Runtime Time

Convergence

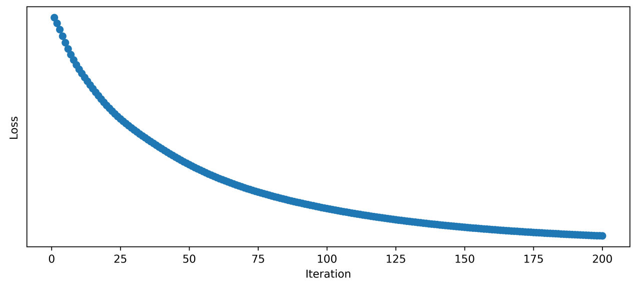

Figure 1: The process of LSDMLO

Figure 1 shows the iterative optimization process of W based on the rcv1subset1 dataset.

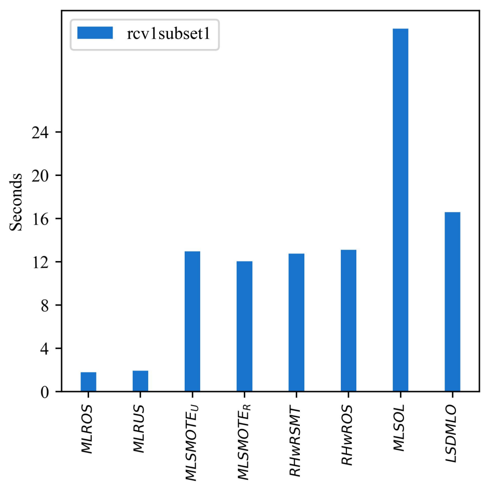

Figure 2: The time comparison of sampling methods

Figure 1 shows the sampling methods time comparasion based on the rcv1subset1 dataset.

Time complexity analysis

Time complexity mainly consists of two parts: the acquisition process of W and the sampling process.

W initialization and iterations: In the initialization step, the time complexity of initializing W is $O(nd^2 + d^3 + ndq + d^2q)$, where n is the number of instances, d is the dimensionality of the dataset, and q is the number of labels. The time complexities for calculating label correlation using cosine similarity is $O(nq^2)$. The initialization of L_f has a time complexity of $O(d^3 + q^3)$. In the iteration step, the time complexity of calculating the gradient ϕ_1 is $O(nd^2 + d^2q + ndq + n^2d + dq^2)$. Therefore, the total time complexity of the acquisition process of W can be represented as $O((t + 1)(nd^2 + d^2q + ndq) + t(n^2d + dq^2) + d^3 + q^3)$, where t is the number of iterations. Sampling process: The complexity of creating synthetic instances is $O(pn(q + d))$. Therefore, the overall complexity of LSDMLO is $O((t + 1)(nd^2 + d^2q + ndq) + t(n^2d + dq^2) + d^3 + q^3 + pn(q + d))$.

Theoretical analysis and contribution

Our primary contribution lies in the novel weighting strategy introduced for defining neighborhoods in multi-label datasets. It is crucial to note that the proposed metric, the label-specific distance, had not been previously considered within the context of machine learning. Additionally, the incorporation of feature selection methods for developing a weighted distance metric is a unique aspect that has not been explored in previous studies. These factors establish LSDMLO as a sophisticated oversampling strategy with substantial potential, as it extends the original distance-based sampling algorithms.

There is an interesting relation between LSDMLO and the application of some distance-based oversampling methods in an embedding space. Feature extraction strategies such as Locally Linear Embedding (LLE) have some mathematical similarities with the process of W: it constructs a k-NN graph and defines weights to multiply the input variables. However, this is done with a completely different purpose: to map the data into a low-dimensional manifold, where the patterns become more distinguishable.

LSDMLO exhibits a remarkable degree of flexibility, allowing it to partially emulate feature extraction methods. Specifically, the coefficient matrix W penalizes attributes in the data matrix X that lack correlation with a certain label. However, we emphasize the significant distinctions between LSDMLO and studies applying some distance-based sampling method (such as MLSMOTE) in the embedding space.

Remark:

- Dimensionality Reduction Relevance: While transforming the original space into a lower-dimensional feature space is crucial in applications like computer vision or text analytics, in domains such as business analytics, this step often doesn't enhance prediction. In tabular datasets, weak correlations between variables are common, and predictive performance relies on the relationship between each original variable and the labels.
- Weighting Strategy vs. Feature Extraction: In LSDMLO, the weighted validity based on correlation incorporates the coefficient matrix W, which is constrained by cosine similarity. This constraint ensures the validity of the weights but also imposes limits on the sparsity of the resulting regression matrix W. LSDMLO constructs neighborhoods by evaluating the distinct contributions of each label's corresponding features. It does this while minimizing the loss of essential common features, striking a balance between preserving meaningful correlations and capturing the uniqueness of individual features within each label.
- Normalization when calculating weights: In some cases, the normalized coefficient matrix can make the weights of different features comparable, especially when the input features have different units or orders of magnitude.

This can make it easier to explain and compare the impact of different features on the target variable. In this article, we assume that features have similar orders of magnitude. and we used the softmax function to smooth the coefficients in W.

- Task-Dependent Considerations: There are task-specific embedding strategies, and it's not clear which feature extraction strategy works best for a given application (datasets and models). These issues are task-dependent, whereas LSDMLO can be universally applied across diverse domains.

Partial supplementary experiments

BR	Macro-F	Macro-AUC	Macro-PR	RankLoss
Base	0.2275±0.0055	0.6032±0.0033	0.0992±0.0029	0.6352±0.0055
$\text{MLSMOTE}_{\text{U}}$	0.2222±0.0076	0.6081±0.0049	0.0952±0.0035	0.6300±0.0060
$\text{MLSMOTE}_{\text{R}}$	0.2248±0.0068	0.6039±0.003	0.0982±0.0033	0.638±0.0041
MLSOL	0.2243±0.0068	0.6049±0.0032	0.0973±0.0033	0.6347±0.0071
MLROS	0.2239±0.0059	0.6017±0.0031	0.0978±0.0020	0.6392±0.0078
MLRUS	0.2284±0.0076	0.6033±0.0040	0.0996±0.0047	0.6376±0.0057
RHwRSMT	0.1162±0.0056	0.5986±0.0035	0.0982±0.0015	0.647±0.0052
RHwRROS	0.2237±0.0051	0.6012±0.0030	0.0976±0.0017	0.6392±0.0079
LSDMLO	0.2307±0.0035	0.6076±0.0012	0.1020±0.0019	0.6231±0.0024
MLkNN	Macro-F	Macro-AUC	Macro-PR	RankLoss
Base	0.1376±0.0087	0.7137±0.0049	0.1678±0.0039	0.1197±0.0013
$\text{MLSMOTE}_{\text{U}}$	0.1795±0.0057	0.737±0.0046	0.1741±0.0057	0.1282±0.0034
$\text{MLSMOTE}_{\text{R}}$	0.1847±0.0055	0.7247±0.0058	0.1835±0.0053	0.1241±0.0027
MLSOL	0.2242±0.0061	0.7301±0.0057	0.1942±0.0067	0.1195±0.0024
MLROS	0.1673±0.0076	0.7056±0.0042	0.1645±0.0059	0.1228±0.0016
MLRUS	0.1361±0.0072	0.7125±0.0049	0.1671±0.0039	0.1200±0.0013
RHwRSMT	0.0973±0.0073	0.7015±0.0050	0.1709±0.0061	0.1258±0.0017
RHwRROS	0.1673±0.0076	0.7056±0.0042	0.1645±0.0059	0.1228±0.0016
LSDMLO	0.2376±0.0043	0.7286±0.0061	0.1869±0.0044	0.1178±0.0017
CC	Macro-F	Macro-AUC	Macro-PR	RankLoss
Base	0.2182±0.0076	0.5994±0.0052	0.0945±0.0034	0.6416±0.0094
$\text{MLSMOTE}_{\text{U}}$	0.2152±0.0055	0.6031±0.0036	0.092±0.0025	0.6372±0.0078
$\text{MLSMOTE}_{\text{R}}$	0.2203±0.0087	0.6026±0.0049	0.0958±0.0037	0.6414±0.0058
MLSOL	0.2217±0.0082	0.6042±0.0041	0.0954±0.0036	0.6313±0.0085
MLROS	0.2185±0.0057	0.5995±0.0035	0.0946±0.0026	0.6453±0.0075
MLRUS	0.2173±0.0072	0.5988±0.0047	0.0942±0.0034	0.6442±0.0079
RHwRSMT	0.2042±0.0071	0.5738±0.0035	0.0824±0.0023	0.7232±0.0062
RHwRROS	0.2185±0.0057	0.5995±0.0035	0.0946±0.0026	0.6453±0.0075
LSDMLO	0.2236±0.0043	0.6075±0.0040	0.0975±0.0016	0.6302±0.0062
RAkEL	Macro-F	Macro-AUC	Macro-PR	RankLoss
Base	0.2139±0.0076	0.5967±0.0044	0.0934±0.003	0.6433±0.0057
$\text{MLSMOTE}_{\text{U}}$	0.2138±0.0048	0.6023±0.003	0.0911±0.0018	0.6371±0.0046
$\text{MLSMOTE}_{\text{R}}$	0.2174±0.0076	0.599±0.0042	0.0952±0.0027	0.6454±0.0071
MLSOL	0.2165±0.0053	0.601±0.005	0.0938±0.0032	0.6403±0.0085
MLROS	0.2159±0.0052	0.5977±0.0034	0.0946±0.0037	0.6432±0.0075
MLRUS	0.2155±0.007	0.5969±0.0041	0.0945±0.0033	0.6437±0.0022
RHwRSMT	0.1936±0.0084	0.5965±0.0027	0.0962±0.0027	0.651±0.0059
RHwRROS	0.2136±0.0044	0.5961±0.0027	0.0927±0.0019	0.6449±0.0037
LSDMLO	0.2174±0.0025	0.6037±0.0018	0.0994±0.0027	0.6378±0.0036
COCOA	Macro-F	Macro-AUC	Macro-PR	RankLoss
Base	0.2969±0.0031	0.8758±0.0034	0.2903±0.0041	0.0747±0.0006
$\text{MLSMOTE}_{\text{U}}$	0.3229±0.003	0.9042±0.0034	0.299±0.0046	0.0685±0.0007
$\text{MLSMOTE}_{\text{R}}$	0.3257±0.0022	0.8917±0.0031	0.3005±0.0043	0.0687±0.0005
MLSOL	0.3334±0.0052	0.8909±0.0024	0.3003±0.004	0.0605±0.0014
MLROS	0.3263±0.0034	0.8950±0.0035	0.2997±0.0036	0.0671±0.0005
MLRUS	0.2960±0.0024	0.8749±0.0035	0.2893±0.0037	0.0752±0.0006
RHwRSMT	0.3002±0.0035	0.8683±0.0037	0.2794±0.005	0.0678±0.0007
RHwRROS	0.3257±0.0037	0.8947±0.0033	0.2997±0.0036	0.0674±0.0005
LSDMLO	0.3382±0.0041	0.9060±0.0023	0.3014±0.0024	0.0624±0.0008

Table 1: Results of sampling methods on rcv1subset1 with different base classifier

The total table presents the results of different base classifiers on the rcv1subset1 dataset. In addition to MLSOL's performance on Macro-AUC, MLSOL is the second best performing sampling method in 2-fold cross-validation. We will supplement the experimental results for each dataset in the future.

Metric	Classifier	Base	$\text{MLSMOTE}_{\text{U}}$	$\text{MLSMOTE}_{\text{R}}$	MLSOL	MLROS	MLRUS	RHwRSMT	RHwRROS	LSDMLO
Macro-F	BR	0.2275(3)	0.2222(8)	0.2248(4)	0.2243(5)	0.2239(6)	0.2284(2)	0.1162(9)	0.2237(7)	0.2307(1)
	MLkNN	0.1376(7)	0.1795(4)	0.1847(3)	0.2242(2)	0.1673(5)	0.1361(8)	0.0973(9)	0.1673(6)	0.2376(1)
	CC	0.2182(6)	0.2152(8)	0.2203(3)	0.2217(2)	0.2185(4)	0.2173(7)	0.1442(9)	0.2185(5)	0.2236(1)
	RAkEL	0.2139(6)	0.2138(7)	0.2174(1)	0.2165(3)	0.2159(4)	0.2155(5)	0.1236(9)	0.2136(8)	0.2174(2)
	COCOA	0.2969(8)	0.3229(6)	0.3257(4)	0.3334(2)	0.3263(3)	0.2960(9)	0.3002(7)	0.3257(5)	0.3382(1)
	Avg	6.0	6.6	3	2.8	4.4	6.2	8.6	6.2	**1.2**
Macro-AUC	BR	0.6032(6)	0.6081(1)	0.6039(4)	0.6049(3)	0.6017(7)	0.6033(5)	0.5986(9)	0.6012(8)	0.6076(2)
	MLkNN	0.7137(5)	0.7370(1)	0.7247(4)	0.7301(2)	0.7056(7)	0.7125(6)	0.7015(9)	0.7056(8)	0.7286(3)
	CC	0.5994(7)	0.6031(3)	0.6026(4)	0.6042(2)	0.5995(5)	0.5988(8)	0.5738(9)	0.5995(6)	0.6075(1)
	RAkEL	0.5967(7)	0.6023(2)	0.5990(4)	0.6010(3)	0.5977(5)	0.5969(6)	0.5965(8)	0.5961(9)	0.6037(1)
	COCOA	0.8758(7)	0.9042(2)	0.8917(5)	0.8909(6)	0.8950(3)	0.8749(8)	0.8683(9)	0.8947(4)	0.9060(1)
	Ave(Total)	6.4	1.8	4.2	3.2	5.4	6.6	8.8	7	**1.6**
Macro-PR	BR	0.0992(3)	0.0952(9)	0.0982(4)	0.0973(8)	0.0978(6)	0.0996(2)	0.0982(5)	0.0976(7)	0.1020(1)
	MLkNN	0.1678(6)	0.1741(4)	0.1835(3)	0.1942(1)	0.1645(8)	0.1671(7)	0.1709(5)	0.1645(9)	0.1869(2)
	CC	0.0945(6)	0.0920(8)	0.0958(2)	0.0954(3)	0.0946(4)	0.0942(7)	0.0824(9)	0.0946(5)	0.0975(1)
	RAkEL	0.0934(7)	0.0911(9)	0.0952(3)	0.0938(6)	0.0946(4)	0.0945(5)	0.0962(2)	0.0927(8)	0.0994(1)
	COCOA	0.2903(7)	0.2990(6)	0.3005(2)	0.3003(3)	0.2997(4)	0.2893(8)	0.2794(9)	0.2997(5)	0.3014(1)
	Ave(Total)	5.8	7.2	2.8	4.2	5.2	5.8	6	6.8	**1.2**
RankLoss	BR	0.6352(4)	0.6300(2)	0.6380(6)	0.6347(3)	0.6392(7)	0.6376(5)	0.6470(9)	0.6392(8)	0.6231(1)
	MLkNN	0.1197(3)	0.1282(9)	0.1241(7)	0.1195(2)	0.1228(5)	0.1200(4)	0.1258(8)	0.1228(6)	0.1178(1)
	CC	0.6416(5)	0.6372(3)	0.6414(4)	0.6313(2)	0.6453(7)	0.6442(6)	0.7232(9)	0.6453(8)	0.6302(1)
	RAkEL	0.6433(5)	0.6371(1)	0.6454(8)	0.6403(3)	0.6432(4)	0.6437(6)	0.6510(9)	0.6449(7)	0.6378(2)
	COCOA	0.0747(8)	0.0685(6)	0.0687(7)	0.0605(1)	0.0671(3)	0.0752(9)	0.0678(5)	0.0674(4)	0.0624(2)
	Ave(Total)	5.0	4.2	6.4	2.2	5.2	6	8	6.6	**1.4**