SQUARE: A Benchmark for Research on Computing Crowd Consensus

Presentation by

Srikanth Muralidharan

Case Study – Brexit

- Referendum to leave/stay
- 17.4 M wanted to leave
- 16.1 M wanted to stay back
- Result: By Majority vote, ‘leave’ by margin of 1.3 M

Source: Independent news article

Case Study – Brexit

INDEPENDENT News Voices Culture Lifestyle Tech Sport Daily Edition

News › UK › UK Politics

Brexit research suggests 1.2 million Leave voters regret their choice in reversal that could change result

The research suggests that if a second referendum were held, the vote would be much closer

Lizzie Dearden | @lizziedearden | Friday 1 July 2016 | 735 comments

[Source: Independent news article]

REMEMBER MARGIN WAS 1.3 M!

Case Study – Brexit

[Source: Fortune]

MAJORITY VOTE CONSENSUS DOESN'T SEEM OPTIMAL HERE!

Take Away’s

- Getting Accurate Data label is a challenging process

- Possible solution: Get labels from multiple workers for a given data, combine them

Take Away’s

- MV might not be optimal method for combining.

- Better consensus methods might consider worker (voter) expertise / reliability and so on.

Problem

- Multiple consensus methods exist in literature
- Compare and contrast between different consensus methods
- Design rules about when to use a consensus method

Contributions

- SQUARE Benchmark
  - “(**Statistical Quality Assurance Robustness Evaluation**”
- Collection of diverse datasets, consensus methods
- Simulation of varying noise, supervision
- Heuristics for picking right consensus method given the scenario

Outline

- Benchmark Datasets for comparisons
- Consensus methods
- Experiment Setup and Results
- Concluding Remarks

- Benchmark Datasets for comparisons
- Consensus methods
- Experiment Setup and Results
- Concluding Remarks

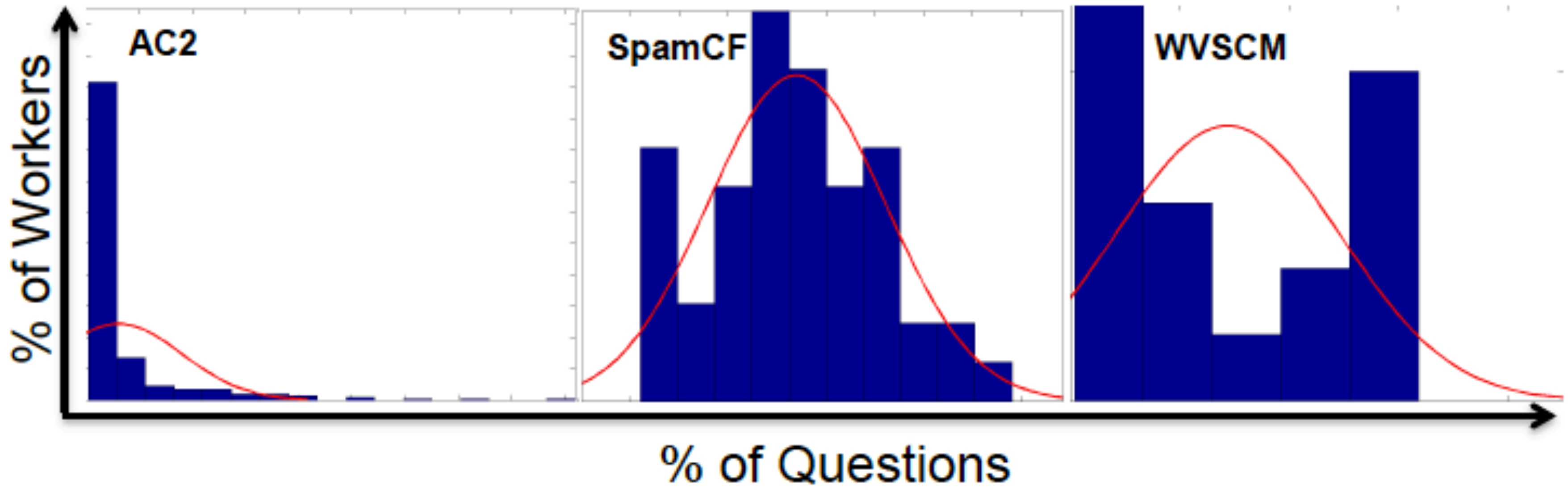
Datasets

Dataset	Categories	Examples	Workers	Labels	MV *Acc.*
AC2	4	333	269	3317	88.1
BM	2	1000	83	5000	69.6
HC	3	3275	722	18479	64.9
HCB	2	3275	722	18479	64.8
RTE	2	800	164	8000	91.9
SpamCF	2	100	150	2297	66.0
TEMP	2	462	76	4620	93.9
WB	2	108	39	4212	75.9
WSD	3	177	34	1770	99.6
WVSCM	2	159	17	1221	72.3

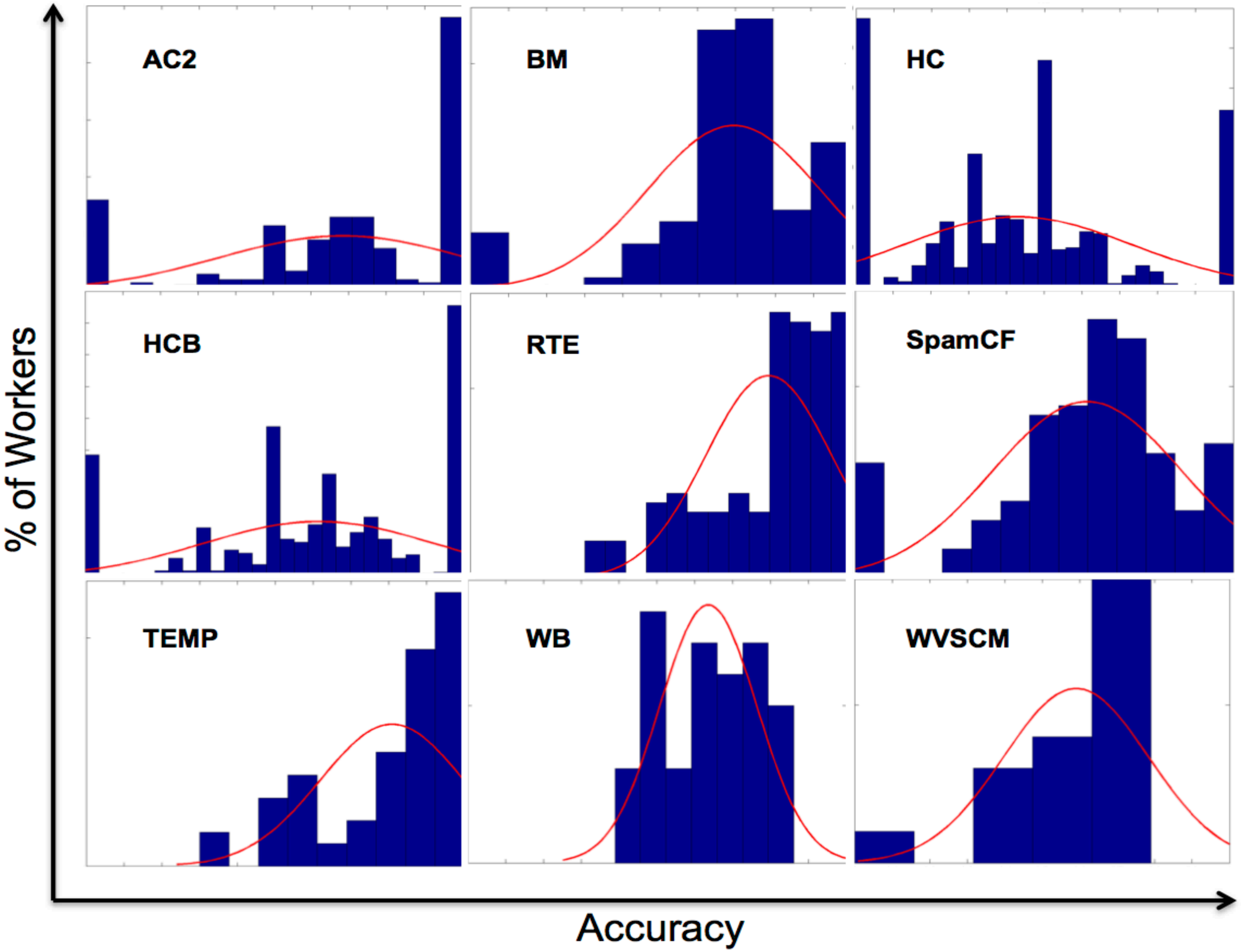
Datasets

Dataset	**Categories**	**Examples**	**Workers**	**Labels**	**MV *Acc.***
AC2	4	333	269	3317	88.1
BM	2	1000	83	5000	69.6
HC	3	3275	722	18479	64.9
HCB	2	3275	722	18479	64.8
RTE	2	800	164	8000	91.9
SpamCF	2	100	150	2297	66.0
TEMP	2	462	76	4620	93.9
WB	2	108	39	4212	75.9
WSD	3	177	34	1770	99.6
WVSCM	2	159	17	1221	72.3

Labels per worker count

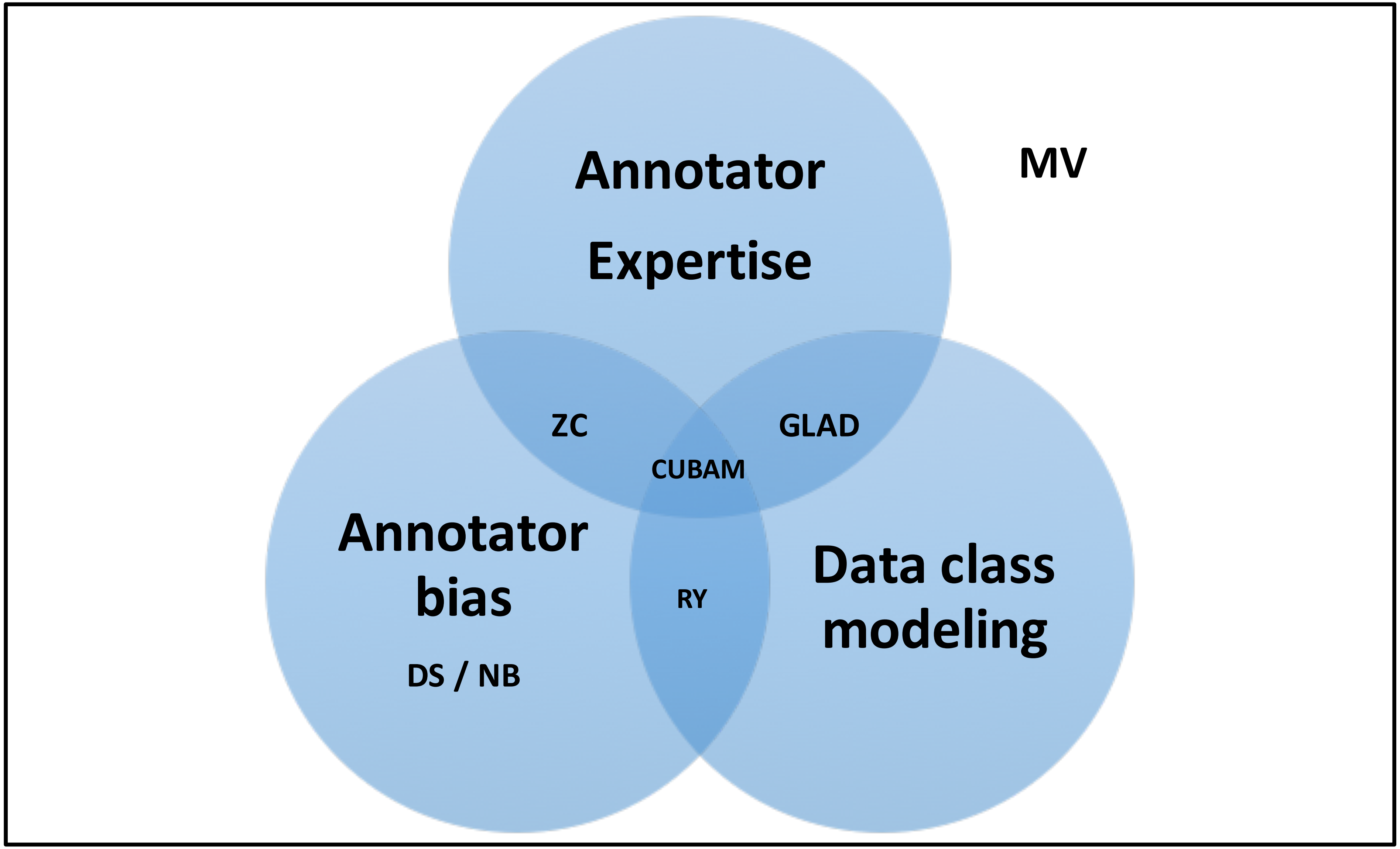


Worker accuracy distribution



- Benchmark Datasets for comparisons
- Consensus methods
- Experiment Setup and Results
- Concluding Remarks

Modelling considerations



Consensus methods – An overview

- Many methods exist for consensus

- Vast variations present between each methods

Key sources of Diversity

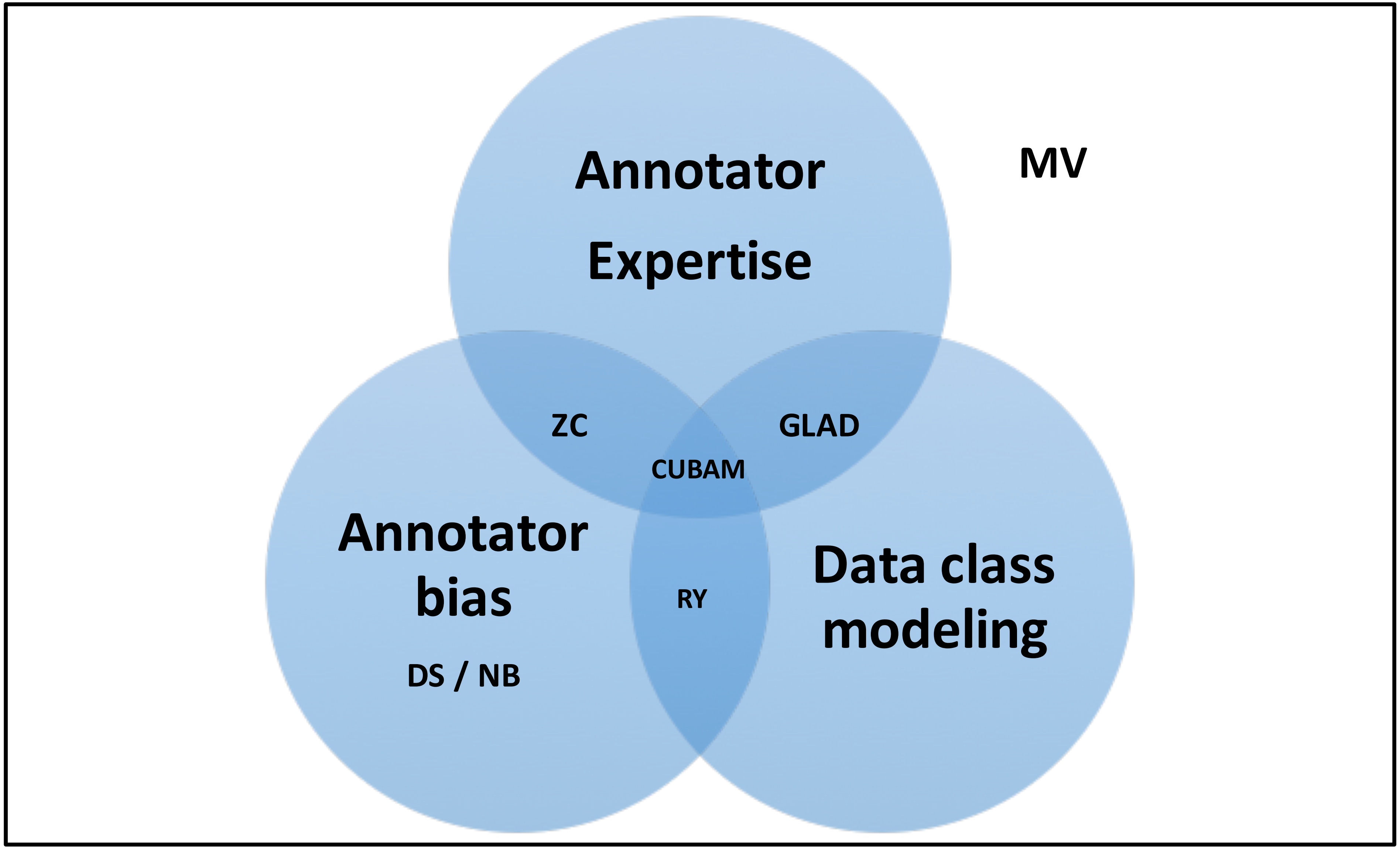
- Modelling of Worker Behaviour

- Modelling of class label distribution

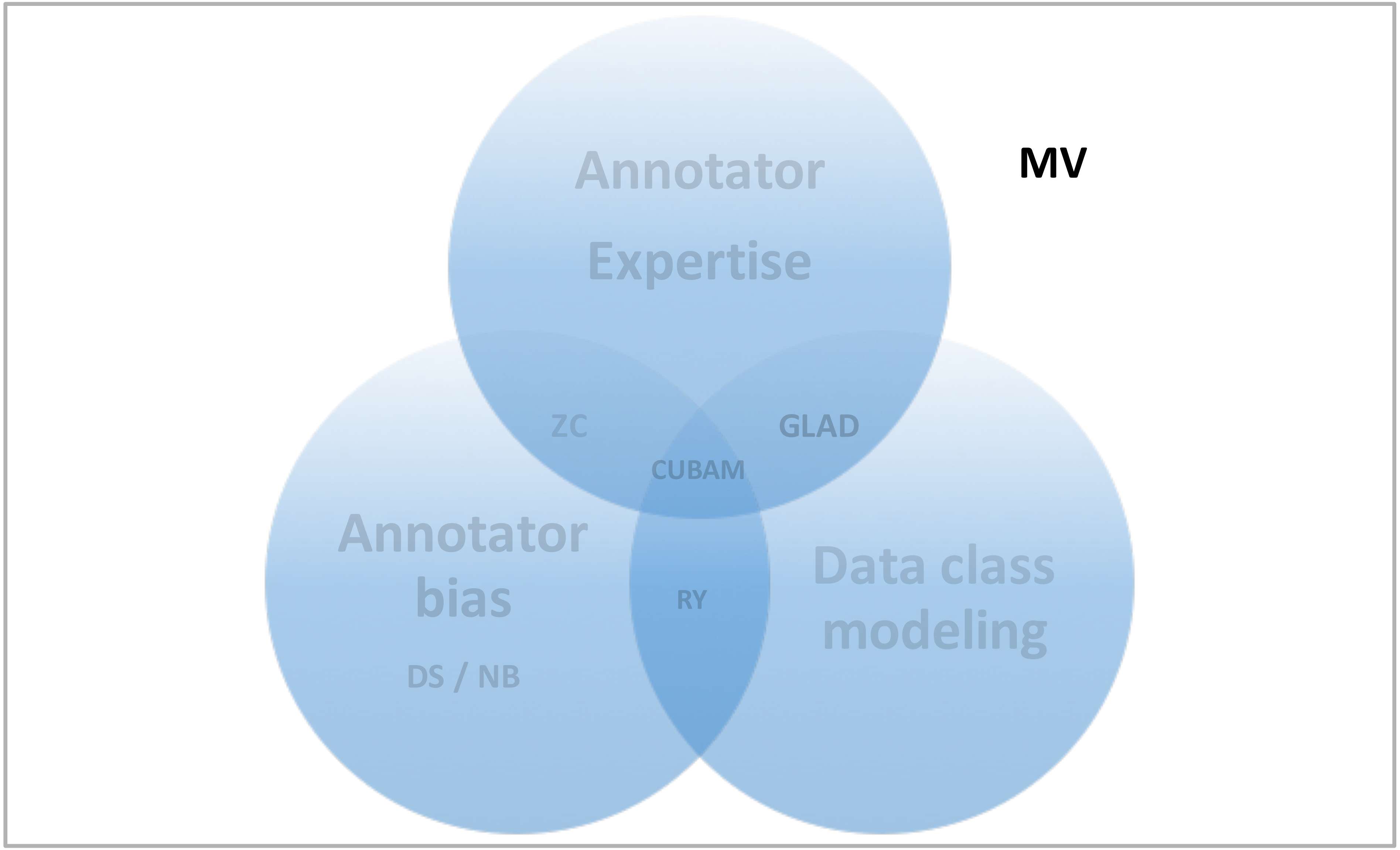
Consensus methods used in our experiments

- Majority voting (MV)
- ZenCrowd (ZC) [Demartini et al. 2012]
- David and Skene (DS) & Naïve Bayes (NB)
- GLAD [Whitehill et al. 2009]
- RY [Raykar et al. 2010]
- CUBAM [Welinder et al. 2010]

Modelling considerations



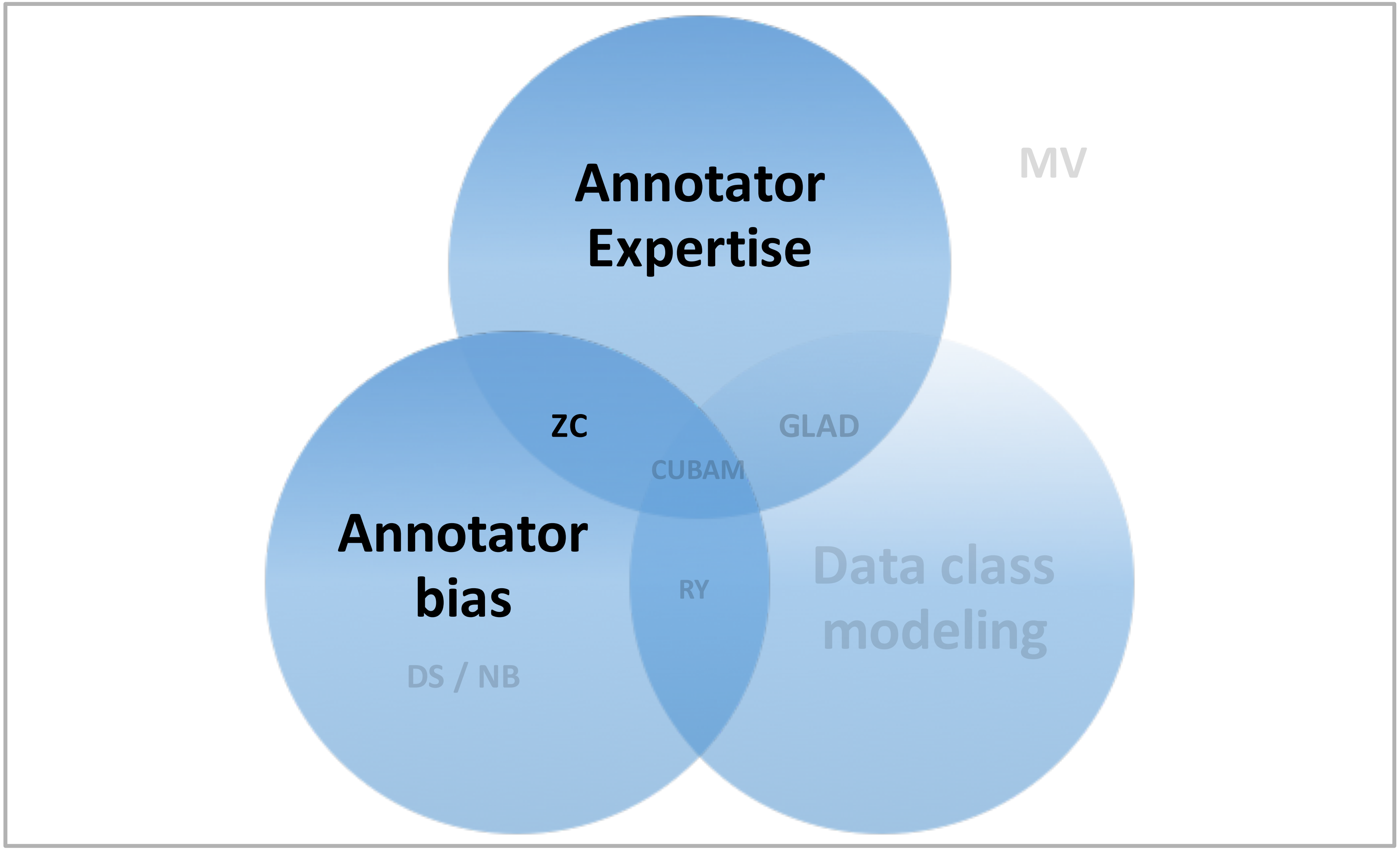
Modelling considerations



Majority voting

- Most widely used consensus method
- Assumes workers to be high quality and Independent, imply
  1. No modeling of worker behavior
  2. No modeling of annotation process
  3. Independent of task
- No estimations -> Fast label inferencing
- Applied under all supervision-classification modes

Modelling considerations



ZenCrowd

- Parameterizes worker reliability
  - Single Parameter per worker
  - Measure of worker expertise
  - Easily Generalizable to multi-class / Multi choice tasks
  - Easy mechanism to identify and handle Adversarial workers (bias)

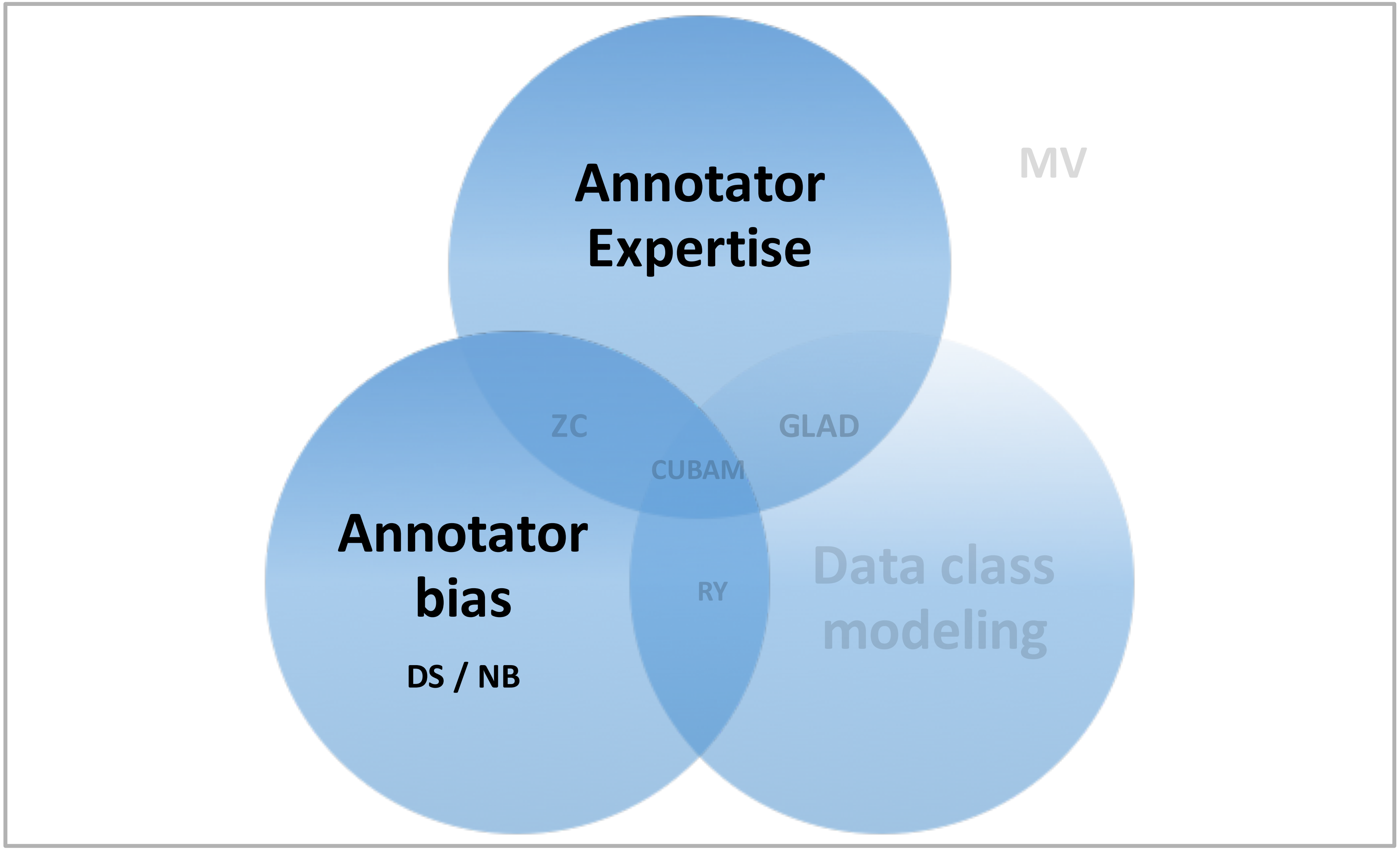
- Other assumptions apply as in MV

More about ZenCrowd

- Unsupervised in original version
  - ➢ Use Expectation-Maximization (EM) to estimate worker reliability and labels

- Could be operated under other supervision modes
  - ➢ Lightly Supervised by only providing class priors
  - ➢ Semi Supervised by providing gold labels for a subset [Wang et al. 2011]
  - ➢ Fully Supervised using MLE [Snow et al. 2008]

Modelling considerations



Dawid Skene & Naïve Bayes

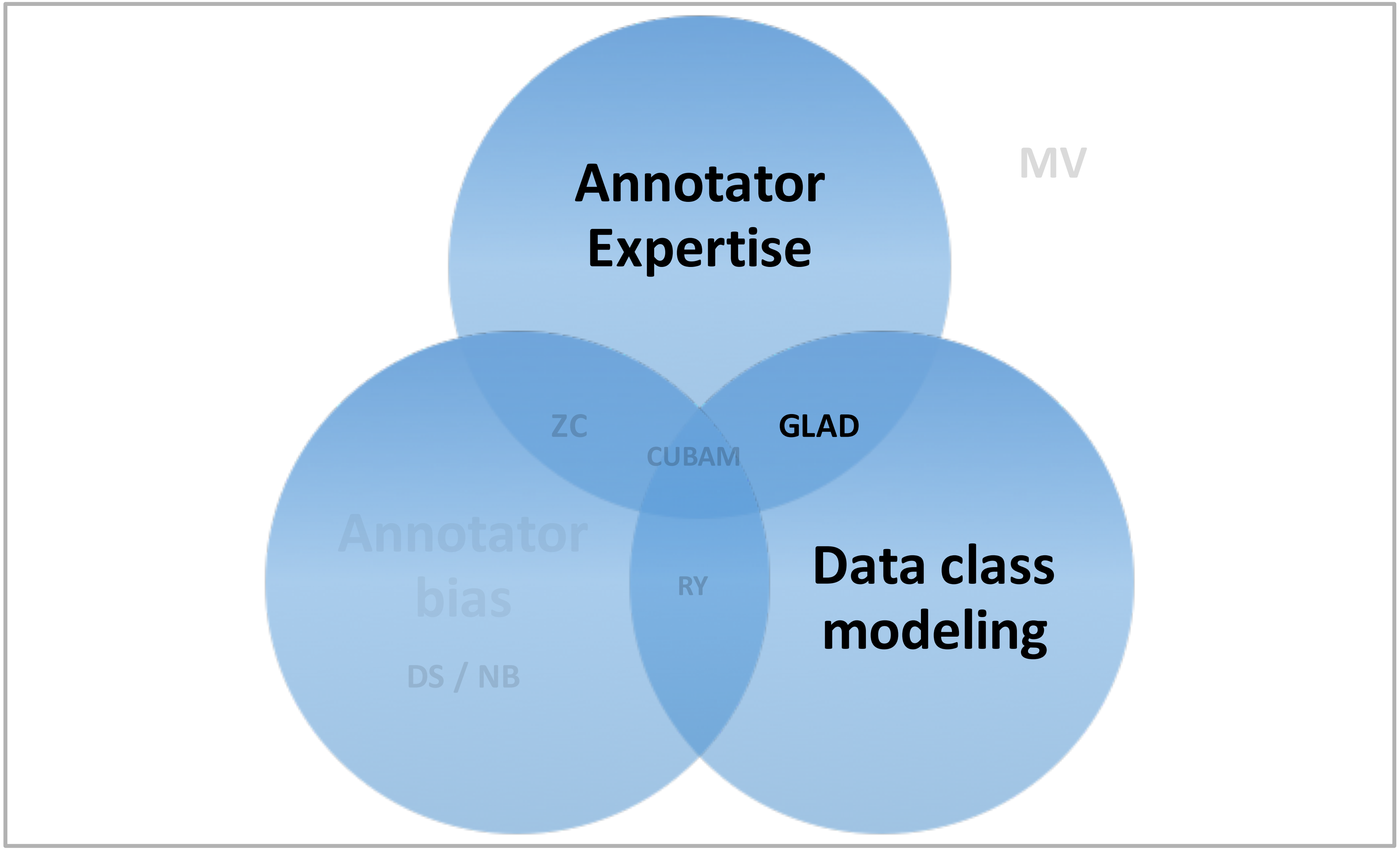
- Parameterizes worker bias w.r.t all classes
- Models Confusion Matrix for each worker and class prior
  - Implies one parameter per class per worker
  - Easy detection of perfections/imperfections of each worker w.r.t a class
  - Sparsity could be problematic.
  - Cannot be generalized to multiple choice Selection task
- Other assumptions apply as in MV

More about DS / NB

- Unsupervised in original version
  - Use EM to estimate worker CM and class prior

- Could be operated under other supervision modes
  - Under Lightly Supervised and semi-supervised conditions
    - Variant Estimation technique to distinguish b/w noise and worker bias
  - Under Full supervision by MLE using Laplacian smoothing

Modelling considerations

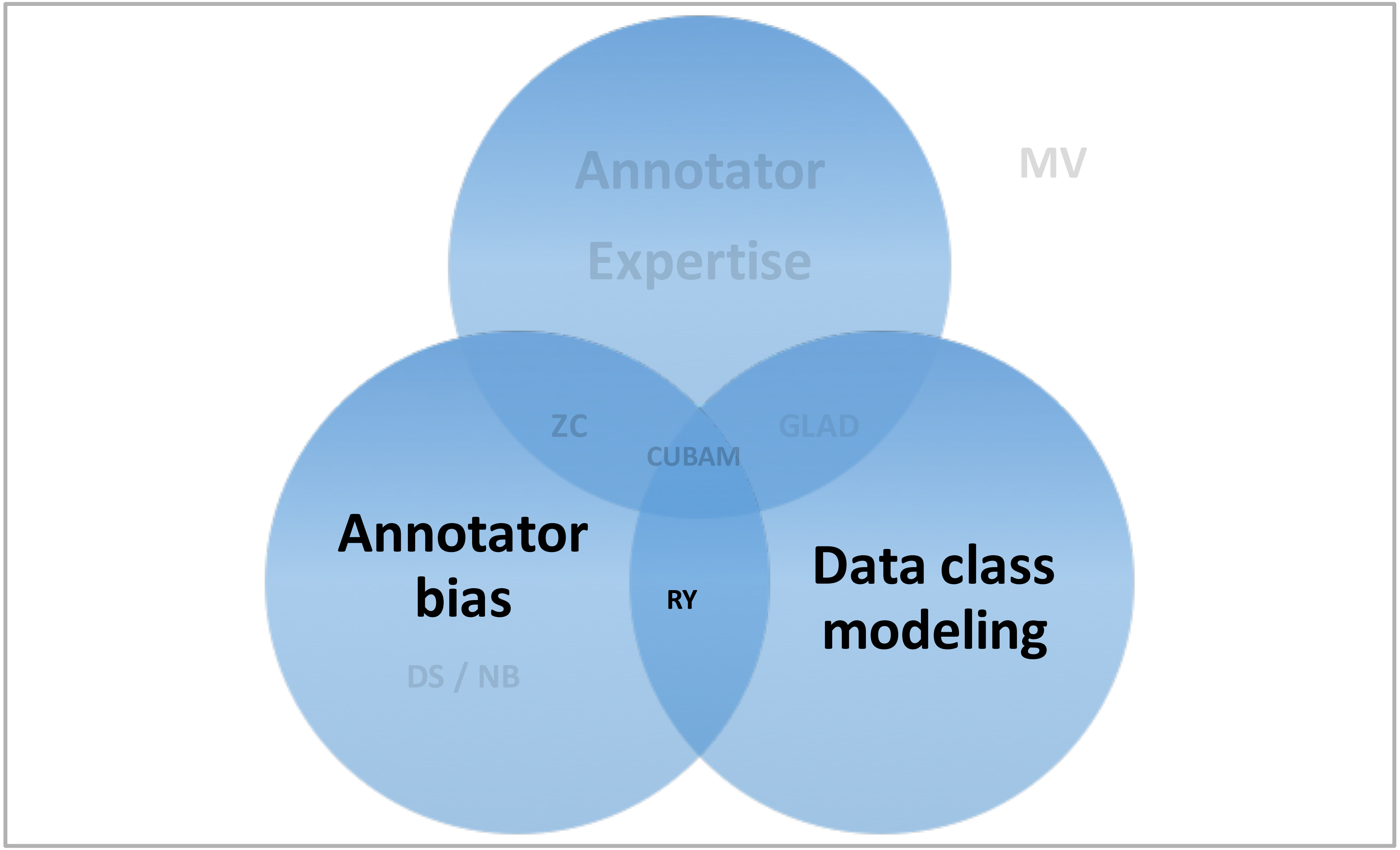


GLAD

- Joint Parameterization of worker expertise and class label (though example difficulty modeling)

- Originally proposed as unsupervised problem
  - Addressed using EM with gradient ascent M step
  - Extendable to Multi-class problems
  - Could be projected under other supervised settings (Fit labels in EM)

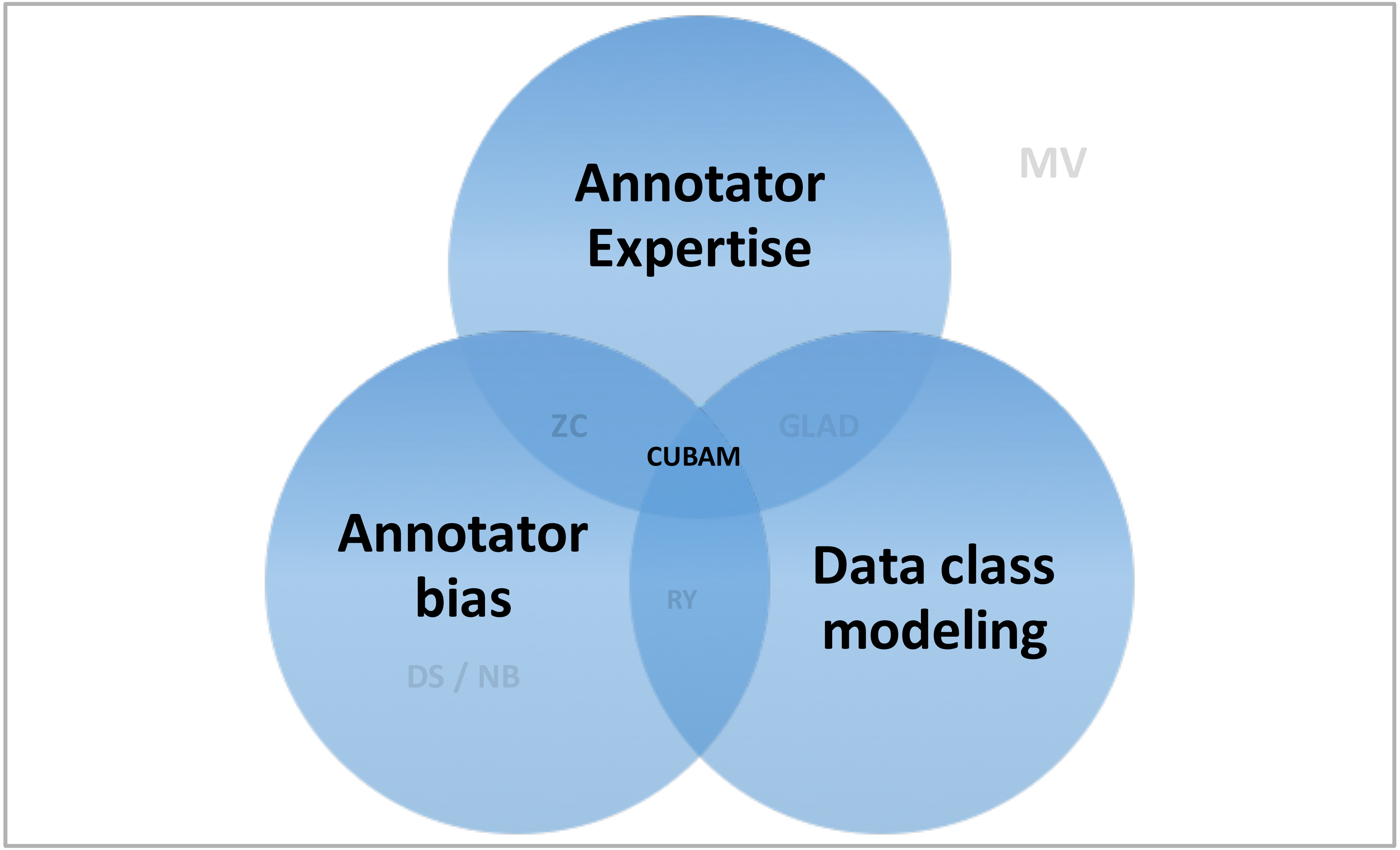
Modelling considerations



RY (Raykar)

- Joint parametrization of worker bias for each class
  - Specificity and Sensitivity in binary case modeled using beta prior
  - Dirichlet prior in multi-class case
- Optional: Feature representation of example
  - Used to estimate labels if available, DS like mechanism otherwise
- Originally proposed as unsupervised problem
  - EM to estimate worker bias parameters
  - Extendable to Multi-class problems
  - Class (example) parameters => No multi-choice

Modelling considerations



CUBAM

- Incorporate all the three modalities
  - Normalized relevance weight to each worker
  - Worker labels could be determined, given input and worker specific parameters
  - Labels obtained by MAP estimation of worker specific parameters and input
  - Worker expertise/bias could be determined.
- Multi-class classification possible
- No direct supervision is apparent

Summary

	Unsup	Light sup.	Semi sup.	Full sup.	Multi-class	Mult-choice Q
MV	✔	✔	✔	✔	✔	✔
ZC	✔	✔	✔	✔	✔	✔
DS	✔	✔	✔	✔	✔	✘
NB	✘	✘	✘	✔	✔	✘
GLAD	✔	✔	✔	✔	✔	✔
RY	✔	✔	✔	✔	✔	✘
CUBAM	✔	✘	✘	✘	✔	✘

- Benchmark Datasets for comparisons
- Consensus methods
- Experiment Setup and Results
- Concluding Remarks

Goal

- Project consensus under exhaustive set of settings
- Analyze importance of each assumption
- Analyze impact of different conditions on the performance (e.g. supervision)

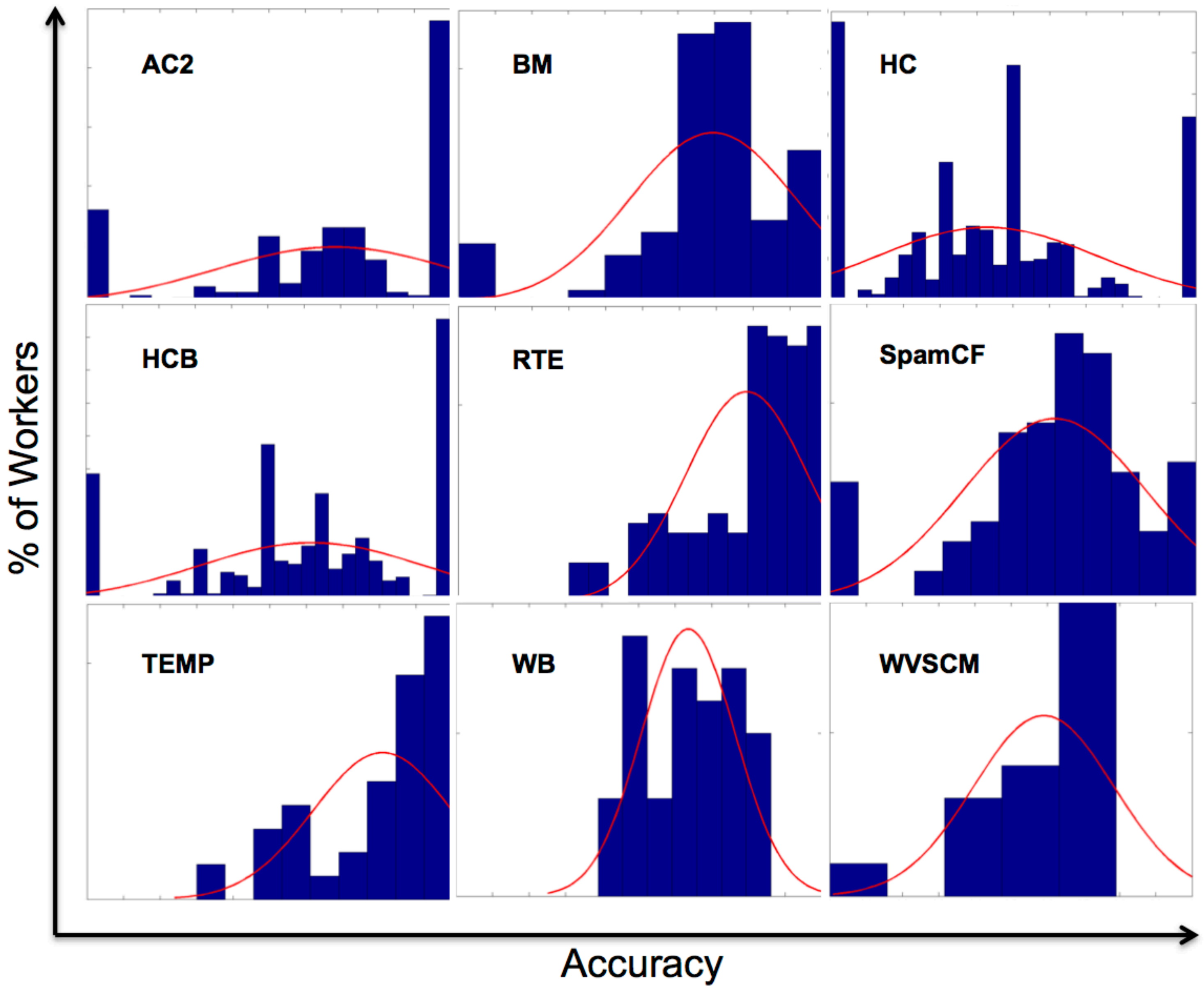
Possible settings

- Dataset
- Degree of Supervision
- Label Noise
- Evaluation Metrics

Degree of Supervision

- Ranging from unsupervised to 90% supervised
- Cross-validation employed in each case, with varying folds
- Unsupervised – No information about hyperparameters/priors
- Light supervision – Data Prior distribution available
- Full Supervision – Gold-labeled training examples available

Sampling visualization



Label Noise

- Noise added to worker label predictions
- Noise added under maximizing data realism
  - Preservation of worker accuracy distribution
  - Preservation of worker-example correspondences
- Worker accuracy parameterized by normal distribution
- Different proportions of noise added
  - For each worker, accuracy sampling is performed, new labels generated

Evaluation Metrics

- Ideal metric: Simplify understanding, easy adaption
- F1 and accuracy metrics employed here
- Other alternatives possible – e.g. Significance testing [Smucker et al. 2007]

Details about consensus implementations - ZC

- Beta priors used for worker reliability
- Dirichlet priors used for class label distribution
- In unsupervised setting, uniform distribution was used for category distribution modeling
- Parameters computed using training data in supervised setting

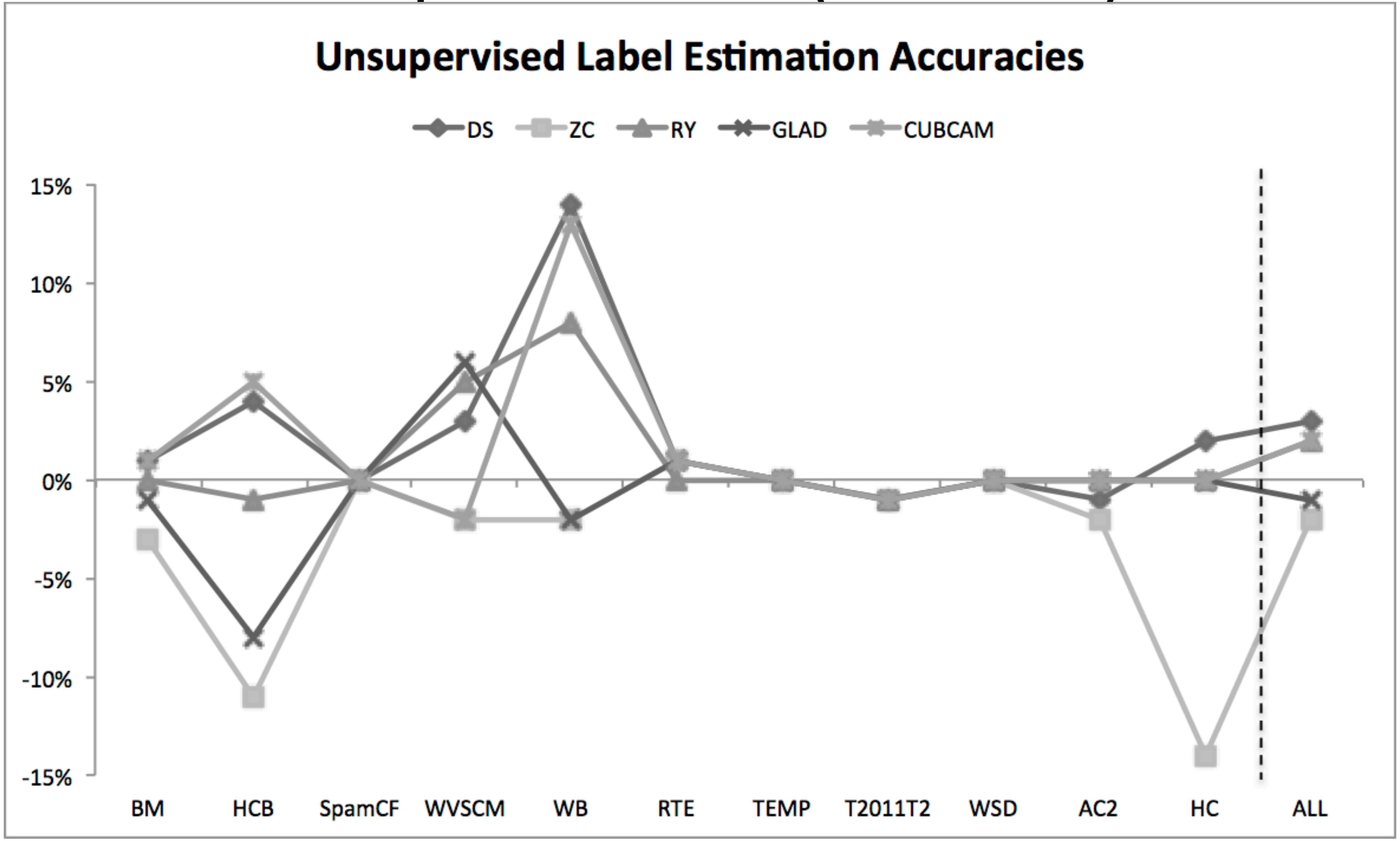
Details about consensus implementations - RY

- Beta priors used for worker specificity – sensitivity
- Only used for binary classification as in original paper [Rayker et al. 2010]
- In unsupervised setting, parameters were assumed as in ZC
- Parameters computed using training data in supervised setting

Details about consensus implementations – CUBAM, DS, GLAD

- CUBAM – same prior assumptions as original implementation
- DS – priors not assumed – computed using training data when available
- GLAD – uniform class distribution assumed – computed using training data when available
- CUBAM / GLAD – support only binary classes

Results on unmodified datasets – Unsupervised (vs MV)



Results on unmodified datasets – Absolute Numbers

Method	Metric	No Supervision	Light-Supervision 10%	20%	50%	80%	90%	Full-Supevision 10%	20%	50%	80%	90%	Count
MV	*Acc*	79.2	79.2	79.2	79.2	79.3	79.3	79.2	79.2	79.2	79.3	79.3	0
	F_1	77.5	77.5	77.5	77.2	78.0	78.1	77.5	77.5	77.2	78.0	78.1	0
ZC	*Acc*	77.2	76.3	77.1	78.4	78.9	78.9	76.8	77.6	78.7	80.4	80.8	0
	F_1	76.4	74.2	75.7	76.8	77.7	77.7	75.4	76.1	77.0	79.2	79.6	0
GLAD	*Acc*	78.7	78.1	78.0	78.2	78.9	78.0	78.3	78.5	79.2	79.8	80.3	0
	F_1	77.3	76.8	76.7	77.0	78.6	77.6	76.9	77.1	77.6	79.0	79.5	0
NB	*Acc*	-	-	-	-	-	-	80.3	80.7	80.5	80.7	80.5	0
	F_1	-	-	-	-	-	-	79.1	79.0	78.5	78.5	78.9	0
DS	*Acc*	**82.2**	**82.3**	**82.2**	**82.0**	80.4	79.5	**82.2**	**82.2**	82.1	81.8	81.9	6
	F_1	80.2	80.2	80.0	79.4	78.9	77.9	80.1	80.0	79.6	79.2	79.9	7
RY	*Acc*	80.9	81.6	81.6	81.5	**80.5**	**80.1**	81.9	82.0	**82.5**	**82.3**	**82.3**	5
	F_1	79.1	79.6	79.5	79.2	78.8	78.8	79.8	79.9	79.9	80.4	80.4	4
CUBAM	*Acc*	81.5	-	-	-	-	-	-	-	-	-	-	0
	F_1	79.8	-	-	-	-	-	-	-	-	-	-	0

General Trends – unmodified datasets

- DS – top average performance under <50% supervision
- RY – top average performance under > 70% supervision
- CUBAM – performs well under noisy conditions of HCB
- MV - comparable to others under noisy and scarce data

Method	Metric	No Supervision	Light-Supervision					Full-Supevision					Count
			10%	20%	50%	80%	90%	10%	20%	50%	80%	90%	
MV	Acc	79.2	79.2	79.2	79.2	79.3	79.3	79.2	79.2	79.2	79.3	79.3	0
	F_1	77.5	77.5	77.5	77.2	78.0	78.1	77.5	77.5	77.2	78.0	78.1	0
ZC	Acc	77.2	76.3	77.1	78.4	78.9	78.9	76.8	77.6	78.7	80.4	80.8	0
	F_1	76.4	74.2	75.7	76.8	77.7	77.7	75.4	76.1	77.0	79.2	79.6	0
$GLAD$	Acc	78.7	78.1	78.0	78.2	78.9	78.0	78.3	78.5	79.2	79.8	80.3	0
	F_1	77.3	76.8	76.7	77.0	78.6	77.6	76.9	77.1	77.6	79.0	79.5	0
NB	Acc	-	-	-	-	-	-	80.3	80.7	80.5	80.7	80.5	0
	F_1	-	-	-	-	-	-	79.1	79.0	78.5	78.5	78.9	0
DS	Acc	**82.2**	**82.3**	**82.2**	**82.0**	80.4	79.5	**82.2**	**82.2**	82.1	81.8	81.9	6
	F_1	80.2	80.2	80.0	79.4	78.9	77.9	80.1	80.0	79.6	79.2	79.9	7
RY	Acc	80.9	81.6	81.6	81.5	**80.5**	**80.1**	81.9	82.0	**82.5**	**82.3**	**82.3**	5
	F_1	79.1	79.6	79.5	79.2	78.8	78.8	79.8	79.9	79.9	80.4	80.4	4
$CUBAM$	Acc	81.5	-	-	-	-	-	-	-	-	-	-	0
	F_1	79.8	-	-	-	-	-	-	-	-	-	-	0

General Trends – unmodified datasets

- Performance limited by worker/class prior modeling

- Smart modelling of workers => Good performance
  - Sufficiency in model complexity further gets aided by supervision

Results on datasets – Different Noise injections

Method	Fold Size	No Supervision Avg. Worker Accuracy 55%	65%	75%	85%	95%	Light-Supervision Avg. Worker Accuracy 55%	65%	75%	85%	95%	Full-Supervision Avg. Worker Accuracy 55%	65%	75%	85%	95%	Count
MV	20%	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	0
	80%	-	-	-	-	-	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	0
ZC	20%	56.6	77.8	85.2	**99.2**	99.8	55.9	70.9	85.1	**99.1**	**99.8**	66.2	85.9	**95.9**	**99.1**	99.7	5
	80%	-	-	-	-	-	57.8	67.7	86.6	**98.3**	**99.6**	85.9	90.0	95.4	**99.0**	**99.9**	5
GLAD	20%	52.6	70.3	84.8	98.4	98.6	53.7	70.8	85.0	98.4	98.7	63.8	71.0	87.3	98.6	99.7	0
	80%	-	-	-	-	-	47.1	69.3	85.9	97.8	98.9	69.6	81.1	95.7	98.4	99.7	0
NB	20%	-	-	-	-	-	-	-	-	-	-	82.2	86.1	93.8	97.2	98.8	0
	80%	-	-	-	-	-	-	-	-	-	-	84.0	88.0	95.0	98.6	99.6	0
DS	20%	45.0	85.3	91.2	99.1	**99.9**	48.3	**85.2**	**91.3**	99.0	99.7	75.3	89.8	95.6	**99.1**	**99.9**	5
	80%	-	-	-	-	-	47.1	**82.0**	**92.1**	98.0	99.4	**86.2**	**90.7**	**96.0**	**99.0**	99.8	7
RY	20%	**56.9**	**86.1**	**95.3**	99.1	99.7	**59.1**	69.9	86.9	98.9	**99.8**	**83.1**	**90.0**	95.8	98.9	99.8	7
	80%	-	-	-	-	-	**71.0**	78.9	89.1	98.1	99.4	85.7	90.3	95.3	98.9	99.7	4
CUBAM	20%	52.4	67.6	83.2	97.7	97.9	-	-	-	-	-	-	-	-	-	-	0
	80%	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

General Trends – Noisy datasets

- MV – Outperformed at modest noise levels (50%)
- DS – top average performance under low noise
- CUBAM – doesn't perform better than others
- RY – performs well with supervisions – highlights significance of priors

Method	Fold Size	No Supervision Avg. Worker Accuracy					Light-Supervision Avg. Worker Accuracy					Full-Supervision Avg. Worker Accuracy					
		55%	65%	75%	85%	95%	55%	65%	75%	85%	95%	55%	65%	75%	85%	95%	Count
MV	20%	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	0
	80%	-	-	-	-	-	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	0
ZC	20%	56.6	77.8	85.2	**99.2**	99.8	55.9	70.9	85.1	**99.1**	**99.8**	66.2	85.9	**95.9**	**99.1**	99.7	5
	80%	-	-	-	-	-	57.8	67.7	86.6	**98.3**	**99.6**	85.9	90.0	95.4	**99.0**	**99.9**	5
GLAD	20%	52.6	70.3	84.8	98.4	98.6	53.7	70.8	85.0	98.4	98.7	63.8	71.0	87.3	98.6	99.7	0
	80%	-	-	-	-	-	47.1	69.3	85.9	97.8	98.9	69.6	81.1	95.7	98.4	99.7	0
NB	20%	-	-	-	-	-	-	-	-	-	-	82.2	86.1	93.8	97.2	98.8	0
	80%	-	-	-	-	-	-	-	-	-	-	84.0	88.0	95.0	98.6	99.6	0
DS	20%	45.0	85.3	91.2	99.1	**99.9**	48.3	**85.2**	**91.3**	99.0	99.7	75.3	89.8	95.6	**99.1**	**99.9**	5
	80%	-	-	-	-	-	47.1	**82.0**	**92.1**	98.0	99.4	**86.2**	**90.7**	**96.0**	**99.0**	99.8	7
RY	20%	**56.9**	**86.1**	**95.3**	99.1	99.7	**59.1**	69.9	86.9	98.9	**99.8**	**83.1**	**90.0**	95.8	98.9	99.8	7
	80%	-	-	-	-	-	**71.0**	78.9	89.1	98.1	99.4	85.7	90.3	95.3	98.9	99.7	4
CUBAM	20%	52.4	67.6	83.2	97.7	97.9	-	-	-	-	-	-	-	-	-	-	0
	80%	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

General Trends – Overall

- MV – Outperformed when noise levels are high
- DS – top average performance under low noise
- RY – top average performance under high supervision – highlights significance of domain knowledge
- CUBAM – doesn’t perform better than others –shows performance sensitivity to datasets
- ZC and GLAD perform similarly

- Benchmark Datasets for comparisons
- Consensus methods
- Experiment Setup and Results
- Concluding Remarks

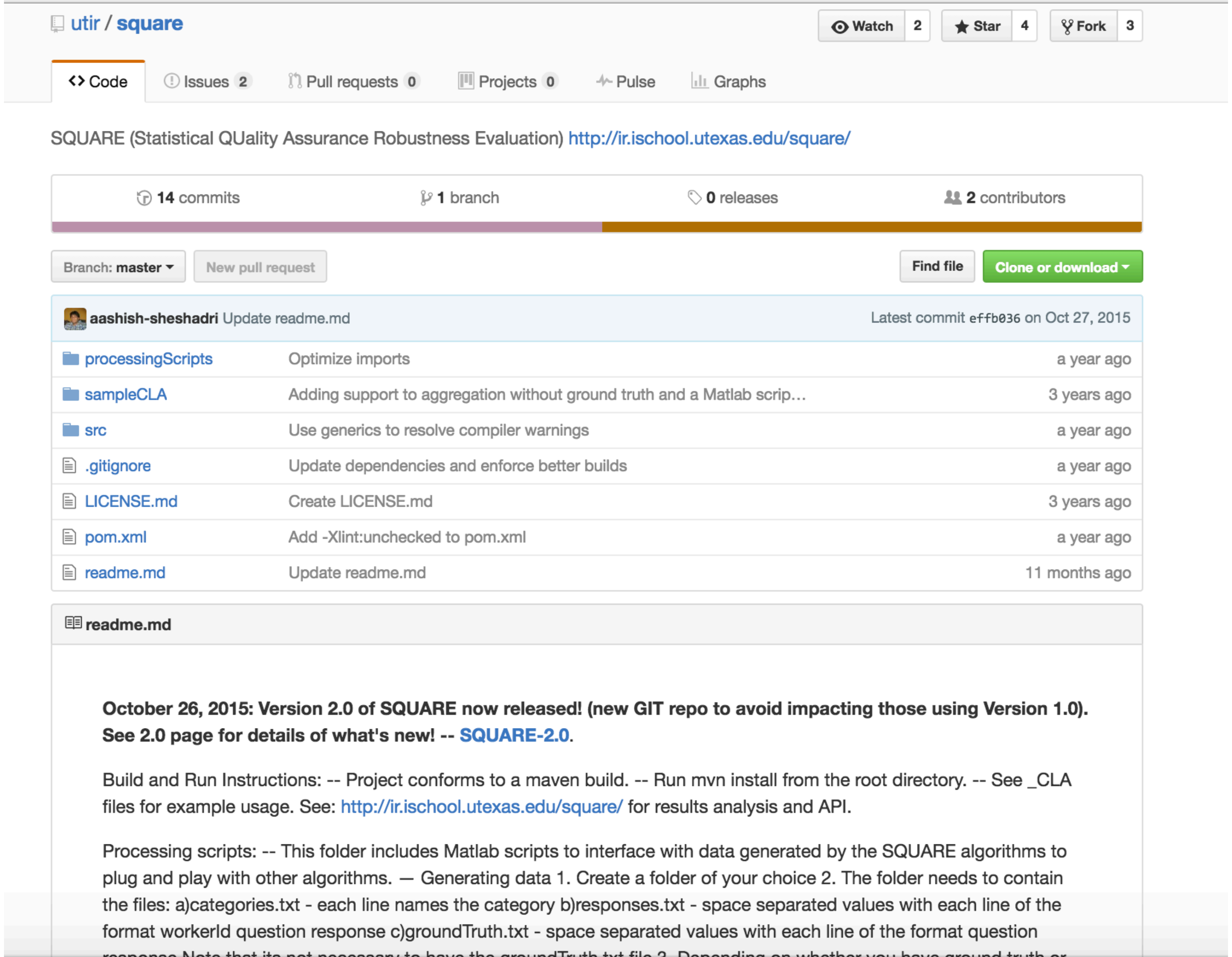
Conclusion

- Design of a comprehensive benchmark for evaluating consensus methods
- Diverse dataset – consensus methods to rely on
- Diverse experiments to compare and contrast different consensus methods

Future work

- Improved Benchmark tests for better analysis
- Tuning of current methods for fair comparisons
- Analysis under sampling from worker emperical distribution rather than normal distribution

SQUARE API on web


utir / square

Watch 2 | Star 4 | Fork 3

Code | Issues 2 | Pull requests 0 | Projects 0 | Pulse | Graphs

SQUARE (Statistical QUality Assurance Robustness Evaluation) http://ir.ischool.utexas.edu/square/

14 commits | 1 branch | 0 releases | 2 contributors

Branch: master | New pull request | Find file | Clone or download

aashish-sheshadri Update readme.md — Latest commit effb036 on Oct 27, 2015

processingScripts	Optimize imports	a year ago
sampleCLA	Adding support to aggregation without ground truth and a Matlab scrip…	3 years ago
src	Use generics to resolve compiler warnings	a year ago
.gitignore	Update dependencies and enforce better builds	a year ago
LICENSE.md	Create LICENSE.md	3 years ago
pom.xml	Add -Xlint:unchecked to pom.xml	a year ago
readme.md	Update readme.md	11 months ago

readme.md

October 26, 2015: Version 2.0 of SQUARE now released! (new GIT repo to avoid impacting those using Version 1.0). See 2.0 page for details of what's new! -- SQUARE-2.0.

Build and Run Instructions: -- Project conforms to a maven build. -- Run mvn install from the root directory. -- See _CLA files for example usage. See: http://ir.ischool.utexas.edu/square/ for results analysis and API.

Processing scripts: -- This folder includes Matlab scripts to interface with data generated by the SQUARE algorithms to plug and play with other algorithms. — Generating data 1. Create a folder of your choice 2. The folder needs to contain the files: a)categories.txt - each line names the category b)responses.txt - space separated values with each line of the format workerId question response c)groundTruth.txt - space separated values with each line of the format question

Thank you