MUEnsemble: Multi-ratio Undersampling-based Ensemble Framework for Imbalanced Data

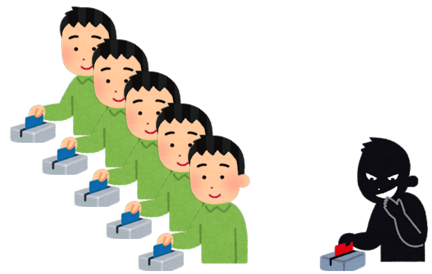
Takahiro Komamizu, Risa Uehara, Yasuhiro Ogawa, Katsuhiko Toyama

Nagoya University

Japan

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.

Imbalance Ratio: $IR = \#major/\#minor$

Table 1: Classification Datasets

Source	ID	Dataset (binary classes if multi-class)	#dim.	#major	#minor	IR
UCI repos	D1	Abalone (9 v. 18)	8	689	42	16.4
	D2	Anuran Calls (Lept. v. Bufo.)	22	4,420	68	65.0
	D3	Covertype (2 v. 5)	54	283,301	9,493	29.8
	D4	default of credit card clients	23	23,364	6,636	3.5
	D5	HTRU2	8	16,259	1,639	9.9
	D6	Online Shoppers Purchasing Intention	18	10,422	1,908	5.5
	D7	Polish companies bankruptcy	64	41,314	2,091	19.8
	D8	Spambase	56	2,788	1,813	1.5
	D9	Wine Quality – Red ((3, 4) v. others)	11	1,536	63	24.4
	D10	Wine Quality – White (7 v. 3)	11	880	20	44.0
Kaggle Dataset	D11	Churn Modelling	9	7,963	2,037	3.9
	D12	Credit Card Fraud Detection	30	284,315	492	577.9
	D13	ECG Heartbeat – Arrhythmia (N v. F)	187	90,589	803	112.8
	D14	Financial Distress	85	3,536	136	26.0
	D15	LoanDefault LTFS AV	39	182,543	50,611	3.6
	D16	Mafalda Opel – Driving Style	14	9,530	2,190	4.4
	D17	Mafalda Peugeot – Driving Style	14	12,559	678	18.5
	D18	Rain in Australia	20	110,316	31,877	3.5
	D19	Surgical	24	10,945	3,690	3.0

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 >> 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?

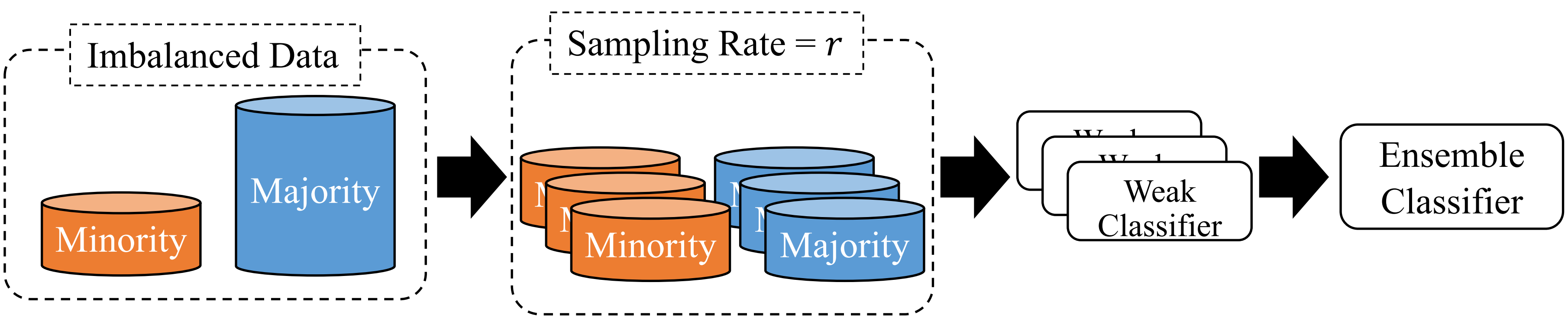
Two Major Approaches for Class Imbalance

- Cost-sensitive learning approach
  - Desing cost function that gives higher penalty when classifiers fail to correctly classify the minority classes.
  - Dependent 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
  - NOT dependent on classification methods.

EasyEnsemble (EE)[19]: ensemble multi samples

- Simple undersampling wastes major part of samples.

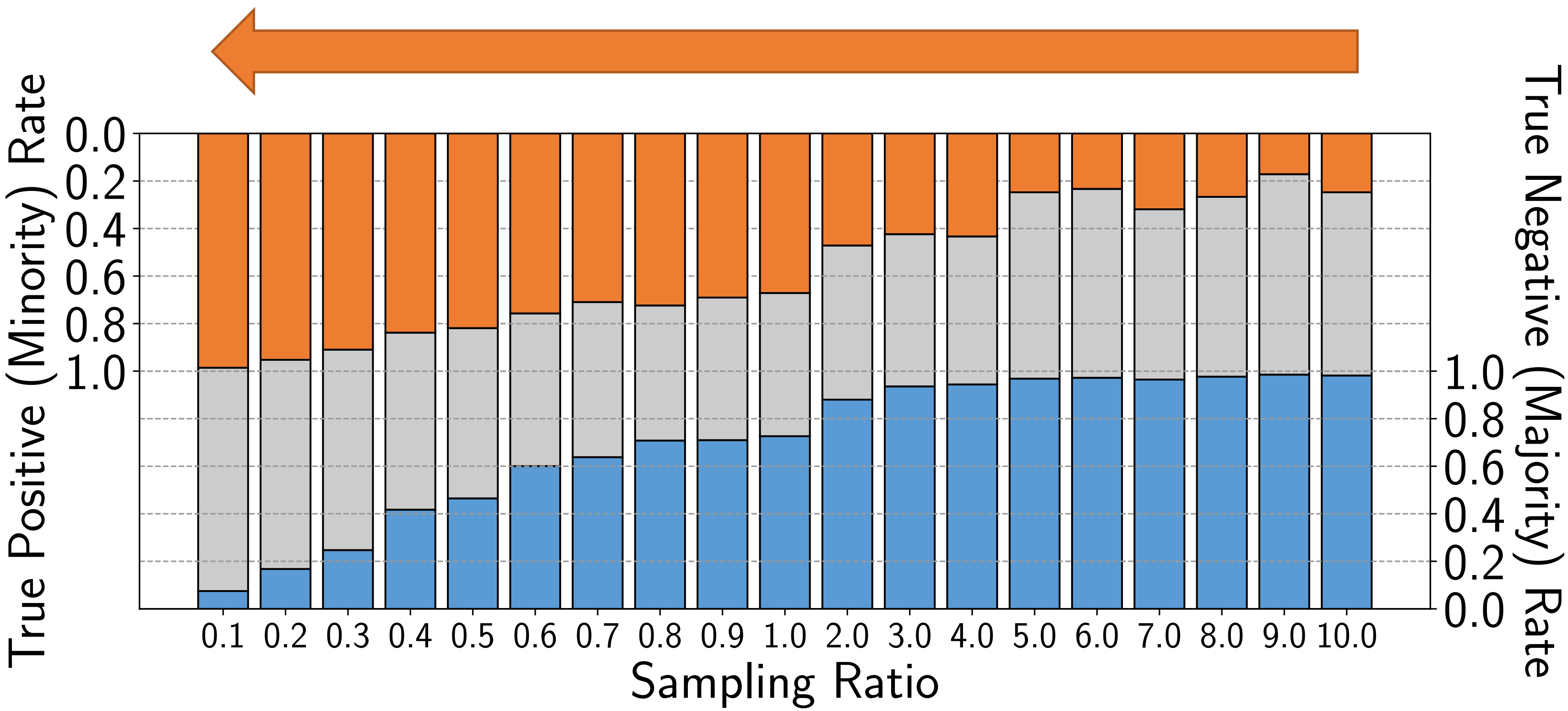
- EE samples multiple times so that most of samples are used in trianing and ensembles classifiers.



multiple sampling w/ replacement

How can we find "good" sampling ratio?

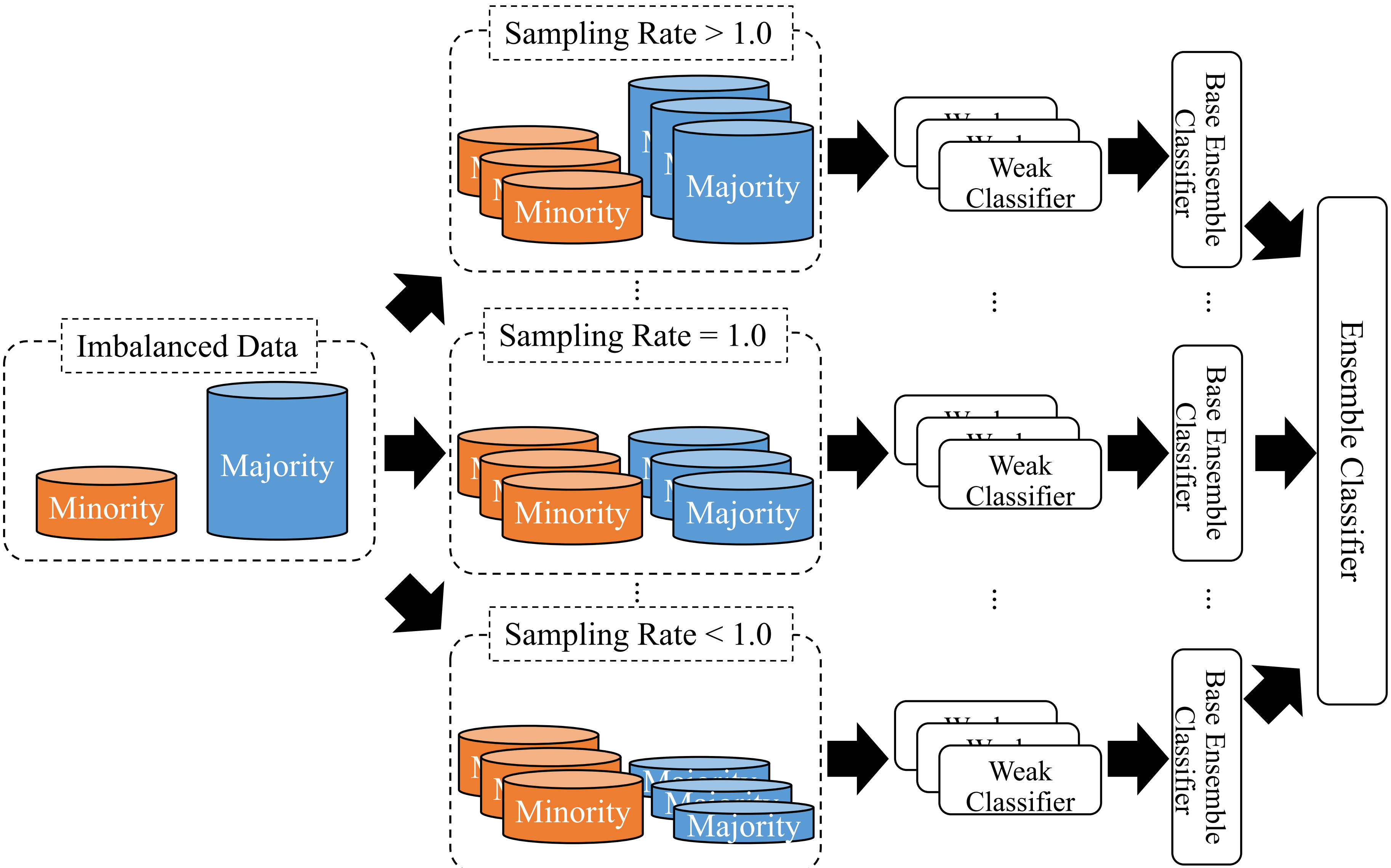
the smaller, classification accuray on **the minority** increases.



(the imbalance ratio after sampling)

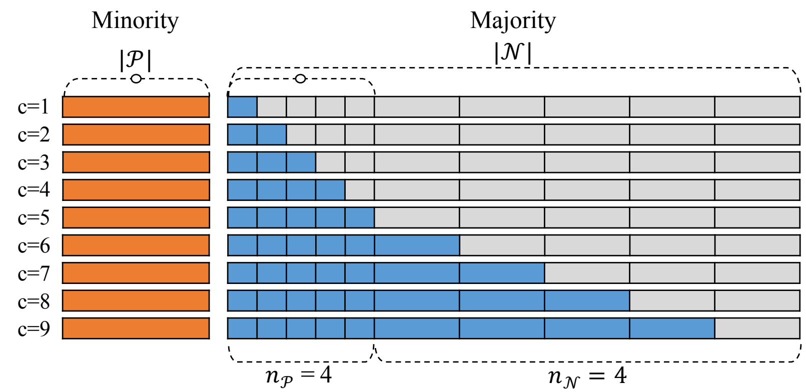
the larger, classification accuray on **the majority** increases.

MUEnsemble: ensemble multiple rates



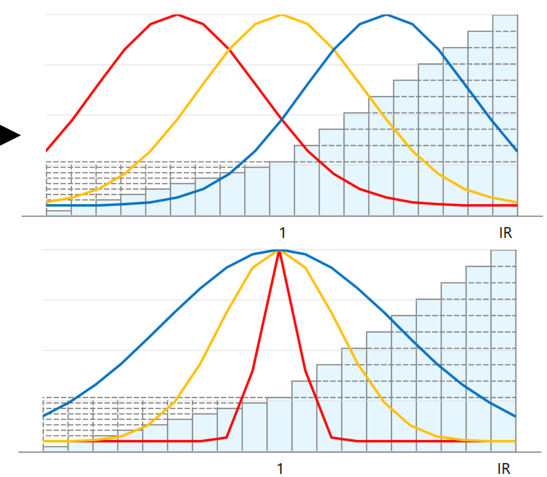
Rate Enumeration and Weighting Scheme

- Automatic rate enumeration:
  - Possible rates differ due to various IR on datasets
  - Excessive undersampling
    - sampling rate below 1



- Weighting scheme: control #base classifiers on rates
  - Constant
  - Concave
  - Convex

  refer to the paper
  - Gaussian $$B_{gauss}(c) = \left\lfloor a_g \cdot exp\left(\frac{(c-\mu)^2}{2\sigma^2}\right)\right\rfloor$$
    - μ and σ^2 are detemined by grid search.



Research Questions in the expriment

- Q1: Does excessive undersampling have a positive effect?
  — Yes.
- Q2: What is a good strategy for the weighting scheme?
  — Gaussian is the best.
- Q3: Does the parameter estimation find optimal parameters?
  — Mostly yes. In some datasets, not optimal but nearly optimal parameters are found.
- Q4: Does MUEnsemble outperform baseline methods?

refer to the paper

Comparison w/ baseline methods

Table 7: Comparison over Baselines. The best scores are boldfaced.

Dataset	ORG	Oversampling			Undersampling			MUEnsemble	
		SMT	ADA	SWIM	RUS	RBST	EE	Gauss	(optimal)
D1	.580	.675	.671	.642	.670	.577	.741	**.753**	(.772)
D2	.915	.931	.897	.909	.925	.954	.963	**.971**	(.971)
D3	.891	.924	.916	.747	**.928**	.852	.798	.808	(.809)
D4	.581	.585	.584	.580	.616	.528	.689	**.701**	(.701)
D5	.896	.910	.908	.906	.907	.897	.930	**.936**	(.936)
D6	.713	.733	.739	.709	.790	.731	.845	**.849**	(.849)
D7	.810	.829	.834	.760	.854	.786	**.908**	**.908**	(.910)
D8	.900	.900	.898	.896	.896	**.931**	.916	.919	(.919)
D9	.420	.467	.473	.519	.624	.436	.680	**.705**	(.705)
D10	.475	.444	.574	.666	.616	.412	.662	**.735**	(.735)
D11	.642	.652	.647	.642	.678	.619	.761	**.762**	(.762)
D12	.876	.877	.865	.917	.905	.895	.937	**.938**	(.939)
D13	.822	.859	.853	.829	.883	.831	.895	**.900**	(.900)
D14	.546	.548	.576	.562	.775	.606	**.863**	.862	(.865)
D15	.466	.474	.476	.442	.538	.463	.592	**.593**	(.593)
D16	.708	.755	.737	.724	**.794**	.702	.779	.789	(.789)
D17	.760	.780	.771	.757	.770	.747	.710	**.791**	(.791)
D18	.677	.690	.689	.678	.714	.641	.762	**.767**	(.767)
D19	**.803**	.787	.760	**.803**	.785	.761	.760	**.803**	(.803)
Avg.	.710	.727	.730	.720	.772	.704	.800	**.815**	(.817)
Ranks	6.1	4.3	5.0	5.8	3.5	6.2	2.8	**1.4**	(–)

Baselines

Oversampling
- SMT: SMOTE [7]
- ADA: ADASYN [13]
- SWIM: SWIM [4]

Undersampling
- RUS: random US
- RBST: RUSBoost [27]
- EE: EasyEnsemble [19]

Metric

gmean: geometric mean of TPR and TNR

$$gmeam = \sqrt{TPR \cdot TNR}$$

Result

MUEnsemble is the best in 15 out of 19 datasets

Summary of Experiment

- Q1: Does excessive undersampling have a positive effect?
  — Yes.
- Q2: What is a good strategy for the weighting scheme?
  — Gaussian is the best.
- Q3: Does the parameter estimation find optimal parameters?
  — Mostly yes. In some datasets, not optimal but nearly optimal parameters are found.
- Q4: Does MUEnsemble outperform baseline methods?
  — MUEnsemble is the best in15 out of 19 datasets.

Conclusion and Future Directions

- Conclusion
  - [Proposal] MUEnsemble is a multi-ratio undersampling-based ensemble framework.
    - Excessive undersampling, Gaussian-based weighting function
  - [Result] It outperformds basedline methods.
  - [Limitation] It is costly due to the heavy ensemble structure.
- Future directions
  - Find the trade-off between exec. time and accuracy.
  - Apply to deep learning-based classification methods.
    - Soft and repetitive undersampling*

*T. Yamakoshi, T. Komamizu, Y. Ogawa, K. Toyama,
"Japanese Mistakable Legal Term Correction using Infrequency-aware BERT Classifier",
Transactions of the Japanese Society for Artificial Intelligence, Vol. 35, Iss. 4, pp.E-K25_1-17, 2020

Answers to Research Questions (Q1, Q2)

- Q1: Does excessive undersampling have a positive effect?

  — Yes.

Table 2: Effect of Excessive Undersampling. The best scores are boldfaced.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19
w/o excesive US	.616	.908	.425	.624	.910	.784	.844	.916	.505	.419	.711	.855	.688	.703	.207	.461	.523	.721	.762
w/ excesive US	**.732**	**.956**	**.778**	**.694**	**.931**	**.844**	**.907**	**.917**	**.643**	**.664**	**.761**	**.939**	**.896**	**.859**	**.592**	**.771**	**.712**	**.765**	**.767**

- Q2: What is a good strategy for the weighting scheme?

  — Gaussian is the best.

Table 3: Comparison of Balancing Functions. The best scores are boldfaced.

Func.	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	Avg.	Rank
Cns	.732	.956	.778	.694	.931	.844	.907	.917	.643	.664	.761	**.939**	.896	.859	.592	.771	.712	.765	.767	.796	2.9
Cnv	.674	.955	.779	**.700**	.934	.846	.904	.914	.549	.689	.758	.938	.897	.847	.577	.786	.713	.766	.760	.789	3.0
Cnc	.751	.956	.777	.689	.930	.843	.906	**.919**	.685	.665	**.762**	.938	.897	.858	.591	.746	.708	.764	.770	.798	3.0
Gauss	**.753**	**.971**	**.808**	**.700**	**.936**	**.849**	**.908**	**.919**	**.705**	**.700**	**.762**	**.939**	**.900**	**.862**	**.593**	**.789**	**.791**	**.767**	**.803**	**.813**	**1.0**

Answers to Research Questions (Q3)

- Q3: Does the parameter estimation find optimal parameters?
  — Mostly yes. In some datasets, not optimal but nearly optimal parameters are found.

Table 4: Effect of Optimization. The differences larger than 0 are boldfaced.

Method	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19
Estimated	.753	.971	.808	.701	.936	.849	.908	.919	.705	.735	.762	.938	.900	.862	.593	.789	.791	.767	.803
Optimal	.772	.971	.809	.701	.936	.849	.910	.919	.705	.735	.762	.939	.900	.865	.593	.789	.791	.767	.803
Diff	**.019**	.000	**.001**	.000	.000	.000	**.002**	.000	.000	.000	.000	**.001**	.000	**.003**	.000	.000	.000	.000	.000

Table 5: Estimated and Optimal Parameters (μ and σ^2). The estimated parameters equal to the optimal parameters are boldfaced.

Method	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Estimated	(8, 2)	(10, 50)	**(6, 50)**	**(6, $\frac{1}{8}$)**	(8, 30)	(4, 50)	(8, 10)	(10, $\frac{1}{8}$)	**(12, 30)**	**(8, $\frac{1}{8}$)**
Optimal	(10, 2)	(4, 50)	**(6, 50)**	**(6, $\frac{1}{8}$)**	(6, 30)	(6, 30)	(8, $\frac{1}{2}$)	(14, $\frac{1}{8}$)	**(12, 30)**	**(8, $\frac{1}{8}$)**

Method	D11	D12	D13	D14	D15	D16	D17	D18	D19
Estimated	**(10, 5)**	(8, 20)	**(8, 30)**	(10, 1)	**(10, 50)**	**(10, $\frac{1}{8}$)**	**(8, 50)**	**(8, 50)**	**(14, 50)**
Optimal	**(10, 5)**	(8, 30)	**(8, 30)**	(10, $\frac{1}{2}$)	**(10, 50)**	**(10, $\frac{1}{8}$)**	**(8, 50)**	**(8, 50)**	**(14, 50)**

Why not precision, recall or F1, but gmean?

- The weight of TP is imbalanced between precision and recall.
  - Precision tends to be small because FP can be large.
  - Recall tends to be large because its denominator $TP + TN$ is very small.
- gmean is more robust than others in the imbalanced classificatin scenario [17].
  - Different datasets have different $TP + TN$, recall can be easily varied.
- Review of precision, recall and F1 score
  - Precision: $precision = \frac{TP}{TP + FP}$
  - Recall: $recall = \frac{TP}{TP + FN}$
  - F1 score: $f1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$