A Related Work

In recent years, label integration has been widely studied, and many label integration methods have been proposed. These methods can be divided into two categories depending on whether the information from the original feature view is used.

The first category does not use the information from the original feature view. Weighted Majority Voting (WMV) methods are typical among these methods, which infer the integrated label by directly combining each worker's weight and majority voting. The weight here reflects each worker's quality, which indicates the difference between a worker's labels and true labels, and it will be calculated through the multiple noisy label view. For example, Li *et al.* [1] propose IWMV. This method uses the multiple noisy label view and the integrated labels inferred by MV to calculate workers' weights. Then IWMV iteratively updates workers' weights and instances' labels until it reaches the condition of convergence or the number of iterations. Sheng *et al.* [2] propose four soft MV methods, which use the multiple noisy label view to obtain the majority class's certainty information. Tao *et al.* [3] propose DEWSMV for multi-class classification. DEWSMV represents workers' weights by defining a two-dimensional weight matrix on the multiple noisy label view and searches for the optimal result by using differential evolution (DE). Recently, Yang *et al.* [4] propose DTWMV. DTWMV evaluates workers' weights by workers' location and frequency information in a series of sequential decision trees, which are built based on the multiple noisy label view and the integrated labels inferred by MV.

The second category uses the information from the original feature view. In recent years, some researchers have noticed that relying solely on the multiple noisy view would limit the performance of label integration. Therefore, they have attempted to use the information from both views. For example, Rodrigues *et al.* [5] propose a novel general-purpose crowd layer (CL) to learn deep neural networks from crowds. It first constructs a single hidden layer with the potential true label, then uses the multiple noisy label view to build the output layer, and finally uses the original feature view to train the deep neural network (DNN) through backpropagation. Jiang *et al.* [6] propose multiple noisy label distribution propagation (MNLDP). It utilizes the original feature view to find the nearest neighbors and the multiple noisy label view to calculate label distributions. Subsequently, it absorbs and propagates the nearest neighbors' label distributions. Chen *et al.* [7] propose label augmented and weighted majority voting (LAWMV). It first makes use of the original feature view to find the nearest neighbors, and then utilizes the original feature view and the multiple noisy label view to weight each neighbor's labels. Recently, Pan *et al.* [8] propose three-way decision-based label integration (TDLI), which uses three-way decision theory to apply the different label integration strategies to different instances. For instances in the negative set, TDLI conducts more complex strategies which utilize both the original feature view and the multiple noisy label view. For instances in the positive set and boundary set, TDLI conducts easier strategies which only utilize the multiple noisy label view.

Although the second category utilizes the information from the original feature view as a supplement, it ignores the joint information between the original feature view and the multiple noisy view. Benefiting from the consensus and complementarity principle, dual-view learning can make use of the joint information. Therefore, we argue that it is necessary to introduce dual-view learning into crowdsourcing. Recently, some researchers also have noticed the importance of dual-view learning for crowdsourcing, but they focus on noise correction, which further processes the noise of integrated labels after label integration. Here, noise means that the integrated label of an instance is not consistent with its true label [9]. For example, Li *et al.* [10] propose multi-view noise correction (MVNC), and Ji *et al.* [11] propose dual-view noise correction (DVNC). Considering that there is no work to introduce dual-view learning into label integration, we propose a novel method called dual-view label integration (DVLI).

B Motivation

As mentioned above, although most existing label integration methods take into account utilizing the original feature view and the multiple noisy label view together, they merely use the information as a supplement, and ignore the joint information between the two views. To address this issue, our method introduces dual-view learning into label integration. The key to improving performance through dual-view learning is exploring the joint information between two views, that is, the consistent information and the complementary information [12]. Here, the consistent information refers to the consensus between the two views, which means that the results of integrated labels inferred by the two views should be as consistent as possible. The complementary information refers to the mutual supplements between two views, which means that the unique information of the two views can complement each other. It can be seen that the joint information can provide a more reliable and comprehensive description of the data for better inference of integrated labels. Therefore, our method presents two key points: (1) How to mine the joint information between two views. (2) How to utilize the joint information between two views.

For the first key point, the common target of the original feature view and the multiple noisy label view is to infer integrated labels, whose process can reflect the joint information. DVNC and MVNC mine the joint information via building a classifier on each view to infer integrated labels, respectively. Among them, the classifiers constructed by the two views independently reflect the complementary information, while the predicted results of the two views' classifiers reflect the consistent information. However, in the process, the complementary information between the two views is implicitly captured by the classifiers, which is rough and may lead to insufficient use of the joint information. To compensate for the shortcoming, for the original feature view, we build an instance weighted classifier to infer integrated labels. For the multiple noisy labels, we exploit weighted majority voting to infer integrated labels. By

doing so, instances' weights and workers' weights can reflect the complementary information to some extent, while the predicted results of the two views reflect the consistent information. The advantage is that we have two direct indicators — instances' weights and workers' weights, to reflect some unique information of two views, which can be utilized to guide fuller use of the joint information.

For the second key point, motivated by the consensus and complementarity principle, we exploit consistent information determination and complementary information fusion to fully utilize the joint information. For the consistent information, we continuously compare the predicted results from the two views to verify the consensus of the two views. The consensus is used to update the integrated labels iteratively, so that the quality of the integrated labels can be further ensured. For the complementary information, we argue that instances' weights and workers' weights can reflect some unique information of the original feature view and the multiple noisy label view, respectively. We let instances' weights and workers' weights update each other to correct each other's bias, which helps the two views reach the maximum consensus. After they reach maximum consensus, we fuse the complementary information to infer the final integrated labels. That is, we fuse the predicted class probabilities from the two views in the last iteration to fully utilize the complementary information. Based on these two key points, this paper proposes a dual-view label integration (DVLI) method.

C Algorithm

The learning process of DVLI is shown in **Algorithm 1**.

D Time complexity analysis

Seen from **Algorithm 1**, line 1-4 initialize $\boldsymbol{x}_i$'s integrated label $\hat{y}_i$ and weight λ_i, whose time complexity is $O(NQ)$. Lines 5-6 construct two datasets D^x and D^l. Line 8 builds a classifier G. If we assume that the training time complexity of G is $O(t_G)$, the time complexity of line 8 is $O(t_G)$. Lines 9-11 calculate workers' weights on D^l, whose time complexity is $O(NR)$. Lines 13-14 estimate predicted class probabilities and infer the integrated labels on D^x, and if we assume that the classification time complexity is $O(t_1)$, the time complexity of lines 13-14 is $O(t_1 + Q)$. Lines 15-16 estimate predicted class probabilities and infer the integrated labels on D^l, whose time complexity is $O(QR + Q)$. Line 20 update instances' weights, whose time complexity is $O(R)$. Thereby the time complexity of lines 12-21 is $O(N(t_1 + Q + QR + Q + R))$. If the number of iteration is T, the time complexity of lines 7-22 are $O(T(NR + G + N(t_1 + Q + QR + Q + R)))$. Lines 23-24 infer the final integrated labels, whose time complexity is $O(NQ)$. In summary, the total time complexity of DVLI is $O(NQ + T(NR + G + N(t_1 + Q + QR + Q + R)) + NQ)$. If we only take the highest order term into consideration, we can simplify the time complexity of

Algorithm 1 DVLI

Input: $D = \{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$ - a crowdsourced dataset
Parameter: α - the controlling factor
Output: $\{\hat{y}_i\}_{i=1}^{N}$ - the integrated labels

for $i = 1$ to N **do**
  Initialize the integrated label $\hat{y}_i$ of $\boldsymbol{x}_i$ by Eq.(1);
  Initialize the weight λ_i of $\boldsymbol{x}_i$ as 1;
end for
Construct dataset D^x on the original feature view, i.e., $D^x = \{(\boldsymbol{x}_i, \hat{y}_i, \lambda_i)\}_{i=1}^{N}$;
Construct dataset D^l on the multiple noisy label view, i.e., $D^l = \{(\boldsymbol{l}_i, \hat{y}_i, \lambda_i)\}_{i=1}^{N}$;
repeat
  Use D^x to build an instance weighted classifier G;
  for $r = 1$ to R **do**
    Use D^l to calculate the weight w_r of u_r by Eq.(2);
  end for
  for $i = 1$ to N **do**
    Estimate $P^x(c_q \mid \boldsymbol{x}_i)$ of $\boldsymbol{x}_i$ by Eq.(3);
    Infer the integrated label $\hat{y}_i^x$ of $\boldsymbol{x}_i$ by Eq.(4);
    Estimate $P^l(c_q \mid \boldsymbol{l}_i)$ of $\boldsymbol{x}_i$ by Eq.(5);
    Infer the integrated label $\hat{y}_i^l$ of $\boldsymbol{x}_i$ by Eq.(6);
    if $(\hat{y}_i^x == \hat{y}_i^l)$ and $(\hat{y}_i^x \neq \hat{y}_i)$ **then**
      Update the integrated label $\hat{y}_i$ of $\boldsymbol{x}_i$ as $\hat{y}_i^x$;
    end if
    Update the the weight λ_i of $\boldsymbol{x}_i$ by Eq.(7);
  end for
until ($\{\hat{y}_i\}_{i=1}^{N}$ don't change)
for $i = 1$ to N **do**
  Infer the integrated label $\hat{y}_i$ of $\boldsymbol{x}_i$ by Eq.(8);
end for
return $\{\hat{y}_i\}_{i=1}^{N}$.

DVLI as $O(TNQR + TNt_1 + Tt_G)$.

E Experiments on simulated datasets

We conduct a group of experiments on the whole 34 simulated crowdsourced datasets from the CEKA platform to validate the effectiveness of DVLI. The description of these datasets are summarized as Table 1, and these datasets are all published on the CEKA platform at https://ceka.sourceforge.net/. Considering that some simulated datasets exist missing values which may influence the construction of the original feature view, we use the mean of numerical features or the mode of nominal features from the available data to replace missing

feature values by the unsupervised feature filter ReplaceMissingValues in the WEKA platform [13]. We then simulate crowd workers to label the data after preprocessing. we first hide each instance's original true label. Subsequently, we obtain labels for each instance from simulated crowd workers. Although employing crowd workers to label data is a cheaper approach compared to employing experts, the cost of labeling still needs to be taken into consideration. If a method has a superior performance while a small number of workers are employed, that is to say, the method can reduce the labeling cost in real applications by employing a small number of workers. So in the following simulated experiments, we fix a small value of 5 for the number of workers. Since the probability of guessing the true label correctly in binary classification datasets is 50%, we set each worker's labeling quality q to be no less than 0.5, where q represents the probability that the labels obtained from this worker are exactly correspond to the true labels. Thus, the labeling quality q of each crowd worker is randomly generated from a uniform distribution in the interval $(0.5, 0.9]$. To mitigate the effects of randomness in our simulation, each experiment is repeated 10 times independently and the final result is each method's average integration accuracy.

Table 2 shows the comparison results in terms of the integration accuracy on these 34 simulated datasets. The last row details the average integration accuracy of each method. To futhur compare the difference between each pair of methods, we perform the Wilcoxon signed-rank test [14, 15] via the knowledge extraction based on evolutionary learning (KEEL) [16]. Table 3 summarizes the Wilcoxon test results. In Table 3, the symbol $\bullet$ means that the method in the row is significantly superior to the method in the corresponding column. In contrast, the symbol $\circ$ indicates that the method in the column is significantly superior to the method in the corresponding row. The lower and upper diagonal levels of significance are α=0.05 and α=0.1, respectively. We summarize these experimental results as follows:

1) The average integration accuracy of DVLI, MNLDP, MVNC and DVNC are 93.39%, 91.63%, 91.95% and 91.54%, which are much higher than those of IWMV (89.07%) and DTWMV (88.97%). This result proves that utilizing the information from both the original feature view and the multiple noisy label view can effectively improve the performance of label integration.

2) The average integration accuracy of DVLI, MVNC and DVNC are 93.39%, 91.95% and 91.54%, which are much higher than that of TDLI (89.85%). This result proves that utilizing the joint information between the original feature view and the multiple noisy label view can further improve the performance of label integration.

3) The average integration accuracy of DVLI is 93.39%, which is much higher than that of MVNC (91.95%) and DVNC (91.54%). This result proves that our DVLI makes fuller use of the joint information between the original feature view and the multiple noisy label view than MVNC and DVNC.

Table 1: Description of simulated datasets.

Dataset	Instances	Features	Classes	Missing values	Feature type
anneal	898	38	6	yes	hybrid
audiology	226	69	24	yes	nominal
autos	205	25	7	yes	hybrid
balance-scale	625	4	3	no	numeric
biodeg	1055	41	2	no	numeric
breast-cancer	286	9	2	yes	nominal
breast-w	699	9	2	yes	numeric
car	1728	6	4	no	nominal
credit-a	690	15	2	yes	hybrid
credit-g	1000	20	2	no	hybrid
diabetes	768	8	2	no	numeric
heart-c	303	13	5	yes	hybrid
heart-h	294	13	5	yes	hybrid
heart-statlog	270	13	2	no	numeric
hepatitis	155	19	2	yes	hybrid
horse-colic	368	22	2	yes	hybrid
hypothyroid	3772	29	4	yes	hybrid
ionosphere	351	34	2	no	numeric
iris	150	4	3	no	numeric
kr-vs-kp	3196	36	2	no	nominal
labor	57	16	2	yes	hybrid
letter	20000	16	26	no	numeric
lymph	148	18	4	no	hybrid
mushroom	8124	22	2	yes	nominal
segment	2310	19	7	no	numeric
sick	3772	29	2	yes	hybrid
sonar	208	60	2	no	numeric
spambase	4601	57	2	no	numeric
tic-tac-toe	958	9	2	no	nominal
vehicle	846	18	4	no	numeric
vote	435	16	2	yes	nominal
vowel	990	13	11	no	hybrid
waveform	5000	40	3	no	numeric
zoo	101	17	7	no	hybrid

4) Seen from the Wilcoxon signed-rank test results, whatever at the significance levels of the lower diagonal α=0.05 or the upper diagonal α=0.1, DVLI significantly outperforms all its competitors, validating the effectiveness of DVLI.

Furthermore, to validate whether DVLI is still effective on other labeling quality distributions, we also conduct another group of experiments. The label

Table 2: Integration accuracy (%) comparisons on the Uniform distribution.

dataset	IWMV	DTWMV	MNLDP	TDLI	MVNC	DVNC	DVLI
anneal	95.43	95.20	95.77	97.55	97.87	98.33	96.55
audiology	98.67	92.92	93.81	99.56	88.98	93.36	99.56
autos	97.07	92.68	90.24	96.10	94.88	93.66	97.56
balance-scale	92.00	91.68	96.96	93.28	88.56	89.60	95.52
biodeg	78.77	83.51	92.51	78.77	82.72	86.54	86.27
breast-cancer	72.38	74.83	84.97	72.38	70.52	71.68	74.48
breast-w	72.82	78.11	90.41	72.82	90.70	92.99	91.85
car	94.50	94.66	91.96	95.20	98.00	95.56	98.17
credit-a	91.30	92.75	91.88	91.30	92.12	91.30	93.77
credit-g	92.40	91.60	78.70	92.40	91.40	85.30	94.20
diabetes	91.54	91.28	83.98	87.24	86.77	81.51	92.45
heart-c	90.76	89.11	90.10	95.71	93.89	90.76	95.71
heart-h	91.84	90.82	94.90	94.56	90.31	89.46	95.24
heart-statlog	91.85	93.70	88.89	87.78	92.07	91.85	93.33
hepatitis	76.77	71.61	92.26	76.77	83.03	81.94	83.58
horse-colic	73.64	78.26	83.42	73.64	77.64	83.15	83.34
hypothyroid	90.54	89.89	96.87	95.60	99.32	99.84	98.14
ionosphere	88.89	86.89	86.32	85.75	92.36	90.60	89.17
iris	88.67	88.67	97.33	94.00	97.93	97.33	98.67
kr-vs-kp	94.09	95.28	97.37	94.09	98.24	99.72	97.25
labor	75.44	84.21	91.23	75.44	87.54	84.21	87.72
letter	97.96	95.43	99.82	99.79	99.48	97.69	99.52
lymph	87.16	88.24	92.57	93.24	93.31	91.22	97.43
mushroom	89.30	90.26	98.36	85.49	97.01	97.93	97.14
segment	95.97	94.46	97.79	98.74	99.63	99.52	96.97
sick	94.46	93.64	97.75	93.35	99.15	99.47	94.70
sonar	92.79	92.79	89.42	92.79	93.19	91.83	92.79
spambase	94.85	95.04	83.18	94.85	97.42	97.07	92.50
tic-tac-toe	80.06	79.96	73.90	80.06	82.65	83.51	83.59
vehicle	97.02	96.57	92.43	97.40	96.55	94.92	94.68
vote	82.07	81.15	87.13	82.07	95.07	95.86	93.33
vowel	97.27	94.44	99.29	99.09	98.73	95.15	99.09
waveform	88.12	88.14	97.02	90.96	88.31	89.50	92.86
zoo	92.08	87.13	97.03	97.03	90.99	90.10	98.02
Average	89.07	88.97	91.63	89.85	91.95	91.54	93.39

Table 3: Wilcoxon test results on the uniform distribution.

	IWMV	DTWMV	MNLDP	TDLI	MVNC	DVNC	DVLI
IWMV	-		○	○	○	○	○
DTWMV		-	○		○	○	○
MNLDP		•	-				○
TDLI				-	○		○
MVNC	•	•		•	-		○
DVNC	•	•				-	○
DVLI	•	•	•	•	•	•	-

quality q of each crowd worker is randomly generated from a Gaussian distribution $N(0.7, 0.2^2)$. The related experiment results are shown in Tables 4–5. Seen from these results, we can also summarize that DVLI significantly outperforms other related methods on Gaussian labeling quality distribution.

Table 4: Integration accuracy (%) comparisons on the Gaussian distribution.

dataset	IWMV	DTWMV	MNLDP	TDLI	MVNC	DVNC	DVLI
anneal	96.55	96.04	96.99	98.00	98.14	97.42	96.82
audiology	96.90	88.50	94.25	99.12	86.46	92.04	98.23
autos	96.10	92.73	90.24	96.10	95.12	95.32	95.76
balance-scale	94.40	92.53	93.76	90.08	87.23	86.16	96.00
biodeg	79.72	81.23	85.40	79.72	85.77	88.44	79.72
breast-cancer	91.96	91.26	62.24	91.96	83.08	82.52	91.26
breast-w	63.81	76.39	79.97	63.81	87.81	89.56	89.27
car	96.12	96.27	91.90	96.53	98.71	95.89	99.11
credit-a	95.51	96.52	88.41	85.65	90.75	88.41	92.90
credit-g	99.70	99.70	90.00	98.50	94.80	90.10	98.60
diabetes	89.45	88.41	82.94	79.43	80.87	76.04	87.24
heart-c	89.80	88.84	92.08	96.37	93.66	90.23	94.98
heart-h	92.18	94.25	93.20	93.54	91.90	87.41	97.41
heart-statlog	78.89	83.70	83.33	78.89	85.41	83.33	87.04
hepatitis	80.00	80.65	85.81	80.00	88.45	89.68	89.03
horse-colic	87.50	90.76	89.13	87.50	89.08	88.32	91.30
hypothyroid	97.53	97.58	96.71	98.09	99.91	99.79	99.96
ionosphere	82.91	68.95	84.33	67.81	77.66	82.91	80.91
iris	98.00	97.33	99.33	99.33	97.80	98.00	100.00
kr-vs-kp	95.90	94.99	82.79	90.36	98.06	99.37	96.25
labor	82.46	89.47	84.21	82.46	85.09	84.21	91.23
letter	95.94	95.08	95.74	96.68	99.39	99.57	99.86
lymph	93.92	94.73	95.95	97.97	94.39	90.47	97.50
mushroom	96.17	95.84	94.36	94.72	99.97	99.47	98.01
segment	84.81	84.02	96.97	95.11	98.26	97.69	92.24
sick	84.44	81.92	62.35	84.97	96.08	97.80	87.25
sonar	69.23	69.23	92.31	69.23	71.83	74.52	75.96
spambase	58.07	59.77	92.91	58.51	75.83	84.83	76.68
tic-tac-toe	84.24	86.85	70.46	84.24	86.92	87.79	88.94
vehicle	95.63	95.65	89.72	95.27	95.33	87.28	92.47
vote	91.26	90.57	96.09	91.26	97.17	97.01	94.48
vowel	96.16	90.56	99.90	99.29	97.69	93.95	97.87
waveform	92.80	93.15	95.20	94.44	94.21	91.88	94.99
zoo	95.05	93.07	97.03	96.04	95.94	96.44	100.00
Average	88.92	88.72	89.00	88.56	91.14	90.70	92.63

Table 5: Wilcoxon test results on the Gaussian distribution.

	IWMV	DTWMV	MNLDP	TDLI	MVNC	DVNC	DVLI
IWMV	-				○		○
DTWMV		-			○		○
MNLDP			-				○
TDLI				-	○		○
MVNC	●	●		●	-		○
DVNC						-	○
DVLI	●	●	●	●	●	●	-

F Sensitivity analysis

a) Controlling factor

In this subsection, we conduct a set of sensitivity analysis experiments of the controlling factor α. In DVLI, we infer the final integrated labels by fusing the predicted class probabilities of the original feature view and the multiple noisy label view. As mentioned above, α is the fusion proportion of the original feature view. Figure 1 shows the results of the integration accuracy comparisons when α varies from 0 to 1.0.

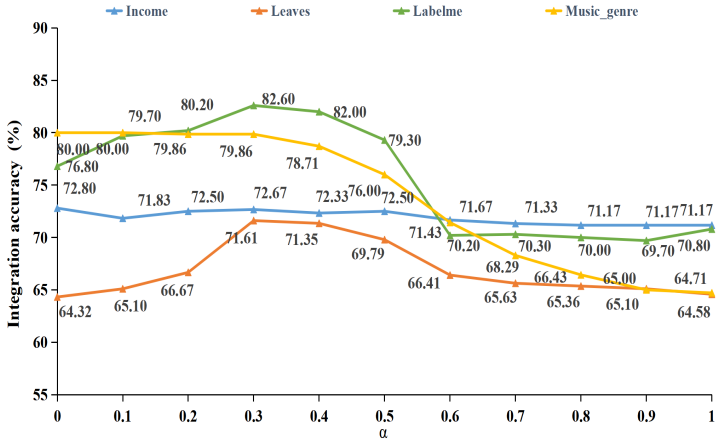
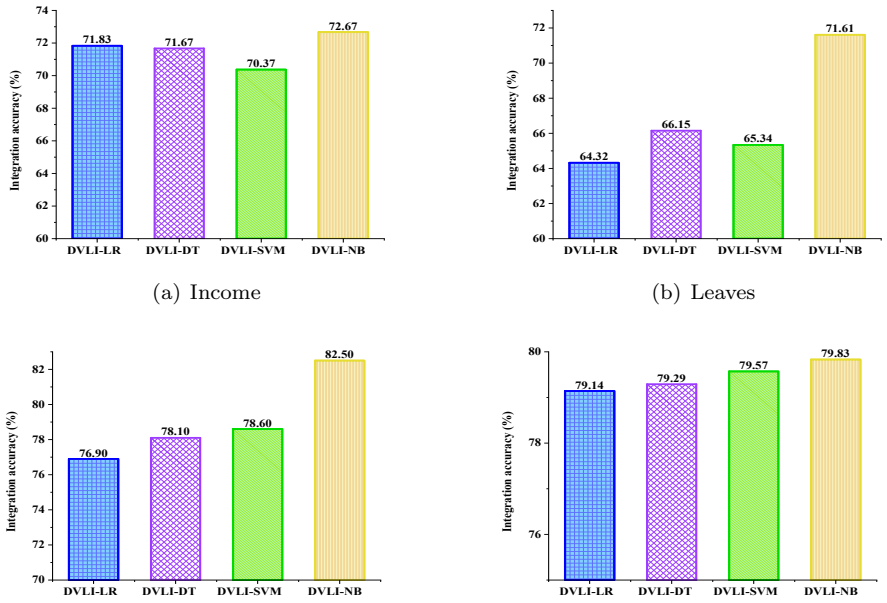


Figure 1: Integration accuracy (%) comparisons when α varies from 0 to 1.0.

Based on the experimental results, we summarize and analyze which value of α should be set in DVLI. On the "Income" dataset, the integration accuracy is insensitive to the value of α. On datasets "Leaves" and "Labelme", the integration accuracy presents a trend of first rising and then falling, reaching its peak at $\alpha = 0.3$. On the "Music_genre" dataset, the integration accuracy presents a trend of gradually falling when $\alpha > 0.3$. We can conclude that in most cases the multiple noisy label view plays a leading role in complementary information fusion. Therefore, we set α to 0.3 in all our experiments.

b) Classifier

In this subsection, we conduct a set of sensitivity analysis experiments of the classifier. Considering the execution efficiency of the algorithm, we choose four widely used and efficient classifiers to analyze the impact of using different classifiers on DVLI, including Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), and Naive Bayes (NB). Figure 2 shows the results of using different classifiers. It can be observed that choosing different classifiers will have a certain impact on the experimental results. When NB is



(a) Income (b) Leaves

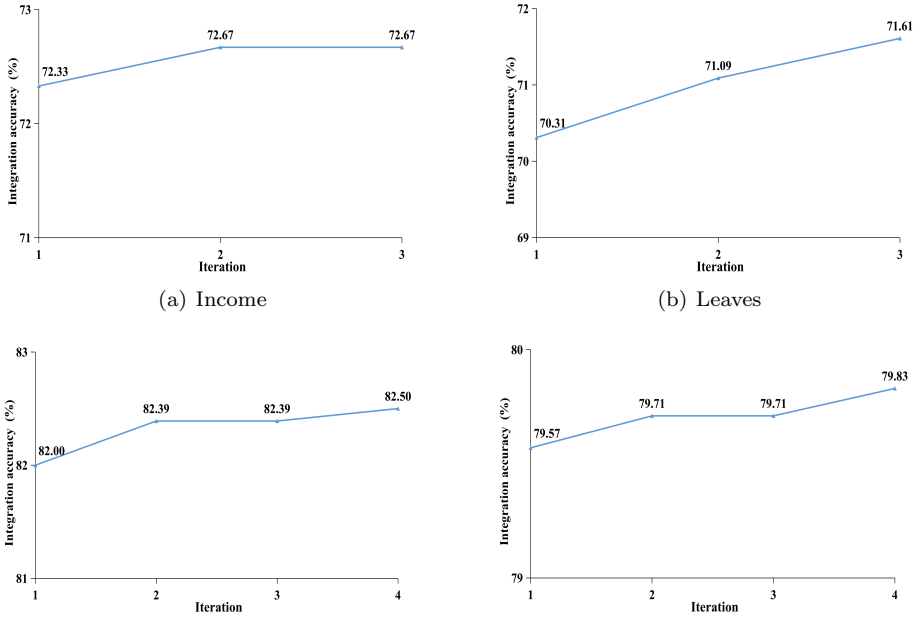
(c) Labelme (d) Music_genre

Figure 2: Integration accuracy (%) comparisons with different classifiers.

chosen as the classifier, the performance of DVLI is the best, while when other classifiers are chosen, the performance of DVLI is slightly inferior.

Based on the experimental results, we summarize and analyze what kind of classifier should be chosen in DVLI. In our DVLI, the classifier is used to infer the integrated labels of the original feature view. The consistency of the integrated labels between the original feature view and the multiple noisy label view serves for the iterative judgment. During this process, we have two requirements for the classifier: 1) The classifier should be insensitive to noise. In the crowdsourcing scenario, each instance corresponds to a multiple noisy label set instead of a single true label. The initial training labels of the classifier are the initialized integrated labels. However, no matter which label integration algorithm is used for initial integration, there is still a certain degree of noise in the integrated labels. The more insensitive the classifier to noise, the more accurate the trained classifier, the more accurate the iterative judgment, and the more correct the direction of iterative progress. 2) The classifier should be efficient. Considering that we need to retrain the classifier in each iteration, an efficient classifier is better. Compared to the other classifiers we chose, NB is relatively insensitive to noise due to its probabilistic framework. What's more, the training process of NB is simple and computationally efficient. Therefore, we use NB as the classifier in our DVLI.

G Convergence analysis



(a) Income (b) Leaves

(c) Labelme (d) Music_genre

Figure 3: Integration accuracy (%) comparisons during iterations.

In this subsection, we conduct a set of experiments to verify the convergence of DVLI on the four real-world datasets. We compare the changes of DVLI in terms of the integration accuracy during iterations, shown in Figure 3.

We find that the integration accuracy of the four real-world datasets changes only within a maximum of four iterations. After four iterations, the integration accuracy stabilizes and reaches its highest value, indicating that DVLI has effectively converged to a stable state. These experimental results demonstrate that DVLI can achieve rapid convergence within a limited number of iterations.

H Ablation experiments

To assess the contribution of each component in DVLI, we compare DVLI with its three variants in terms of integration accuracy. Considering that "Income" is a binary classification dataset which may have limitations, we conduct the experiment on the multi-class dataset "Leaves". The three variants are referred as DVLI-I, DVLI-L, and DVLI-F. DVLI-I removes the iterative process from DVLI, which directly fuses the predicted class probabilities of the original feature view and the multiple noisy label view. That is, DVLI-I does not utilize the

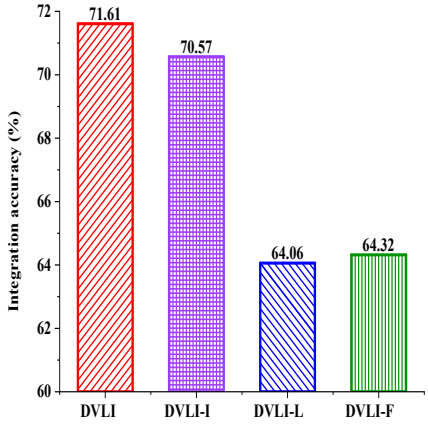


Figure 4: Integration accuracy (%) comparisons for DVLI versus DVLI-I, DVLI-L, and DVLI-F.

consistent information between the two views for iteration. DVLI-L removes the proportion of the multiple noisy label view in the final inference, which directly takes the converged integrated labels of the original feature view as the final results. Similarly, DVLI-F removes the proportion of the original feature view in the final inference, which directly takes the converged integrated labels of the multiple noisy label view as the final results. That is, DVLI-L and DVLI-F do not utilize the complementary information for final integrated label fusion. Figure 4 presents the comparison results for DVLI, DVLI-I, DVLI-L, and DVLI-F. Seen from it, we can observe as follows:

1) The integration accuracy of DVLI-I is lower than that of DVLI, proving the necessity of utilizing the consistent information. That is, as the consistency between the two views becomes increasingly greater, the integration accuracy also increases.

2) The integration accuracy of DVLI-L and DVLI-F is lower than that of DVLI, demonstrating the necessity of utilizing complementary information. That is, the performance becomes worse when any view is taken away, validating the effectiveness of the complementary information fusion.

I Conclusions and Future Work

In this paper, we propose dual-view label integration (DVLI) for crowdsourcing. In DVLI, we introduce dual-view learning into label integration to make full use of the joint information between the original feature view and the multiple noisy label view. For the consistent information, we conduct a series of iterations to

determine the prediction consistency of the two views, ensuring that the two views reach the maximum consensus. For the complementary information, we fuse the final predicted class probabilities from the two views for final inference, allowing that the two views use complementarity to make up for their own drawbacks. The extensive experiment results demonstrate that DVLI significantly outperforms all the other state-of-the-art algorithms used to compare.

In the current version, we set the controlling factor α to a fixed value, which may not be suitable for all datasets. Therefore, our future work will focus on how to design an adaptive controlling factor. In addition, the current version of DVLI is designed for general crowdsourced datasets. We will consider to develop other versions for some special scenarios in our future work, such as class imbalance, label sparsity, and so on.

References

[1] Li H, Yu B. Error rate bounds and iterative weighted majority voting for crowdsourcing. CoRR, 2014, abs/1411.4086

[2] Sheng V S, Zhang J, Gu B, Wu X. Majority voting and pairing with multiple noisy labeling. IEEE Transactions on Knowledge and Data Engineering, 2017, 31(7): 1355–1368

[3] Tao F, Jiang L, Li C. Differential evolution-based weighted soft majority voting for crowdsourcing. Eng. Appl. Artif. Intell., 2021, 106: 104474

[4] Yang W, Li C, Jiang L. Learning from crowds with decision trees. Knowl. Inf. Syst., 2022, 64(8): 2123–2140

[5] Rodrigues F, Pereira F C. Deep learning from crowds. In: McIlraith S A, Weinberger K Q, eds, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. 2018, 1611–1618

[6] Jiang L, Zhang H, Tao F, Li C. Learning from crowds with multiple noisy label distribution propagation. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(11): 6558–6568

[7] Chen Z, Jiang L, Li C. Label augmented and weighted majority voting for crowdsourcing. Inf. Sci., 2022, 606: 397–409

[8] Pan C, Jiang L, Li C. Three-way decision-based label integration for crowdsourcing. Pattern Recognit., 2025, 158: 111034

[9] Li C, Sheng V S, Jiang L, Li H. Noise filtering to improve data and model quality for crowdsourcing. Knowl. Based Syst., 2016, 107: 96–103

[10] Li X, Li C, Jiang L. A multi-view-based noise correction algorithm for crowdsourcing learning. Inf. Fusion, 2023, 91: 529–541

[11] Ji Q, Jiang L, Zhang W. Dual-view noise correction for crowdsourcing. IEEE Internet Things J., 2023, 10(13): 11804–11812

[12] Xu C, Tao D, Xu C. A survey on multi-view learning. CoRR, 2013, abs/1304.5634

[13] Witten I H, Frank E, Hall M A. Data mining: practical machine learning tools and techniques, 3rd Edition. Morgan Kaufmann, Elsevier, 2011

[14] Demsar J. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res., 2006, 7: 1–30

[15] Jiang L, Zhang L, Li C, Wu J. A correlation-based feature weighting filter for naive bayes. IEEE Trans. Knowl. Data Eng., 2019, 31(2): 201–213

[16] Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Multiple Valued Log. Soft Comput., 2011, 17(2-3): 255–287