the distribution over the bags is $P_1(s) = P_1(s|1)$, so $P_1(\mathcal{S}_1) = 0.6$, $P_1(\mathcal{S}_2) = 0.5$, and $P_1(\mathcal{S}_3) = 0.4$, and therefore P_1 has label-aligned bags . However,

$$\sum_{s \in \mathcal{S}_1} \frac{1}{|s|} P_1(s|1,x) = 0.35 < 0.4 = \sum_{s \in \mathcal{S}_3} \frac{1}{|s|} P_1(s|1,x)$$

and therefore P_1 does not satisfy the previous assumptions.

On the other hand, let P_2 be the underlying distribution of other partial labels problem with three labels and only one instance, with distribution over the labels

$$P_2(1) = 0 \;\; P_2(2) = 0 \;\; P_2(3) = 1,$$

and bag generation process when 3 is the ground-truth label

$$\begin{aligned} &P_2(\{1\}|3) = 0.1 \quad P_2(\{2\}|3) = 0 \quad P_2(\{3\}|3) = 0.4 \\ &P_2(\{1,2\}|3) = 0.5 \;\; P_2(\{2,3\}|3) = 0 \;\; P_2(\{1,3\}|3) = 0 \\ &P_2(\{1,2,3\}|3) = 0. \end{aligned}$$

We then have that the distribution over the bags is $P_2(s) = P_2(s|3)$, so $P_2(\mathcal{S}_1) = 0.6$, $P_2(\mathcal{S}_2) = 0.5$, and $P_2(\mathcal{S}_3) = 0.4$ and therefore P_2 does not have label-aligned bags. However,

$$\sum_{s\in\mathcal{S}_1} \frac{1}{|s|} P_2(s|3,x) = 0.35 \;\; \sum_{s\in\mathcal{S}_1} \frac{1}{|s|} P_2(s|3,x) = 0.25 \;\; \sum_{s\in\mathcal{S}_3} \frac{1}{|s|} P_2(s|3,x) = 0.4,$$

so P_2 satisfies the assumptions of the model.

B Extended Theoretical Analysis of PL A-kNN

B.1 Notation

Let $\{(x_l, s_l)\}_{l=1}^n$ be the set of training examples. For any subset $G \subseteq \mathcal{X}$, the empirical count and mass are taken as:

$$\begin{aligned} \#_n(G) &= |\{l : x_l \in G\}| \\ P_n(G) &= \frac{\#_n(G)}{n}. \end{aligned}$$

In addition, for any subset $G \subseteq \mathcal{X}$ with non zero empirical mass, the empirical frequencies of the labels in the bags of G are by definition:

$$P_n(\mathcal{S}_y|G) = \frac{\sum_{l=1}^n \mathbb{I}(y \in s_l)\mathbb{I}(x_l \in G)}{\#_n(G)} \qquad \forall y \in \mathcal{Y}.$$

B.2 Fundamental differences between PL A-kNN and A-kNN

PL A-kNN is inspired by the adaptive k-nearest neighbor (A-kNN) method for binary classification from Balsubramani et al. (2019). PLL requires to consider multiclass settings, and the proposed PL A-kNN differs fundamentally from the multiclass supervised classification extension suggested in Balsubramani et al. (2019). Such an extension increases the neighborhood until one label satisfies

$$P_n(y \,|\, B_k(x)) - \frac{1}{|\mathcal{Y}|} \geq \Delta(n,k,\delta),$$

where $B_k(x)$ represents the ball with exactly the k nearest neighbors of x, and then predicts that label. In other words, the A-kNN extension increases the neighborhood until the empirically most frequent label has higher frequency than threshold Δ, thereby only considering the frequency of that label. This criterion naturally extends to the partial-labels case as follows:

$$P_n(\mathcal{S}_y \,|\, B_k(x)) - \frac{1}{|\mathcal{Y}|} \geq \Delta(n,k,\delta),$$

In contrast, PL A-kNN progressively eliminates labels i whose frequency deviates from the maximum frequency by more than Δ, that is,

$$\max_{y \in \hat{s}} P_n(\mathcal{S}_y \mid B_k(x)) - P_n(\mathcal{S}_i \mid B_k(x)) \geq \Delta(n, k, \delta),$$

gradually narrowing the candidate set while enlarging the neighborhood. This margin-based elimination considers all of the label frequencies, not just the largest one.

To illustrate the differences between the two methods, consider two multiclass classification problems (which can be viewed as trivial partial-label problems where each bag contains only the ground-truth label) with a single instance, three labels, and the following label distributions: in the first case, $Q_1 = (0.50, 0.49, 0.01)$; in the second case, $Q_2 = (0.40, 0.30, 0.30)$. These vectors represent both the label distributions and the frequencies of the labels within the bags. For A-kNN, which only checks frequency of the empyrically most frequent label, the second case may require more neighbors to satisfy the threshold, even though the first case is harder due to the nearly tied most probable labels. PL A-kNN, by considering frequency differences, naturally requires more neighbors in the first case and fewer in the second case. This shows that PL A-kNN increases the neighborhood size appropriately for harder cases, whereas A-kNN can use too many or too few neighbors, because it only considers the frequency of the most frequent label and disregards the rest of the information.

B.3 Finite sample convergence rates

To establish finite-sample convergence rates, the data must satisfy certain smoothness conditions. Therefore, we assume two smoothness conditions over the underlying probability distribution $P \in \Delta(\mathcal{X} \times \mathcal{Y} \times \mathcal{S})$ for the next result.

First, we adapt the Tsybakov-margin for binary distributions (Audibert and Tsybakov, 2007; Mammen and Tsybakov, 1999; Tsybakov, 2004) to the frequencies of the labels over the bags. For any $\beta > 0$, we say P satisfies the β-margin condition if there exists a constant $C_2 > 0$ such that

$$P(\{x \in \mathcal{X} : \forall i \in \arg\max_{y \in \mathcal{Y}} P(\mathcal{S}_y|x) \;\; \exists j \notin \arg\max_{y \in \mathcal{Y}} P(\mathcal{S}_y|x) \;\; P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x) \leq t\}) \leq C_2 \cdot t^\beta \qquad \forall t \geq 0.$$

We essentially require the probability mass of the instances with a frequency margin smaller than t to be gracefully bounded by a power function of t. Larger values of β imply smoother distributions.

For the second smoothness condition we adapt to PLL a generalization of the usual α-Holder condition (Chaudhuri and Dasgupta, 2014), which is a common assumption over P for nonparametric estimators. For $\alpha > 0, L > 0$ we say that P is (α, L)-smooth in the finite-dimensional normed space $(\mathcal{X}, P)$ if for all $x \in \mathcal{X}$

$$|P(\mathcal{S}_y|B(x,r)) - P(\mathcal{S}_y|x)| \leq LP(B(x,r))^\alpha \qquad \forall y \in \mathcal{Y}, \forall r \geq 0.$$

Thus, we require the drift of the frequencies of the labels over the bags in balls to be bounded by α-exponential values of the mass of the ball. Larger values of α indicate that P behaves smoother.

The following theorem provides rates of convergence that depend explicitly on the number of samples.

Theorem B.1. (Explicit rates of convergence) Let h be the classifier from PL A-kNN Algorithm 1 with maximum number of iterations T and confidence parameter δ satisfying (4). Let the underlying probability distribution satisfy the label-aligned condition (3), be (α, L)-smooth, and satisfy the β-margin condition for $\alpha > 0$ and $\beta > 0$. Then there exists a constant C_3 such that

$$R(h) - R^* \leq \delta + C_3 \left(\frac{1}{n} \max\Big\{ \log(n), \log(|\mathcal{Y}|/\delta) \Big\} \right)^{\frac{\beta\alpha}{2\alpha+1}}$$

holds with probability at least $1 - \delta$.

***Proof**: See Appendix C.*

As expected, larger values of α and β provide faster rates of convergence.

C Theorems, Lemmas, and Proofs

C.1 Notation and definitions

Let $\{(x_l, s_l)\}_{l=1}^n$ be the set of training examples. For any subset $G \subseteq \mathcal{X}$, the empirical count and mass are taken as:

$$\#_n(G) = |\{l : x_l \in G\}|$$
$$P_n(G) = \frac{\#_n(G)}{n}.$$

In addition, for any subset $G \subseteq \mathcal{X}$ with non zero empirical mass, the empirical frequencies of the labels in the bags of G are by definition:

$$P_n(\mathcal{S}_y|G) = \frac{\sum_{l=1}^n \mathbb{I}(y \in s_l)\mathbb{I}(x_l \in G)}{\#_n(G)} \qquad \forall y \in \mathcal{Y}.$$

Definition C.1. The *support* of the distribution P, denoted by $\text{supp}(P)$, is the set

$$\text{supp}(P) = \{x \in \mathcal{X} \mid \forall\, r > 0,\ P(B(x, r)) > 0\}, \tag{8}$$

where $B(x, r)$ is the ball of radius r centered at x.

We proceed with a smoothness condition that holds for finite dimensional normed spaces.

Definition C.2. (Lebesgue differentiation condition) Let $(\mathcal{X}, d, P)$ be a metric measure space. We say that $(\mathcal{X}, d, P)$ satisfies the Lebesgue differentiation condition if for any bounded measurable $f : \mathcal{X} \longrightarrow \mathbb{R}$ and for almost all (P-a.e.) $x \in \mathcal{X}$, we have

$$\lim_{r \to 0} \frac{1}{P(B(x, r))} \int_{B(x,r)} f \, dP = f(x). \tag{9}$$

C.2 Proof of Theorem 2.4

Proof. Given an instance $x \in \mathcal{X}$, let $Q_1, Q_2 \in \Delta(\mathcal{Y})$ be two distributions over the labels such that $\sum_{y \in \mathcal{Y}} P(s|y, x)Q_1(y) = \sum_{y \in \mathcal{Y}} P(s|y, x)Q_2(y)\ \ \forall s \in \mathcal{S}$. Since $P(s|y, x)$ is label-aligned, we have that

$$\arg\max_{y \in \mathcal{Y}} Q_1(y) = \arg\max_{y \in \mathcal{Y}} \sum_{s \in \mathcal{S}_y} \sum_{i \in \mathcal{Y}} P(s|i, x)Q_1(i) =$$
$$\arg\max_{y \in \mathcal{Y}} \sum_{s \in \mathcal{S}_y} \sum_{i \in \mathcal{Y}} P(s|i, x)Q_2(i) = \arg\max_{y \in \mathcal{Y}} Q_2(y)$$

and therefore the proof is concluded. □

C.3 Proof of Theorem 3.2

Before providing the proof of the theorem we state and prove some technical lemmas. We first state two results from previous work that are needed for our main result

Lemma C.3. (Chaudhuri and Dasgupta (2010) Lemma 7) There is a universal constant c_0 such that the following holds. Let $\mathcal{B}$ be any class of measurable subsets of $\mathcal{X}$ of VC dimension d_0. Pick any $0 < \delta < 1$. Then with probability at least $1 - \delta^2/2$ over the choice of $x_1, x_2 \ldots, x_n$, for all $B \in \mathcal{B}$ and for any integer k, we have

$$P(B) \geq \frac{k}{n} + \frac{c_0}{n} \max\left(k, d_0 \log \frac{n}{\delta}\right) \quad \Rightarrow \quad P_n(B) \geq \frac{k}{n}.$$

Theorem C.4. (Balsubramani et al. (2019) Theorem 8) Let P be a probability distribution over $\mathcal{X}$, and let $\mathcal{A}, \mathcal{B}$ be two families of measurable subsets of $\mathcal{X}$ such that $\text{VC}(\mathcal{A}), \text{VC}(\mathcal{B}) \leq d_0$. Let $n \in \mathbb{N}$, and let $x_1, \ldots, x_n$ be n i.i.d. samples from P. Then, the following event occurs with probability at least $1 - \delta$:

$$\forall A \in \mathcal{A},\ \forall B \in \mathcal{B} : \quad |P(A|B) - P_n(A|B)| \leq \sqrt{\frac{k_0}{\#_n(B)}},$$

where $k_0 = 1000\left(d_0 \log(8n) + \log\left(\frac{4}{\delta}\right)\right)$.

We now state and prove two technical lemmas that are needed for the main proof of the theorem.

Lemma C.5. Let P be a probability distribution over $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$. There is a universal constant $c_1 > 0$ such that, for each $x \in \mathcal{X}$, the following events occur with probability at least $1 - \delta^2/2$:

$$|P(\mathcal{S}_y|B(x,r)) - P_n(\mathcal{S}_y|B(x,r))| \leq \frac{c_1}{2}\sqrt{\frac{\log(n) + \log(|\mathcal{Y}|/\delta)}{\#_n(B(x,r))}} \qquad \forall r > 0, \forall y \in \mathcal{Y}. \tag{10}$$

Let d_0 be the VC dimension of the set of balls in $\mathcal{X}$. Then, we have that the following events occur with probability at least $1 - \delta^2/2$:

$$|P(\mathcal{S}_y|B(x,r)) - P_n(\mathcal{S}_y|B(x,r))| \leq \frac{c_1}{2}\sqrt{\frac{d_0 \log(n) + \log(|\mathcal{Y}|/\delta)}{\#_n(B(x,r))}} \qquad \forall r > 0, \forall y \in \mathcal{Y}, \forall x \in \mathcal{X}. \tag{11}$$

Proof. Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\mathcal{B}_x$ be the set of all balls that are centered on x, which has VC dimension 1. We define the two following sets of sets:

$$\mathbf{A}_x^y = \{B \times \mathcal{S}_y : B \in \mathcal{B}_x\}$$
$$\mathbf{B}_x = \{B \times 2^{\mathcal{Y}} : B \in \mathcal{B}_x\}.$$

It is easy to see then than $VC(\mathbf{A}_x^y) = 1$ and $VC(\mathbf{B}_x) = 1$, as both $\mathbf{A}_x^y$ and $\mathbf{B}_x$ are obtained by taking Cartesian products of sets in $\mathcal{B}_x$ (which has VC dimension 1) with fixed sets.

Therefore, taking $c_1 > 2\sqrt{8000\log(8)}$ and by Theorem C.4 we have proven that with probability at least $1 - \delta^2/(2|\mathcal{Y}|)$.

$$|P(\mathcal{S}_y|B(x,r)) - P_n(\mathcal{S}_y|B(x,r))| \leq \frac{c_1}{2}\sqrt{\frac{\log(n) + \log(|\mathcal{Y}|/\delta)}{\#_n(B(x,r))}} \qquad \forall r > 0.$$

Then, the first part of the lemma in (10) follows by applying the union bound for each label.

Suppose now that d_0 is the VC dimension of the set of balls $\mathcal{B}$ in $\mathcal{X}$ and let $y \in \mathcal{Y}$. We define the following sets of sets

$$\mathbf{A}^y = \{B \times \mathcal{S}_y : B \in \mathcal{B}\}$$
$$\mathbf{B} = \{B \times 2^{\mathcal{Y}} : B \in \mathcal{B}\}.$$

We then have, by following the same reasoning as before, that $VC(\mathbf{A}) = d_0$ and $VC(\mathbf{B}) = d_0$.

Therefore, taking $c_1 > 2\sqrt{4000(d_0+1)\log(8)}$ and proceeding analogously to the first part of this proof we have proven the second part of the lemma. □

Lemma C.6. The following set of points

$$\left\{x \in \mathcal{X} : \ \mathrm{adv}_{\mathrm{A}}(x) = 0\right\}$$

has zero P-measure.

Proof.

The instance space $(\mathcal{X}, d)$ is a finite dimensional normed space. Therefore, the Lebesgue differentiation condition C.2 holds (Heinonen, 2001). Let $\mathcal{X}'$ be the subset of the support of distribution P (8) for which condition (9) is satisfied for the $|\mathcal{Y}|$ functions $P(\mathcal{S}_y|x)$. We have $P(\mathcal{X}') = 1$, since the Lebesgue differentiation condition holds and the support of P has measure 1. If we see that all the elements in $\mathcal{X}'$ have positive advantage, then the proof is concluded.

Let x be an instance in $\mathcal{X}'$. If $\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x) = \mathcal{Y}$ then by definition $\text{adv}_\text{A}(x) = 1$. For other cases, since the point x satisfies (9) for all $P(\mathcal{S}_y|x)$ functions we have that $\forall i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ and $\forall j \in \mathcal{Y} \setminus \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ exists $r_{i,j} > 0$ such that

$$P(\mathcal{S}_i|B(x,r)) - P(\mathcal{S}_j|B(x,r)) \geq \frac{P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x)}{2} \quad 0 < r < r_{i,j}.$$

Therefore, the result is obtained, because x is (p,γ)-salient for $p = \min\{A, P(B(x, \min r_{i,j}))\} > 0$ and $\gamma = min_{j\notin \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)} \frac{P(\mathcal{S}_i|x)-P(\mathcal{S}_j|x)}{2}$, having positive advantage in the region $B(x, r_A(x))$. □

Proof of Theorem 3.2. Let c_0 and c_1 be the constants of Lemma C.3 and Lemma C.5. We define $c_2 = \max(c_1, 1/4)\sqrt{1+c_0}$ and take $C = 16c_2^2$.

Let $x \in \mathcal{X}$ such that $\text{adv}_\text{A}(x) > 0$, and $\mathcal{B}$ the set of all balls that are centered on x. Following Lemma C.3 and Lemma C.5 we have that with probability at least $1-\delta^2$ the following two properties hold for all $B \in \mathcal{B}$:

- For any integer k we have $\#_n(B) \geq k$ whenever $nP(B) \geq k + c_0 \max\{k, \log(n/\delta)\}$.
- $|P_n(\mathcal{S}_y|B) - P(\mathcal{S}_y|B))| \leq \frac{1}{2}\Delta(n, \#_n(B), \delta), \quad \forall y \in \mathcal{Y}, \quad$ and therefore
  $|(P_n(\mathcal{S}_i|B) - P_n(\mathcal{S}_j|B)) - (P(\mathcal{S}_i|B) - P(\mathcal{S}_j|B))| \leq \Delta(n, \#_n(B), \delta), \forall i,j \in \mathcal{Y}$.

Assume henceforth that the above two conditions hold. If $\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x) = \mathcal{Y}$, we also have that $\arg\max_{y\in\mathcal{Y}} P(y|x) = \mathcal{Y}$, since the bags are label-aligned, and therefore $h(x) = h^*(x)$. Otherwise, let $y_0 \notin \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$. If we show that $h(x) \neq y_0$, the proof is concluded, as we show that $h(x)$ is not any of the suboptimal labels, which is equivalent to showing that $h(x)$ is one of the most frequent labels in the bags of x.

By the definition of advantage, point x is (p,γ)-salient for some $p\gamma > 0$ with $\text{adv}_\text{A}(x) = p\gamma^2$.

The criterion (5) in the theorem statement implies that:

$$\gamma \geq 2c_2\sqrt{\frac{\log(n) + \log(|\mathcal{Y}|/\delta)}{np}}. \tag{12}$$

Let $k = \frac{np}{1+c_0}$. Then, $k \leq \frac{n\cdot A}{1+c_0} \leq T$, as T satisfies 4. By (12) we have that $np \geq 4c_2^2 \log(n/\delta)$ and thus $k \geq \log(n/\delta)$. As a result $np = (1+c_0)k \geq k + c_0 \max\{k, \log(n/\delta)\}$, and by the first property, the ball $B = B(x, r_p(x))$ has $\#_n(B) \geq k$. Let $B_k(x)$ represent the ball with exactly the k nearest neighbors of x

By the second property, we have that:

$$\begin{aligned}
&P_n(\mathcal{S}_i|B_k(x)) - P_n(\mathcal{S}_{y_0}|B_k(x)) \geq P(\mathcal{S}_i|B_k(x)) - P(\mathcal{S}_{y_0}|B_k(x)) - \Delta(n,k,\delta) \geq \gamma - \Delta(n,k,\delta) \geq \\
&2c_2\sqrt{\frac{\log(|\mathcal{Y}|n/\delta)}{np}} - c_1\sqrt{\frac{\log(|\mathcal{Y}|n/\delta)}{k}} \geq 2c_1\sqrt{\frac{\log(|\mathcal{Y}|n/\delta)}{k}} - c_1\sqrt{\frac{\log(|\mathcal{Y}|n/\delta)}{k}} \geq \\
&c_1\sqrt{\frac{\log(|\mathcal{Y}|n/\delta)}{k}} = \Delta(n,k,\delta) \qquad \forall i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x).
\end{aligned}$$

Therefore, we have that the difference between the empirical frequencies of the elements in $\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ and y_0 in the bags of $B_k(x)$ is at least as big as the threshold Δ. If the algorithm has not eliminated all the labels in $\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ before the kth iteration, it will eliminate y_0 from the set of possible labels $\hat{s}$, so that the result is obtained.

At the same time, for any ball $B' = B_{k'}(x)$ with $k' < k$ we have

$$\begin{aligned}
&P_n(\mathcal{S}_j|B') - P_n(\mathcal{S}_i|B') \leq P(\mathcal{S}_j|B') - P(\mathcal{S}_i|B') + \Delta(n, \#_n(B'), \delta) < \Delta(n, \#_n(B'), \delta) \\
&\forall i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x), \forall j \in \mathcal{Y} \setminus \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x).
\end{aligned}$$

Therefore, although labels from $\arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ can eliminate other labels of the same set before the kth iteration, we can always ensure that one of them will remain in the set of possible labels, since elements in $\mathcal{Y} \setminus \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x)$ cannot eliminate then.

The second part of the theorem is obtained using Lemma C.6. □

C.4 Proof of Theorem 3.3

Proof of Theorem 3.3. This proof is strictly analogous to the proof of Theorem 3.2. The only difference is that the following two properties

- For any integer k we have $\#_n(B) \geq k$ whenever $nP(B) \geq k + c_0 \max\{k, \log(n/\delta)\}$.
- $|P_n(\mathcal{S}_y|B) - P(\mathcal{S}_y|B))| \leq \frac{1}{2}\Delta(n, \#_n(B), \delta), \quad \forall y \in \mathcal{Y}, \quad$ and therefore
  $|(P_n(\mathcal{S}_i|B) - P_n(\mathcal{S}_j|B)) - (P(\mathcal{S}_i|B) - P(\mathcal{S}_j|B))| \leq \Delta(n, \#_n(B), \delta), \forall i, j \in \mathcal{Y}$.

hold with probability $1 - \delta^2$ for all the balls in $\mathcal{X}$ □

C.5 Proof of Theorem 3.4

Before providing the proof of the theorem we state and prove a technical lemma.

Lemma C.7. Let C be the constant from Theorem 3.2. Let h be the classifier from PL A-kNN algorithm 1 with maximum number of iterations T and confidence parameter δ satisfying 4. If the underlying distribution satisfies the label-aligned condition (3) and $a > 0$ satisfies

$$n \geq \frac{C}{a} \max\{\log(1/a), \log(|\mathcal{Y}|/\delta)\},$$

we have

$$R(h) - R^* \leq \delta + P(\mathrm{adv}_{\mathrm{A}}(x) \leq a)$$

with probability at least $1 - \delta$.

Proof. This proof is a simple application of Markov's inequality.

From Theorem 3.2 we have that for each $x \in \mathcal{X}$ such that $\mathrm{adv}_{\mathrm{A}}(x) > a$, $Pr_n(h(x) \neq h^*(x)) \leq \delta^2$, where Pr_n denotes probability over the choice of training points. Thus, for $\mathcal{X} \sim P$

$$\mathbb{E}_n \mathbb{E}_{\mathcal{X}} \mathbb{I}\{h(x) \neq h^*(x) | \, \mathrm{adv}_{\mathrm{A}}(x) > a\} \leq \delta^2,$$

and by Markov's inequality:

$$Pr_n[P(h(x) \neq h^*(x) | \, \mathrm{adv}_{\mathrm{A}}(x) > a) \geq \delta] \leq \delta.$$

Thus, with probability $1 - \delta$ over the training examples:

$$P(h(x) \neq h^*(x) | \, \mathrm{adv}_{\mathrm{A}}(x) > a) \leq \delta,$$

and we can conclude then that, with probability $1 - \delta$ over the training examples,

$$R(h) - R^* \leq P(\mathrm{adv}_{\mathrm{A}}(x) \leq a) + P(h(x) \neq h^*(x) | \, \mathrm{adv}_{\mathrm{A}}(x) > a) \leq \delta + P(\mathrm{adv}_{\mathrm{A}}(x) \leq a).$$

□

Proof of Theorem 3.4. In the following, we show

$$n \geq \frac{C}{a_n} \max\{\log(1/a_n), \log(|\mathcal{Y}|/\delta)\},$$

and such an inequality leads to the result by using Lemma C.7.

We have that $a_n \geq C\frac{2\log(n)}{n}$ from its definition. If we divide both sides by $\log(1/a_n)$, and since $a_n \geq C\frac{2\log(n)}{n}$ implies that $n > \frac{1}{a_n}$, we have that $\frac{a_n}{\log(1/a_n)} \geq \frac{C}{n}$. Therefore, $n \geq C\frac{\log(1/a_n)}{a_n}$.

We also have that $a_n \geq C\frac{\log(|\mathcal{Y}|/\delta)}{n}$ from its definition, thus we have that $n \geq C\frac{\log(|\mathcal{Y}|/\delta)}{a_n}$.

Computing the maximum of the previous two inequalities we obtain

$$n \geq \frac{C}{a_n}\max\{\log(1/a_n), \log(|\mathcal{Y}|/\delta)\},$$

and the proof is concluded. □

C.6 Proof of Theorem B.1

Proof. Let x be an instance from $\mathcal{X}$. Choose $t_0 > 0$ such that:

$$P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x) > t_0 \qquad \forall i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x), \forall j \notin \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x).$$

Since P is (α, L)-smooth, choosing $p_0 = (\frac{t_0}{4L})^{1/\alpha}$ we have :

$$P(\mathcal{S}_y|(B(x, r_p(x))) \geq P(\mathcal{S}_y|x) - Lp^\alpha \qquad \forall p \leq p_0, \forall y \in \mathcal{Y}$$
$$P(\mathcal{S}_y|(B(x, r_p(x))) \leq P(\mathcal{S}_y|x) + Lp^\alpha \qquad \forall p \leq p_0, \forall y \in \mathcal{Y}$$

and therefore :

$$P(\mathcal{S}_i|(B(x, r_p(x))) - P(\mathcal{S}_j|(B(x, r_p(x))) \geq P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x) - 2Lp^\alpha \geq t_0/2$$
$$\forall p \leq p_0, i \in \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x), j \notin \arg\max_{y\in\mathcal{Y}} P(\mathcal{S}_y|x).$$

It is straightforward then that $\text{adv}_\text{A}(x) > \min\{A, p_0\} \cdot t_0^2/4 = \min\{At_0^2/4, \frac{t_0^{2+1/\alpha}}{4(4L)^{1/\alpha}}\}$.

Since P satisfies the β-margin condition, we have that

$$P(\{x \in \mathcal{X} : \forall i \in \arg\max_{y\in Y} P(\mathcal{S}_y|x) \;\; \exists j \notin \arg\max_{y\in Y} P(\mathcal{S}_y|x) \;\; st : P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x) \leq t\}) \leq C_2 \cdot t^\beta$$

and consequently

$$P(\{x \in \mathcal{X} : \forall i \in \arg\max_{y\in Y} P(\mathcal{S}_y|x) \;\; \forall j \notin \arg\max_{y\in Y} P(\mathcal{S}_y|x) \;\; st : P(\mathcal{S}_i|x) - P(\mathcal{S}_j|x) > t\}) \geq 1 - C_2 t^\beta.$$

Therefore, using the implication about the advantage derived at the beginning of the proof we have that $P(\text{adv}_\text{A}(x) > \min\{\frac{At_0^2}{4}, \frac{t_0^{2+1/\alpha}}{4(4L)^{1/\alpha}}\}) \geq 1 - C_2 t^\beta$, so $P(\text{adv}_\text{A}(x) \leq \min\{\frac{At_0^2}{4}, \frac{t_0^{2+1/\alpha}}{4(4L)^{1/\alpha}}\}) \leq C_2 t^\beta$ for any $t > 0$, which we can rewrite as $P(\text{adv}_\text{A}(x) \leq a) \leq C_2' \cdot (a^{\frac{\alpha}{2\alpha+1}})^\beta$ for any $a \in [0, 1]$, where $C_2' = C_2\ ((\frac{4}{A})^\alpha 4L)^{\frac{1}{2\alpha+1}})^\beta$.

Therefore, we have from Theorem 3.4 that with probability $1 - \delta$:

$$R(h) - R^* \leq \delta + P(\text{adv}_\text{A}(x) \leq a_n) \leq \delta + C_3 \left(\frac{1}{n}\max\Big\{\log(n), \log(|\mathcal{Y}|/\delta)\Big\}\right)^{\beta(\frac{\alpha}{2\alpha+1})}$$

where $C_3 = (2C)^{\beta(\frac{\alpha}{2\alpha+1})} \cdot C_2'$, which concludes the proof. □

C.7 Proof of Theorem 3.1

Proof of Theorem 3.1. Given the sequence of confidence parameters $\{\delta_n\}_{n=1}^\infty$, we define a sequence of advantage values as in Theorem 3.4:

$$a_n = \frac{C}{n}\max\big\{2\log(n), \log(|\mathcal{Y}|/\delta_n)\big\}.$$

The conditions established over the sequence $\{\delta_n\}_{n=1}^{\infty}$ imply that $a_n \to 0$.

If $\varepsilon > 0$, by the conditions established for $\{\delta_n\}_{n=1}^{\infty}$ we can choose an N such that $\sum_{n=N}^{\infty} \delta_n < \varepsilon$. Let $\{(x_n, s_n)\}_{n=1}^{\infty}$ denote a random sequence of training examples. We have by Theorem 3.4 that:

$$\begin{aligned} &P(\exists n \geq N : R(h_n) - R^* \geq \delta + P(\mathrm{adv}_{\mathrm{A}}(x) \leq a_n)) \\ &\leq \sum_{n=N}^{\infty} P(R(h_n) - R^* \geq \delta + P(\mathrm{adv}_{\mathrm{A}}(x) \leq a_n)) \\ &\leq \sum_{n=N}^{\infty} \delta_n < \varepsilon. \end{aligned}$$

Thus, with probability at least $1 - \varepsilon$ over the training sequence $\{(x_n, s_n)\}_{n=1}^{\infty}$ we have that $\forall n \geq N$

$$R(h_n) - R^* \leq \delta_n + P(\mathrm{adv}_{\mathrm{A}}(x) \leq a_n)$$

Therefore, since $a_n \to 0$, and $\lim_{a_n \to 0} P(\mathrm{adv}_{\mathrm{A}}(x) \leq a_n) = 0$ (see Lemma C.6) we have that $R(h_n) \to R^*$ almost surely.

We now proof the second part of the theorem, which states that no other algorithm can achieve Bayes consistency under more general scenarios of PLL than PL A-kNN. We proof that statement showing that if an algorithm is Bayes consistent for a case where PL A-kNN fails, then that algorithm fails in a case with label-aligned bags, where PL A-kNN is Bayes consistent.

Let P_1 be a distribution in $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$ for which PL A-kNN is not Bayes consistent. Then P_1 that does not have label-aligned bags on a subset of instances

$$B = \{x \in \mathcal{X} : \arg\max_{y \in \mathcal{Y}} P_1(\mathcal{S}_y | x) \neq \arg\max_{y \in \mathcal{Y}} P_1(y|x)\},$$

with $P_1(B) > 0$, since if $P_1(B) = 0$, Theorem (3.5) already guarantees that PLA-kNN is Bayes consistent. We construct a second distribution P_2 with label-aligned bags that has the same distribution over bags than P_1 but their associated Bayes classifiers have different values for each instance in B.

We keep the same instance marginal for both distributions:

$$P_2(x) = P_1(x), \quad \forall x \in \mathcal{X}.$$

For each $x \notin B$, P_1 and P_2 have the same distributions over labels $P_2(y|x) = P_1(y|x)$ and bag generation process $P_2(s|y,x) = P_1(s|y,x)$. For each $x \in B$, let

$$y_1 = \arg\max_{y \in \mathcal{Y}} P_1(y|x), \qquad y_2 = \arg\max_{y \in \mathcal{Y}} P_1(\mathcal{S}_y|x),$$

where it is clear that $y_1 \neq y_2$. We then *flip* the label distribution and bag-generation process in P_2 with respect to P_1 for y_1 and y_2,while keeping all other components equal in both distributions. Specifically,

$$\begin{aligned} &P_2(y_1|x) = P_1(y_2|x), \quad P_2(y_2|x) = P_1(y_1|x) \\ &P_2(s|y_1,x) = P_1(s|y_2,x), \quad P_2(s|y_2,x) = P_1(s|y_1,x) \\ &P_2(y|x) = P_1(y|x), \qquad P_2(s|y,x) = P_1(s|y,x) \qquad\qquad \forall s \in \mathcal{S}, \forall y \in \mathcal{Y} \setminus \{y_1, y_2\}. \end{aligned}$$

For any $x \in \mathcal{X}$ and $s \in \mathcal{S}$,

$$P_2(s|x) = \sum_{y \in \mathcal{Y}} P_2(y|x) P_2(s|y,x) = \sum_{y \in \mathcal{Y}} P_1(y|x) P_1(s|y,x) = P_1(s|x),$$

as the sum has the same $|\mathcal{Y}|$ components, but in a different order when $x \in B$; so it follows that $P_2(x,s) = P_1(x,s)$. Therefore, P_1 and P_2 induce *exactly the same distribution* over the observable pairs (x,s); any algorithm that learns only from (x,s)-samples cannot distinguish between them. However, by construction, the Bayes-optimal classifiers differs on B:

$$h^*_{P_1}(x) = \arg\max_y P_1(y|x) = y_1, \qquad h^*_{P_2}(x) = \arg\max_y P_2(y|x) = y_2.$$

Moreover, the distribution P_2 has label-aligned bags, since for any $x \in B$ where the condition did not hold for P_1

$$y_2 = \arg\max_{y \in \mathcal{Y}} P_2(y|x), \quad \text{and} \quad y_2 = \arg\max_{y \in \mathcal{Y}} P_1(\mathcal{S}_y|x) = \arg\max_{y \in \mathcal{Y}} P_2(\mathcal{S}_y|x).$$

The Bayes rules corresponding to P_1 and P_2 are distinct on a set of nonzero probability, as $P_1(B) > 0$ and $h^*_{P_1}(x) \neq h^*_{P_2}(x)$ for any instance in B. Since $P_1(x, s) = P_2(x, s)$, any consistent algorithm would converge to the same rule under both P_1 and P_2. Yet, as Bayes rules for P_1 and P_2 differ in a set of non-zero measure, no algorithm can achieve the Bayes risk for both cases. Therefore, if an algorithm is Bayes consistent for P_1 it can't be consistent for P_2, a scenario with label-aligned bags. Hence no algorithm can be Bayes consistent under strictly more general scenarios than those of PL A-kNN. □

C.8 Proof of Theorem 3.5

Before providing the proof of the theorem we state and prove some technical lemmas.

Lemma C.8. (Query-dependent convergence) There is an absolute constant $C > 0$ for which the following holds. Let h be the classifier from PL A-kNN algorithm 1 with maximum number of iterations T and confidence parameter δ satisfying (4). If the underlying distribution satisfies the relaxed label-aligned condition (7) for $G \subseteq \mathcal{X}$ and θ, the classifier h satisfies the following : for each $x \in G$ such that

$$n \geq \frac{C}{\mathrm{adv_A}(x)} \max\{\log(1/\mathrm{adv_A}(x)), \log(|\mathcal{Y}|/\delta)\},$$

we have that $h(x) \in \mathcal{Y}^\theta(x)$ with probability at least $1 - \delta^2$.

In addition, for each $x \in \mathcal{X} \setminus G$ such that

$$n \geq \frac{C}{\mathrm{adv_A}(x)} \max\{\log(1/\mathrm{adv_A}(x)), \log(|\mathcal{Y}|/\delta)\},$$

we have that $h(x) = h^*(x)$ with probability at least $1 - \delta^2$.

Proof. The proof is strictly analogous to the proof of Theorem 3.2, since for the elements $x \in G$ we have that $\arg\max_{y \in \mathcal{Y}} P(\mathcal{S}_y|x) \subseteq \mathcal{Y}^\theta(x)$. □

Lemma C.9. (Rates of convergence) Let C be the constant from Theorem 3.2, and h be the classifier from PL A-kNN algorithm 1 with maximum number of iterations T and confidence parameter δ satisfying (4). If the underlying distribution satisfies the relaxed label-aligned condition (7) for $G \subseteq \mathcal{X}$ and θ , we have

$$R(h) - R^* \leq \delta + P(\mathrm{adv_A}(x) \leq a_n) + \theta \cdot P(G)$$
$$\text{where} \quad a_n = \frac{C}{n} \max\big\{2\log(n), \log(|\mathcal{Y}|/\delta)\big\}.$$

with probability at least $1 - \delta$.

Proof.

Let P_G denote the probability distribution restricted to the set G and $R_G(h)$ denote de risk of a classification rule over P_G. Then , we have that:

$$R(h) = R_G(h)P(G) + R_{\mathcal{X}\setminus G}(h)(1 - P(G)).$$

From the proof of Theorem 3.4, and Lemma C.8 we have that for each $x \in \mathcal{X} \setminus G$ such that $\mathrm{adv_A}(x) > a_n$ we have $Pr_n(h(x) \neq h^*(x)) \leq \delta^2$, where Pr_n denotes probability over the choice of training points. Thus, for $\mathcal{X} \setminus G \sim P_{\mathcal{X}\setminus G}$

$$\mathbb{E}_n \mathbb{E}_{\mathcal{X}\setminus G} \mathbb{I}\{h(x) \neq h^*(x) | \, \mathrm{adv_A}(x) > a_n\} \leq \delta^2$$

and by Markov's inequality:

$$Pr_n[P_{\mathcal{X}\setminus G}(h(x) \neq h^*(x)|\,\mathrm{adv_A}(x) > a_n) \geq \delta] \leq \delta.$$

Thus, with probability $1-\delta$ over the training examples:

$$P_{\mathcal{X}\setminus G}(h(x) \neq h^*(x))|\,\mathrm{adv_A}(x) > a_n \leq \delta.$$

Therefore

$$\begin{aligned} &R_{\mathcal{X}\setminus G}(h) - R^*_{\mathcal{X}\setminus G} \leq P_{\mathcal{X}\setminus G}(h(x) \neq h^*(x)|\,\mathrm{adv_A}(x) > a_n) + P_{\mathcal{X}\setminus G}(\mathrm{adv_A}(x) \leq a_n) \\ &\leq \delta + P_{\mathcal{X}\setminus G}(\mathrm{adv_A}(x) \leq a_n). \end{aligned}$$

From the proof of Theorem 3.4 and Lemma C.8 we have that for each $x \in G$ such that $\mathrm{adv_A}(x) > a_n$ then $Pr_n(h(x) \notin \mathcal{Y}^\theta(x)) \leq \delta^2$, where Pr_n denotes probability over the choice of training points. Thus, for $G \sim P_G$

$$\mathbb{E}_n\mathbb{E}_G\mathbb{I}\{h(x) \notin \mathcal{Y}^\theta(x)|\,\mathrm{adv_A}(x) > a_n\} \leq \delta^2$$

and by Markov's inequality:

$$Pr_n[P_G(h(x) \notin \mathcal{Y}^\theta(x)|\,\mathrm{adv_A}(x) > a_n \geq \delta] \leq \delta.$$

Thus with probability $1-\delta$ over the training examples:

$$P_G(h(x) \notin \mathcal{Y}^\theta(x))|\,\mathrm{adv_A}(x) > a_n) \leq \delta.$$

Therefore,

$$\begin{aligned} &R_G(h) - R^*_G \leq P_G(\mathrm{adv_A}(x) \leq a_n) + \theta P_G(h(x) \in \mathcal{Y}^\theta(x)|\,\mathrm{adv_A}(x) > a_n) \\ &+ P_G(h(x) \notin \mathcal{Y}^\theta(x)|\,\mathrm{adv_A}(x) > a_n) \leq \delta + \theta + P_G(\mathrm{adv_A}(x) \leq a_n). \end{aligned}$$

We then conclude the proof since

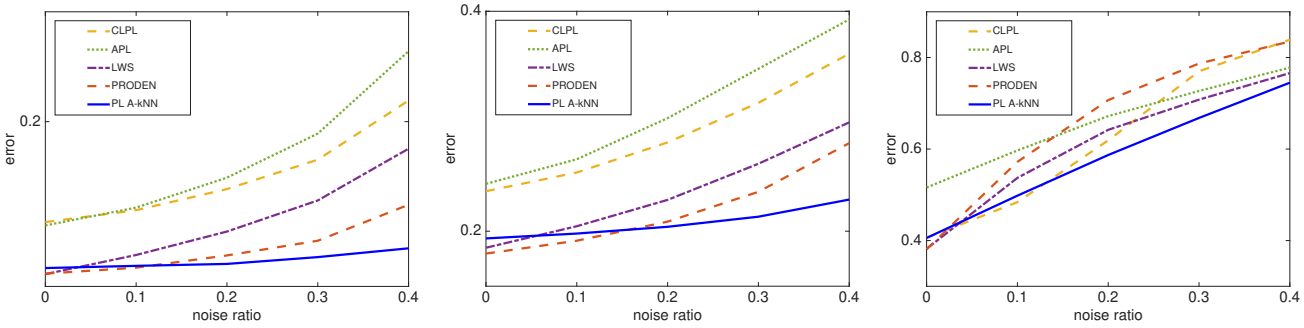
$$\begin{aligned} &R(h) - R^* = (R_G(h) - R^*_G)P(G) + (R_{\mathcal{X}\setminus G}(h) - R^*_{\mathcal{X}\setminus G})(1-P(G)) \\ &\leq \delta + \theta P(G) + P_G(\mathrm{adv_A}(x) \leq a_n)P(G) + P_{\mathcal{X}\setminus G}(\mathrm{adv_A}(x) \leq a_n)(1-P(G)) \\ &\leq \delta + P(\mathrm{adv_A}(x) \leq a_n) + \theta P(G). \end{aligned}$$

□

Proof of Theorem 3.5. The proof is analogous to the proof of Theorem 3.1, but using the rates from Lemma C.9 instead of the rates from Theorem 3.4. □

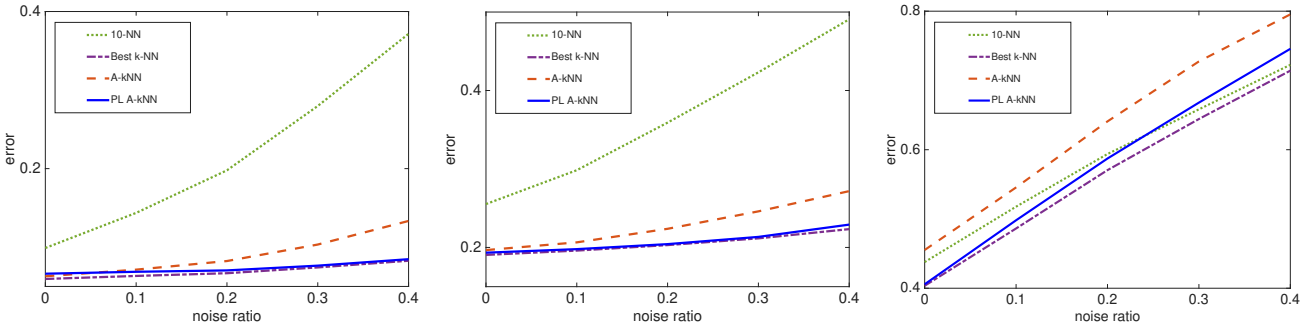
D Additional experimental and implementation details

D.1 Experimental results



(a) MNIST (b) CIFAR-10 (c) MirFlickr

Figure 3: Comparison of the error rates of PL A-kNN and state-of-the-art methods for MNIST, CIFAR-10, and MirFlickr under an increasing noise rate. The results show that PL A-kNN outperforms existing approaches across a wide range of noise levels.



(a) MNIST (b) CIFAR-10 (c) MirFlickr

Figure 4: Comparison of the error rates of PL A-kNN and kNN benchmarks for MNIST, CIFAR-10, and MirFlickr under an increasing noise rate. The results show that PL A-kNN outperforms 10-NN and A-kNN across a wide range of noise levels, while having comparable performance to the best k-NN.

D.2 Feature Preprocessing for PL A-kNN

We apply two preprocessing pipelines to obtain compact and stable representations suitable for nearest-neighbor retrieval: one designed for the vision benchmarks (CIFAR-10, MNIST, and Fashion-MNIST), and one tailored for the real-world partially labeled datasets (MirFlickr and MRSCv2). Both pipelines share a common structure—centering, ℓ_2-normalization, Gaussian-weighted KNN smoothing, and local density point transform—and differ only in that the real-world pipeline utilizes a signed cube-root transform in place of centering. The preprocessing implementations are provided in Adaptive Nearest Neighbours Repository.

Vision datasets (CIFAR-10, MNIST, Fashion-MNIST). For the standard vision benchmarks we apply the following pipeline.

1. **Centering.** Features are mean-centered using the training-set mean, which is then applied without modification to the test set.

2. **ℓ_2-normalization.** The centered features are ℓ_2-normalized, so that similarity computations depend only on the direction of each vector.

3. **Gaussian-weighted KNN smoothing.** Each feature vector is replaced by a convex combination of itself and the Gaussian-weighted mean of its 10 nearest neighbors,

$$\tilde{\mathbf{x}}_i = (1-\alpha)\,\mathbf{x}_i + \alpha \sum_{j \in \mathcal{N}(i)} w_{ij}\,\mathbf{x}_j, \qquad \alpha = 0.25, \tag{13}$$

where the weights $w_{ij} \propto \exp\!\big(-d_{ij}^2/(2\sigma_i^2)\big)$ use a *local* bandwidth σ_i set to the median pairwise distance from point i to its 10 neighbors. The smoothed vectors are re-normalized to unit ℓ_2-norm. Test features are smoothed using neighbors found in the *already-smoothed* training set, and are likewise re-normalized.

4. **Local density point transform.** To mitigate hubness in high-dimensional spaces, each feature vector is rescaled by its local neighbourhood density. Specifically, each point $\mathbf{x}_i$ is divided by $r_k(i)$, the mean distance to its $k = 50$ nearest neighbors,

$$\tilde{\mathbf{x}}_i = \frac{\mathbf{x}_i}{r_k(i)}, \qquad r_k(i) = \frac{1}{k} \sum_{j \in \mathcal{N}(i)} d_{ij}. \tag{14}$$

For test points, $r_k(q)$ is estimated using neighbors found in the smoothed training set. No ℓ_2-renormalization is applied after this step. The transformed vectors are used directly for Euclidean nearest-neighbor retrieval.

Real-world datasets (MirFlickr, MRSCv2). For the two real-world partially labeled benchmarks, which provide pre-extracted, high-dimensional descriptors, we use a slightly adapted pipeline. Steps 2–4 are identical to the vision pipeline above, the only difference is that centering is replaced by a nonlinear transform applied at the very beginning.

1. **Signed cube-root transform.** A signed cube-root transformation,

$$x \;\leftarrow\; \operatorname{sign}(x)\,|x|^{1/3}, \tag{15}$$

is applied element-wise to compress large activations and equalize feature variance, enhancing the similarity structure of the representation (Perronnin et al., 2010).

2. **ℓ_2-normalization, KNN smoothing ($\alpha = 0.1$, $k = 10$), and local density point transform ($k = 100$)** follow identically as in steps 2–4 of the vision pipeline.

In both pipelines, all statistics used for centering and bandwidth estimation are computed exclusively on the training set and applied without modification to the test set, preventing any leakage of test information into the preprocessing stage.

D.3 PL A-kNN: Handling multiple partial labels after T iterations

In the cases where the set of possible labels $\hat{s}$ of an instance contains more than one label after T iterations, we need a criterion to select a single label from $\hat{s}$. To this end, we adopt a heuristic that selects the label in $\hat{s}$ which was *closest to eliminating* all other possible labels during the iterative process. Specifically, we select

$$i = \underset{y \in \hat{s},\, k \le T}{\arg\min} \sqrt{k}\,\left(\Delta(n,k,\delta) - \left(P_n(\mathcal{S}_y | B_k(x)) - m_2(k)\right)\right),$$

where $B_k(x)$ denotes the ball containing the k nearest neighbors of instance x, and $m_2(k)$ represents the frequency of the second most frequent label within the bags of the k nearest neighbors.

The criterion presented above measures how close the difference between the frequencies of each of the labels in $\hat{s}$ and the second most frequent label is to surpassing the decision threshold across an increasing neighborhood size. Note that, if any of the labels in $\hat{s}$ actually surpassed the threshold, it would eliminate all other labels from $\hat{s}$. Since the threshold Δ decreases as k increases (with order $\sqrt{k}$), we scale the difference by $\sqrt{k}$ to have a more fair comparison across different neighborhood sizes. The detailed pseudocode for this selection criterion is provided in Algorithm 2. The algorithm implementation can be found in Adaptive Nearest Neighbours Repository.

Algorithm 2 PL A-kNN algorithm with disambiguation criterion

Inputs: Instance x
Training examples $\{(x_l, s_l)\}_{l=1}^{n}$
Maximum iterations T
Confidence parameter δ
Output: Label $h(x)$

Set $A = c_1\sqrt{\log(n) + \log(|\mathcal{Y}|/\delta)}$
Initialize $\hat{s} = \mathcal{Y}$ and $k = 0$
Initialize $\tau_1, \tau_2, \cdots, \tau_{|\mathcal{Y}|} = 0$ and $M = \mathbf{0} \in \mathbb{R}^{T\times|\mathcal{Y}|}$
while $|\hat{s}| > 1$ and $k < T$ **do**
  $k = k + 1$
  Find the k nearest neighbor of x and take l_k as its index in $l = 1, 2, ...n$
  $\Delta = \frac{A}{\sqrt{k}}$
  $\tau_y = \tau_y + \mathbb{I}\{y \in s_{l_k}\} \quad \forall y \in \mathcal{Y}$
  $\{m_1, m_2\} = \max_{y\in\hat{s}} 2\, \tau_y$ {max2 returns the two largest values}
  for $y \in \hat{s}$ **do**
    $M(k, y) = \sqrt{k}(\Delta - \frac{\tau_y - m_2}{k})$
    if $\frac{m_1 - \tau_y}{k} \geq \Delta$ **then**
      $\hat{s} = \hat{s} \setminus \{y\}$
    end if
  end for
end while
Find $\hat{y} = \arg\min_{y\in\hat{s},\, k\leq T} M(k, y)$
$h(x) = \hat{y}$