MMEnsemble: Imbalanced Classification Framework using Metric Learning and Multi-sampling Ratio Ensemble

Takahiro Komamizu

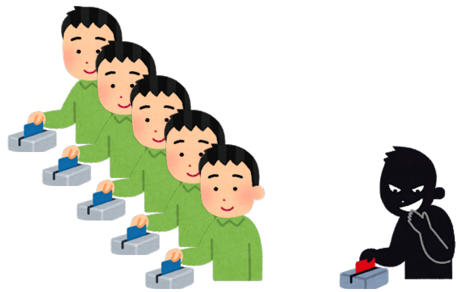
Nagoya University

Japan

This slide is downloadable from
http://taka-coma.pro/pdfs/DEXA2021.pdf

Class Imbalance is Universal Phenomenon

E-mail Spam



Credit Card Fraud

Driving Behavior

- Others
  - clinical domain [5], economic domain [25], agricultural domain [28], software engineering domain [26], computer network domain [11], etc.

Classifiers suffer from Class Imbalance

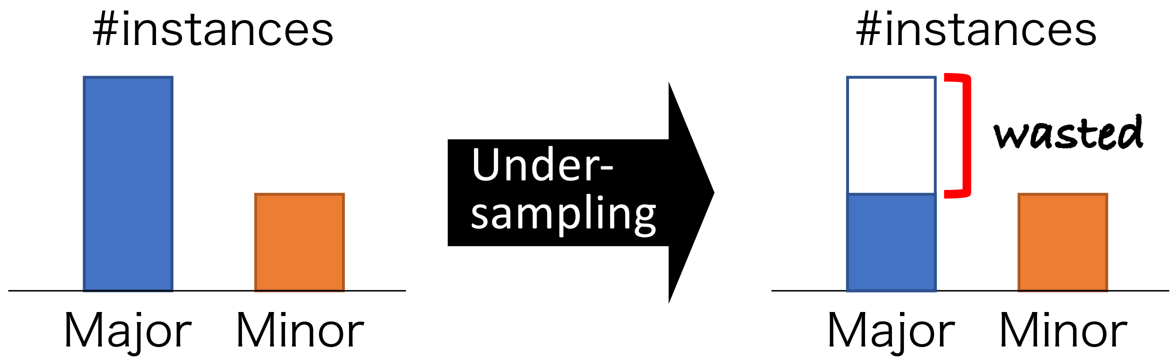
- Classifiers tend to prefer majority class
  - Choosing majority (say negative) class has more chance to increase accuracy score, beacuse $\#TN \gg \#TP$
    - $accuracy = \frac{\#TP + \#TN}{\#TP + \#TN + \#FP + \#FN}$
  - Consider 1 positive instance and 99 negative instances
    - All negative: accuracy = 99%
      - For classifiers, it looks (almost) optimal.
- In reality, minority class is more important.
  - What if your spam filter regards all mail as non-spam?
  - What if your fraud detector rageds all as normal action?

Two Major Approaches for Class Imbalance

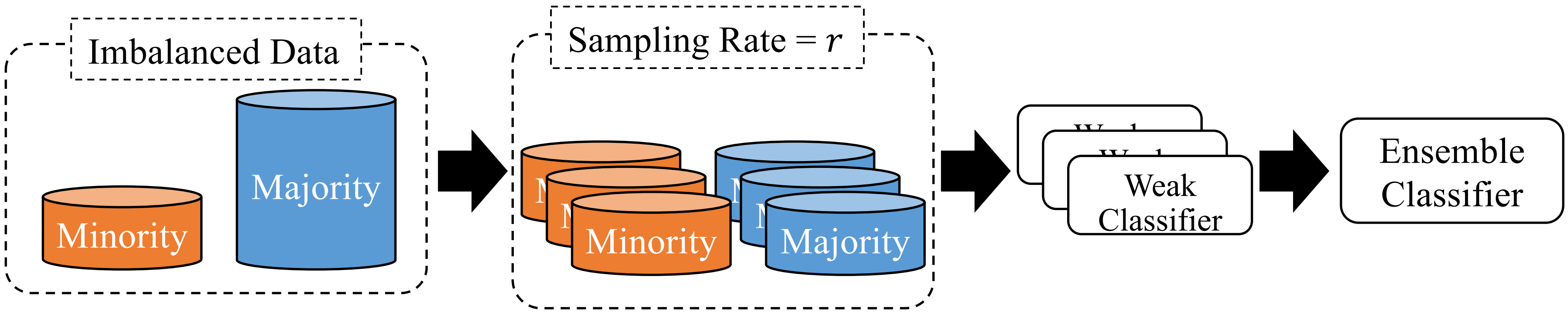
- Cost-sensitive learning approach
  - Desing cost function that gives higher penalty when classifiers fail to correctly classify the minority classes.
  - Depending on classification methods.
- Data-level approach
  - Add or remove data points so that instances of classes are balanced.
    - Adding: Oversampling / Synthetic oversampling (e.g., SMOTE, SWIM)
    - Removing: Undersampling (US)
  - NOT depending on classification methods.

EasyEnsemble (EE)[10]: ensemble multi samples

- Simple undersampling wastes major part of samples.



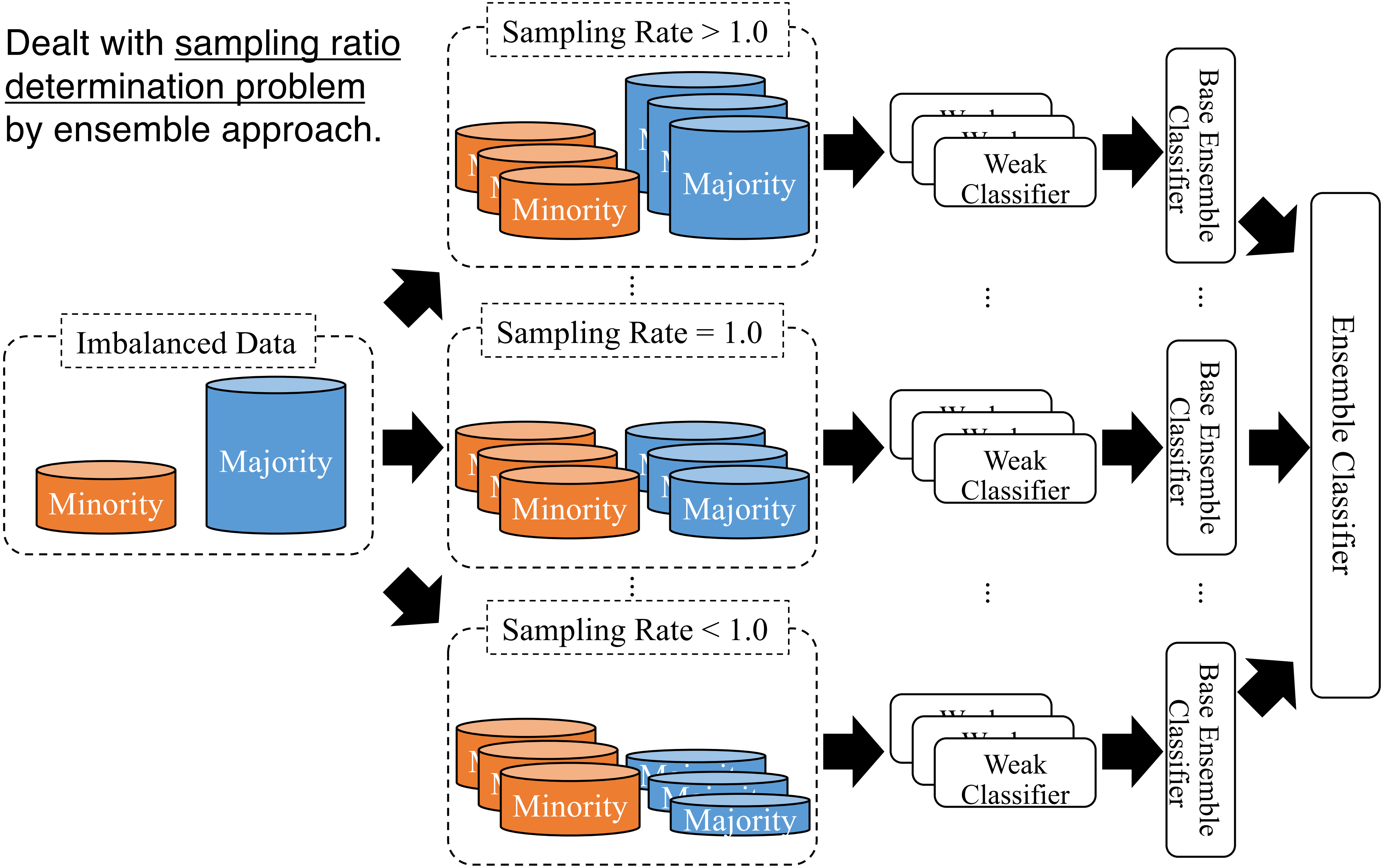
- EE samples multiple times so that most of samples are used in trianing an ensemble classifier.



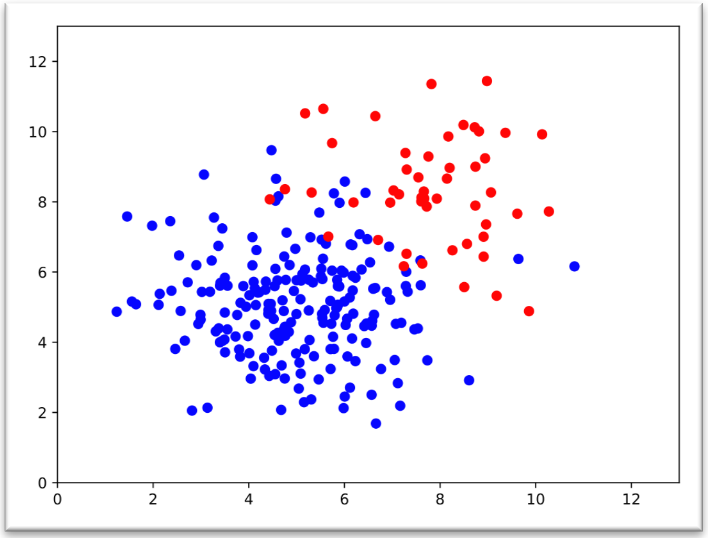
multiple sampling w/ replacement

MUEnsemble[8]: previous work

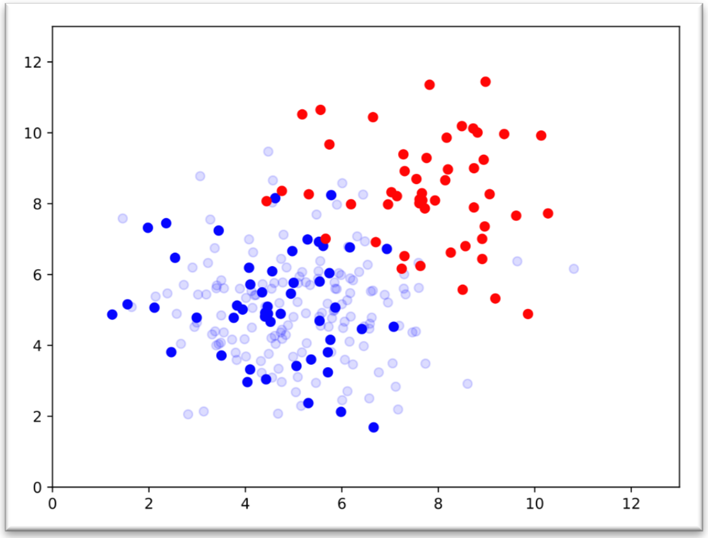
Dealt with sampling ratio determination problem by ensemble approach.

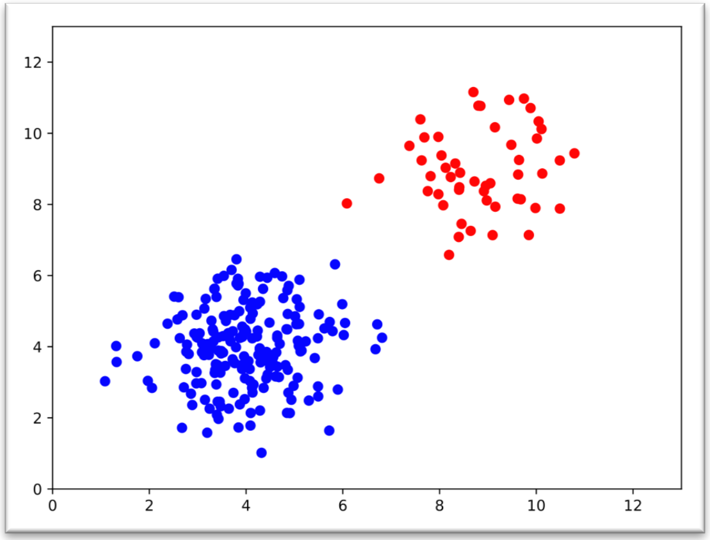


What about feature space?



US





<u>Metric Learning (ML) e.g., LMNN [19]</u>

Learning a transformation s.t.

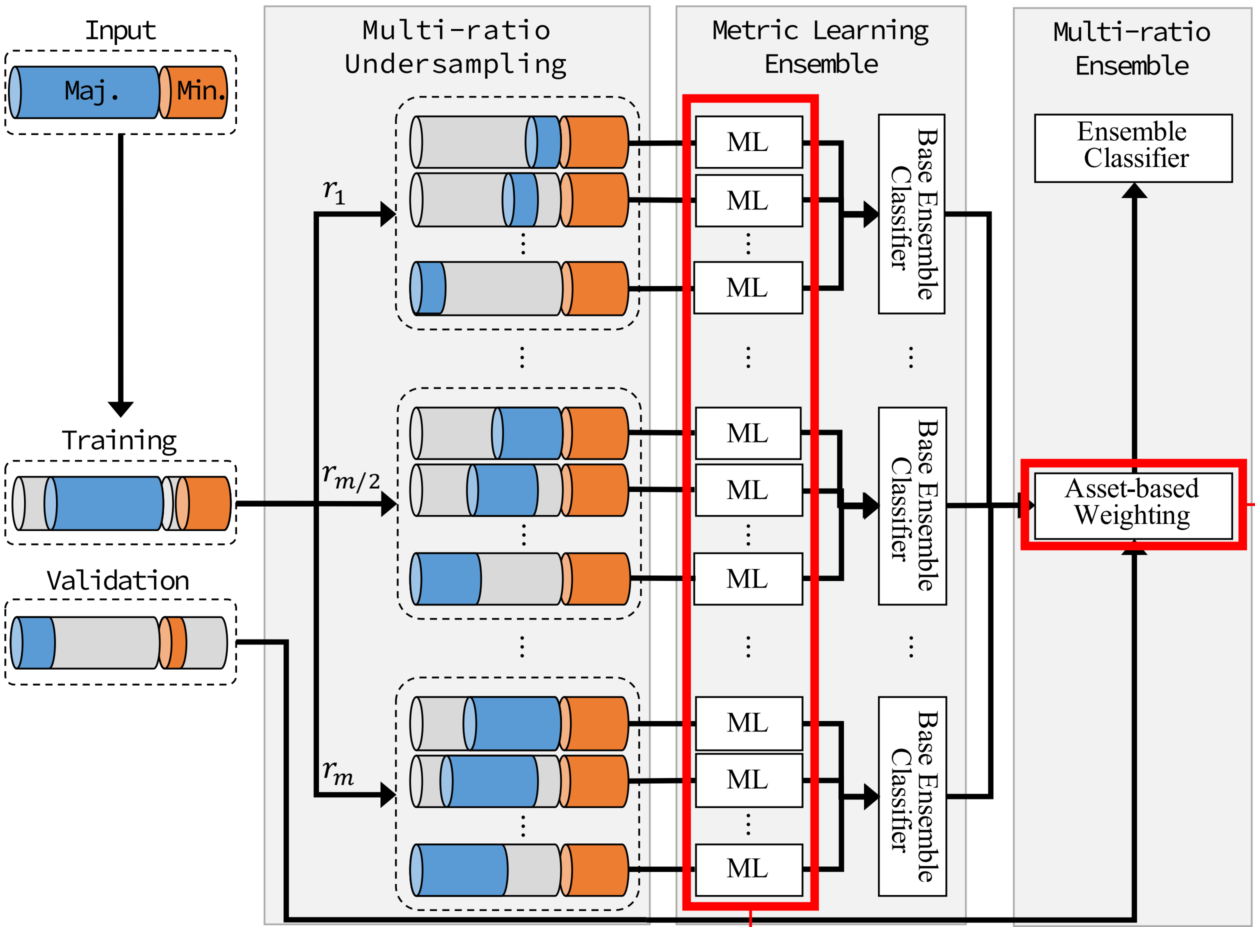
- samples of the same classes get closer,
- samples of the different classes get further

ML also suffers from the class imbalance.

➔ [18] shows US + ML improves classification performance in the class imbalance data.

MMEnsemble: ensemble multiple rates w/ ML

- Overall framework is based on MUEnsemble.



Weighting base classifiers based on *assets*.

ML is incorporated s.t. learning good feature spaces.

(for detail, refer to the paper)

Asset-based Weighting Scheme

- MUEnsemble [8] uses heuristic schemes.
- Idea: Taking classification performances of base classifiers into account
  - High weight for classifiers which can classify difficult samples
    - Easiness of sample i is measured by #classifiers (C_j) correctly classify.

$$T_i = \left|\{C_j \mid C_j \in C, C_j.predict(d_i) = \ell_i\right|$$

- Asset-based Weighting Scheme

Kronecker delta

Weight for difficult samples

$$W_{asset}(r) = \frac{1}{\sum_{r \in R} W_{asset}(r)} \cdot \sum_{(d_i, \ell_i) \in D^{(val)}} \delta\left(C_r.predict(d_i), \ell_i\right) \cdot T_i^{-k}$$

Difficulty is measured for validation set.

Research Questions in the expriment

- Q1: Does MMEnsemble outperform the state-of-the-art imbalanced classification methods of metric learning and undersampling?
- Q2: Is the combination of metric learning and multi-ratio ensemble effective?
- Q3-1: Does the asset-based weighting help improve the classification performance?
- Q3-2: and what is the effect of choice of its hyper-parameter k in the asset-based weighting?
    - Refer to the paper

Datasets from OpenML / KEEL Repositories

- Selection of the datasets is same as SOTA US+ML approach (DDAE) [20].

	ID	Name	#records	#minor	#dim	IR
OpenML	D1	cm1	498	49	21	9.2
	D2	kc3	458	43	39	9.7
	D3	mw1	403	31	37	12.0
	D4	pc1	1,109	77	21	13.4
	D5	pc3	1,563	160	37	8.8
	D6	pc4	1,458	178	37	7.2
KEEL	D7	yeast1-7	459	30	7	14.3
	D8	abalone9-18	731	42	8	16.4
	D9	yeast6	1,484	35	8	41.4
	D10	abalone19	4,174	32	8	129.4
	D11	wine3-5	691	10	11	68.1
	D12	abalone20	1,916	26	8	72.7

$$IR = \frac{\#major}{\#minor}$$

Comparison SOTA methods

- $Gm = \sqrt{TPR \cdot TNR}$: geometric mean of true positive rate and true negative rate
- F_2: recall-weighted f-measure

	ML				US+ML				Proposed			
Data	IML†				DDAE†				MMEnsemble			
	Rec	*Gm*	F_2	*AUC*	*Rec*	*Gm*	F_2	*AUC*	*Rec*	*Gm*	F_2	*AUC*
D1	.313	.520	.287	.589	.813	**.775**	**.580**	.776	**.863**	.756	.546	**.819**
D2	.692	.805	**.652**	.814	.846	**.823**	.625	.823	**.952**	.750	.534	**.868**
D3	.500	.635	.345	.653	.750	**.815**	**.588**	.817	**.793**	.772	.528	**.866**
D4	.852	.657	.408	.679	**.963**	**.819**	**.573**	.830	.944	**.819**	.548	**.895**
D5	.510	.578	.342	.582	.735	.743	.536	.744	**.867**	**.794**	**.598**	**.854**
D6	.814	.725	.574	.730	.932	.804	.676	.813	**.963**	**.873**	**.748**	**.934**
D7	.667	.716	.471	.718	.833	**.841**	**.649**	.841	**.933**	.808	.512	**.883**
D8	.600	.709	.375	.719	.700	.814	.603	.824	**.886**	**.877**	**.650**	**.941**
D9	.700	.798	.407	.805	.900	.883	.421	.883	**.931**	**.920**	**.585**	**.976**
D10	.667	.626	.037	.628	**1.000**	**.839**	.075	.852	.935	.835	**.128**	**.876**
D11	.000	.000	NA	.500	.333	.550	.156	.620	**.894**	**.842**	**.188**	**.939**
D12	.800	.802	.252	.802	**1.000**	**.964**	**.556**	.965	.992	.943	.451	**.982**

MMEnsemble achieves best in Rec and AUC.
Gm and F_2 are comparable, because precision is a little sacrificed.

Is the combination of ML + MR effective?

(MR = multi-ratio ensemble)

MMEnsemble – MR MMEnsemble – ML

Data	MLEnsemble (EE + ML)				MUEnsemble (EE + MR)				MMEnsemble (EE + ML + MR)			
	Rec	*Gm*	F_2	*AUC*	*Rec*	*Gm*	F_2	*AUC*	*Rec*	*Gm*	F_2	*AUC*
D1	.751	.695	.475	.754	.812	.698	**.484**	**.783**	**.820**	**.699**	.483	**.783**
D2	.854	**.742**	**.518**	.831	.821	.718	.490	.826	**.891**	.731	.509	**.862**
D3	.790	.720	.461	.817	.761	.700	.439	.820	**.864**	**.761**	**.506**	**.860**
D4	.875	.804	.533	.871	**.880**	.788	.509	.860	.873	**.816**	**.548**	**.885**
D5	.821	.760	.554	.821	.828	.753	.546	.828	**.844**	**.781**	**.581**	**.837**
D6	.921	.844	.707	.907	.946	**.883**	**.764**	**.934**	**.971**	.873	.747	.921
D7	.787	.746	**.444**	.830	.792	.743	.438	.818	**.860**	**.749**	**.444**	**.859**
D8	.835	.822	**.537**	.913	.769	.757	.440	.840	**.911**	**.835**	.531	**.959**
D9	**.893**	.874	.438	.951	.850	.857	.427	.935	.885	**.890**	**.508**	**.973**
D10	.835	.762	.101	.828	.911	.770	.096	.834	**.999**	**.828**	**.112**	**.887**
D11	.735	.697	.144	.797	**.785**	**.753**	**.178**	**.841**	.765	.724	.160	.795
D12	.882	.875	.330	.951	.870	.840	.248	.931	**.987**	**.923**	**.363**	**.985**

Comparable

Almost best

MMEnsemble can take both advantages of MR and ML

Weighting Scheme Comparison

- Uniform: baselins
- Gauss: Best weighting in MUEnsemble[8] (for detail, refer to the paper)

Data	Uniform				Gauss				Asset			
	Rec	*Gm*	F_2	*AUC*	*Rec*	*Gm*	F_2	*AUC*	*Rec*	*Gm*	F_2	*AUC*
D1	**.893**	.637	.456	.781	.820	.699	.483	.783	.863	**.756**	**.546**	**.819**
D2	.950	.711	.502	.818	.891	.731	.509	.862	**.952**	**.750**	**.534**	**.868**
D3	.813	.692	.435	.815	**.864**	.761	.506	.860	.793	**.772**	**.528**	**.866**
D4	**.954**	.788	.505	.891	.873	.816	**.548**	.885	.944	**.819**	**.548**	**.895**
D5	**.923**	.748	.550	.840	.844	.781	.581	.837	.867	**.794**	**.598**	**.854**
D6	**.972**	.846	.710	.925	.971	**.873**	.747	.921	.963	**.873**	**.748**	**.934**
D7	.915	.742	.432	.882	.860	.749	.444	.859	**.933**	**.808**	**.512**	**.883**
D8	.900	.817	.509	.931	**.911**	.835	.531	**.959**	.886	**.877**	**.650**	.941
D9	.910	.872	.413	.954	.885	.890	.508	.973	**.931**	**.920**	**.585**	**.976**
D10	.924	.758	.091	.837	**.999**	.828	.112	**.887**	.935	**.835**	**.128**	.876
D11	.633	.666	.152	.810	.765	.724	.160	.795	**.894**	**.842**	**.188**	**.939**
D12	.873	.858	.303	.953	.987	.923	.363	**.985**	**.992**	**.943**	**.451**	.982

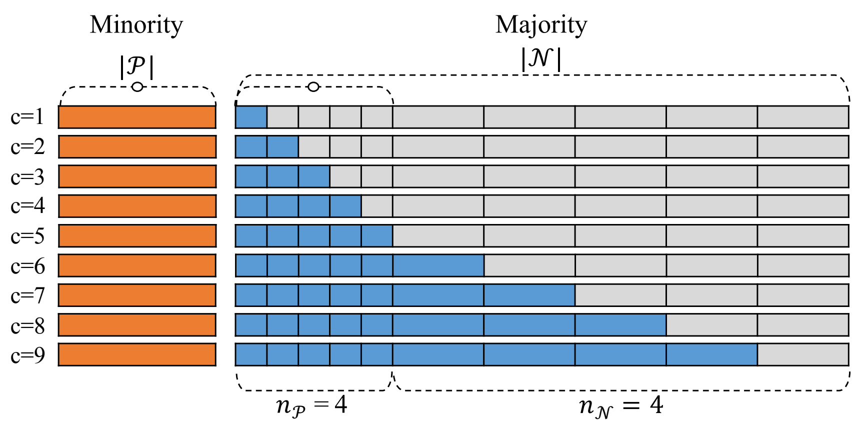
Asset-based weighting performs almost best,
but underperforms in Rec due to precision-recall trade-off.

Conclusion and Future Directions

- Conclusion
  - MMEnsemble: an ensemble framework using multi-ratio ensemble (MR) and metric learning (ML).
  - Asset-based weighting scheme for multiple ratios.
  - MMEnsemble outperforms SOTA methods.
- Future directions
  - Improvement on computational efficiency
  - Class imbalance problem in deep learning models

Rate Enumeration and Weighting Scheme

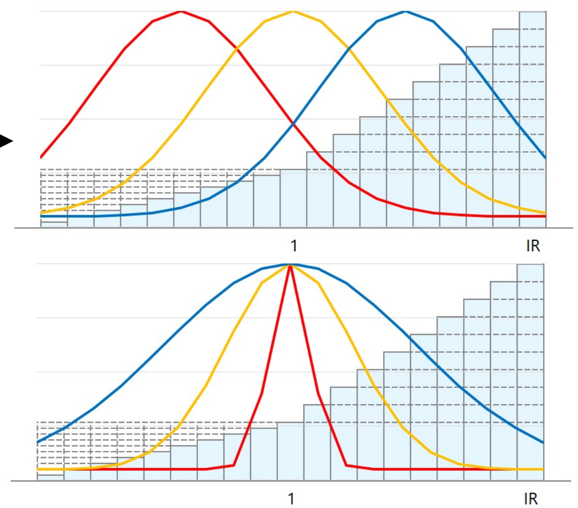
- Automatic rate enumeration:
  - Possible rates differ due to various IR on datasets



- Weighting scheme: control #base classifiers on rates
  - Find well-balanced combination of rates
  - Gaussian

$$W_{gauss}(r) = \frac{1}{\sum_{r \in R} W_{gauss}(r)} \cdot \exp\left(-\frac{(r-\mu)^2}{2\sigma^2}\right)$$

- μ and σ^2 are detemined by grid search.



Sampling rate