Noise Eliminination with Ensemble-Partitioning Filter

A Generic Implementation for Software Quality Engineering

Dr. T. M. Khoshgoftaar and P. Rebours

taghi@cse.fau.edu

Dept. of Computer Science and Engineering

Florida Atlantic University

Boca Raton, FL 33431

0.1 ...

Overview

- Introduction
- Ensemble-Partitioning Filter
- Case Studies
- Conclusion

Software Quality Model (SQM)

- Proved technique in achieving better software quality control
- Two-group classification model

Actual class \ Predicted class	*fp*	*nfp*
fp	true positive	false negative‡
nfp	false positive†	true negative

† Type I error

‡ Type II error

- Type II error more severe

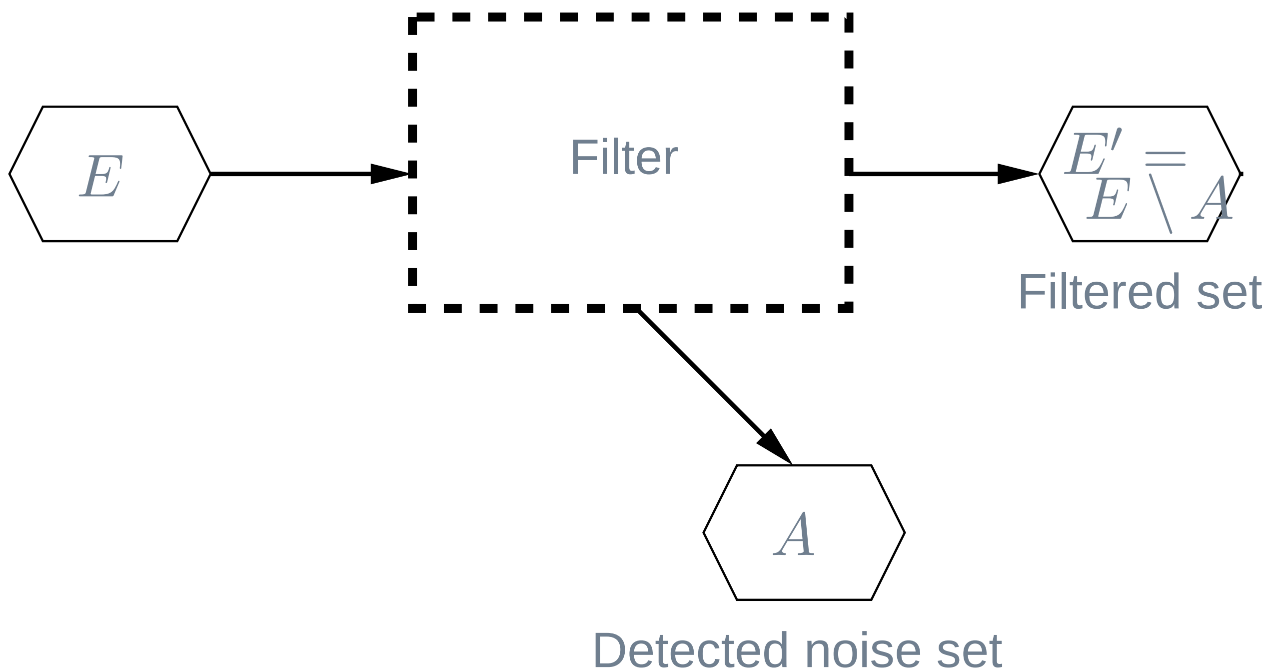
Importance of Data Quality

- Improve the performance accuracies of SQMs
- Information $\Longrightarrow$ key to success for any organization
- Common for large datasets to have noise ($\geq 5\%$)
- Disastrous consequences if not handled correctly

1 Ensemble-Partitioning Filter

What is Filtering?

- A filter removes instances suspected to be noisy
- $f(I_k) = \{\textit{clean}, \textit{noisy}\}$



1.1 Ensemble Filter

Principles

- Use m base learners, $m = 5$ or 25
- I_k identified as *noisy* if it is misclassified by λ classifiers
- λ, filtering level
- Each base learner L_i can be seen as an *expert*

Pros/Cons

Pros

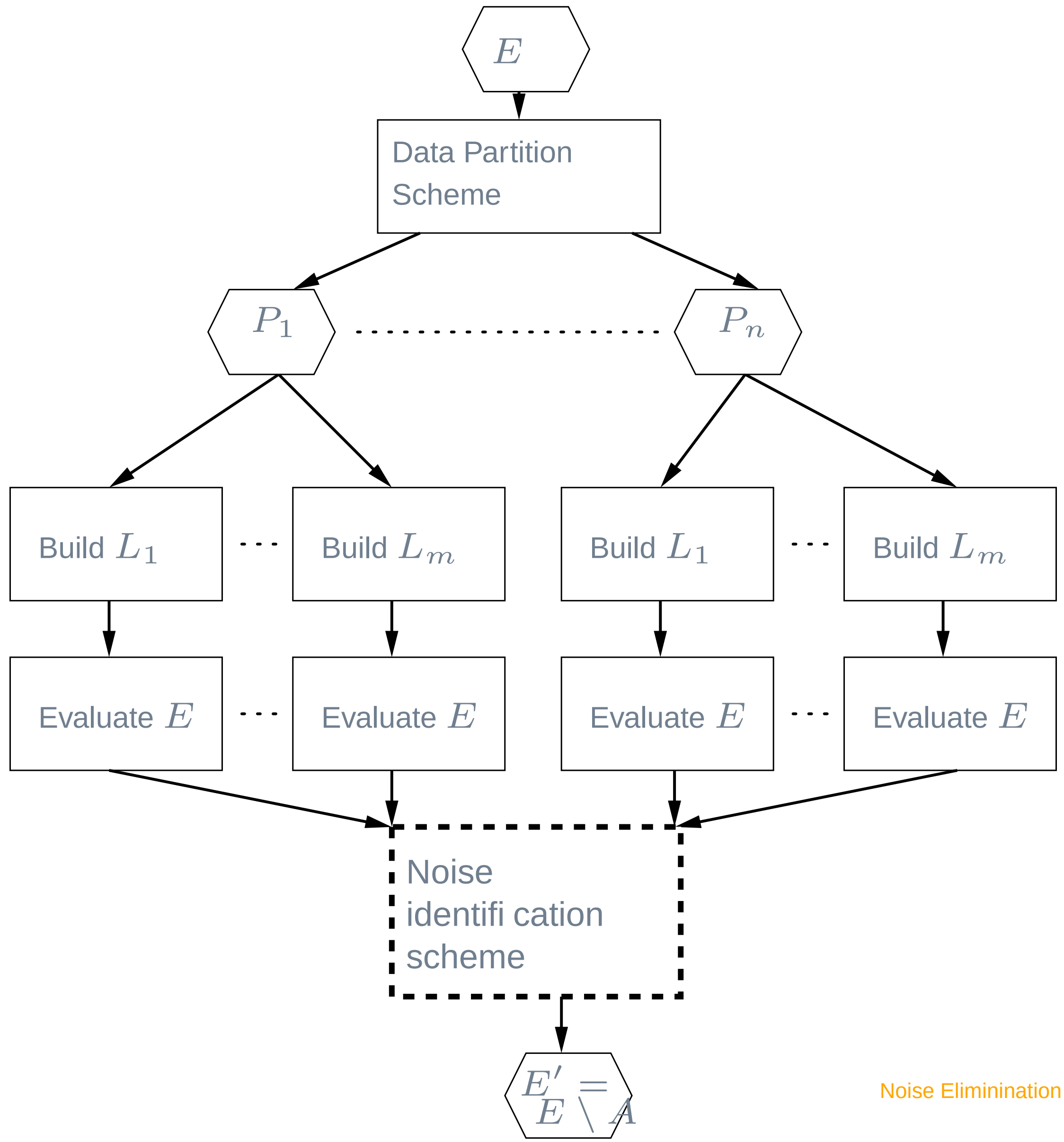
- Flexibility of the level of conservativeness
- Combine bias of different learners
- Higher degree of confidence in tossing out the instances suspects of being noisy.

Cons

- Expertise of different data mining techniques
- Requires to build m models
- Problem with large datasets

1.2 Partitioning Filter

Principles



E
Data Partition Scheme
P_1
P_n
Build L_1
Build L_m
Build L_1
Build L_m
Evaluate E
Evaluate E
Evaluate E
Evaluate E
Noise identifi cation scheme
$E' = E \setminus A$

Local and Global Experts

For each instance, two counters, S_k^{le} and S_k^{ge}:

- $I_k \in P_i$ and $L_j^{cv}(I_k, P_i) \neq c_k \implies S_k^{le}++$
- $I_k \notin P_i$ and $L_j(I_k, P_i) \neq c_k \implies S_k^{ge}++$
- Noisy instances have large value for S_k^{le} and S_k^{ge}

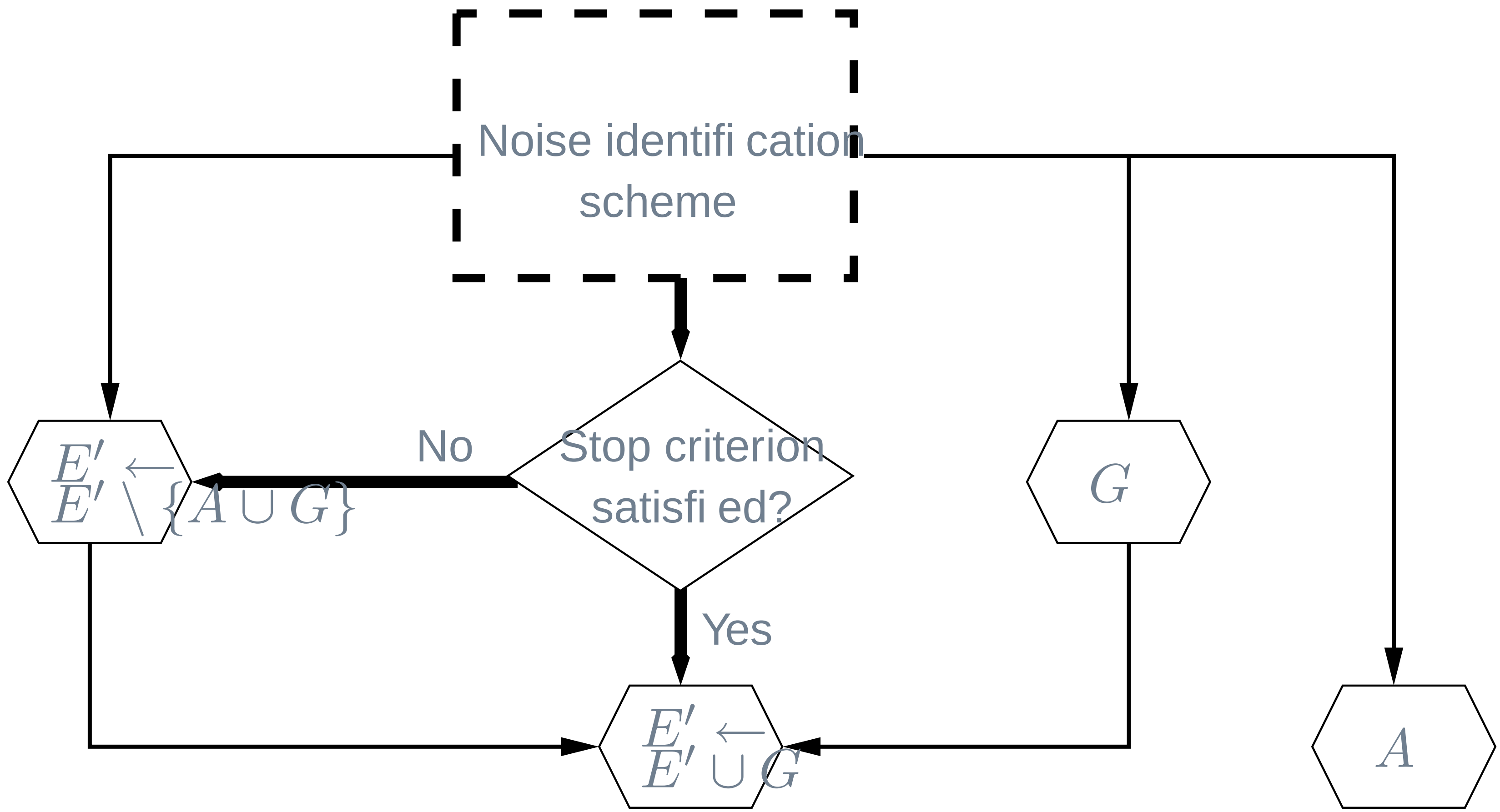
Voting Schemes

- I_k is identified as *noisy* only if $S_k^{le} = m$
- Classifier has a higher prediction accuracy with the instances in its training set
- I_k identified as *noisy* if $S_k^{le} + S_k^{ge} \geq \lambda$
- $m \times n$ experts

Iterative-Partitioning Filter

- $m = 1$ and $n = 5$
- Multi-round execution
- Two voting schemes:
  - Consensus scheme (*ipfcons*)
  - Majority scheme (*ipfmaj*)

Iterative Process



Multiple-Partitioning Filter

- $m = 5$ and $n = 5$
- No iterative execution
- With or without the cross-validation constraint
  - *mpf* $(I_k) =$ *noisy* $\Longrightarrow S_k^{ge} + S_k^{le} \geq \lambda$
  - *mpfcv* $(I_k) =$ *noisy* $\Longrightarrow S_k^{ge} + S_k^{le} \geq \lambda$ and $S_k^{le} = m$
- Use of local and global experts

Example

L_i induced on P_i		I_1	I_2	I_3
L_1	P_1	***fp***	***fp***	*fp*
	P_2	*fp*	*nfp*	***nfp***
	P_3	*nfp*	*nfp*	*fp*
L_2	P_1	***fp***	***nfp***	*fp*
	P_2	*nfp*	*fp*	***nfp***
	P_3	*nfp*	*nfp*	*fp*
L_3	P_1	***fp***	***fp***	*nfp*
	P_2	*nfp*	*nfp*	***nfp***
	P_3	*fp*	*fp*	*fp*
Class c_k		*nfp*	*fp*	*fp*
Partition i (P_i)		1	1	2
Noisy		$\surd$		

$$\lambda = 5$$

$$n = 3$$

$$m = 3$$

Pros/Cons

Pros

- Handle large and distributed datasets
- Iterative process
- Flexibility on the level of conservativness
- Combine bias of different learners
- Need less expertize than the Ensemble Filter

Cons

- Requires to build $m \times n$ models

1.3 Unified Framework

Input Parameters

- n, number of subsets
- $L_i \; i = 1, \ldots, m$, base learners
- bCv, boolean value indicating whether or not the cross-validation constraint is used
- λ, filtering level
- β, the rate of good examples to be removed in each round
- Stopping criterion

Specialization

Symbol	m	n	bCv	λ	Iteration	L	G	T
cvf	1	1	NA	1	no	1	0	1
ef	25	1	NA	13-25	no	25	0	25
sef	5	1	NA	3-5	no	5	0	5
ipfcons	1	5	true	5	yes	1	4	5
ipfmaj	1	5	true	3	yes	1	4	5
mpf	5	5	false	13-25	no	5	20	25
mpfcv	5	5	true	13-25	no	5	20	25

2 Case Studies

Domain Dataset

- Software quality data from NASA projects
- Very high misclassification rates indicated the presence of inherent noise in the data
- 8850 instances

Learner	Type I	Type II
IBk	32.70%	32.48%
OneR	34.50%	34.38%
JRip	33.18%	33.08%
J48	32.56%	32.42%
LWLStump	33.59%	33.61%

2.1 Noise Removal

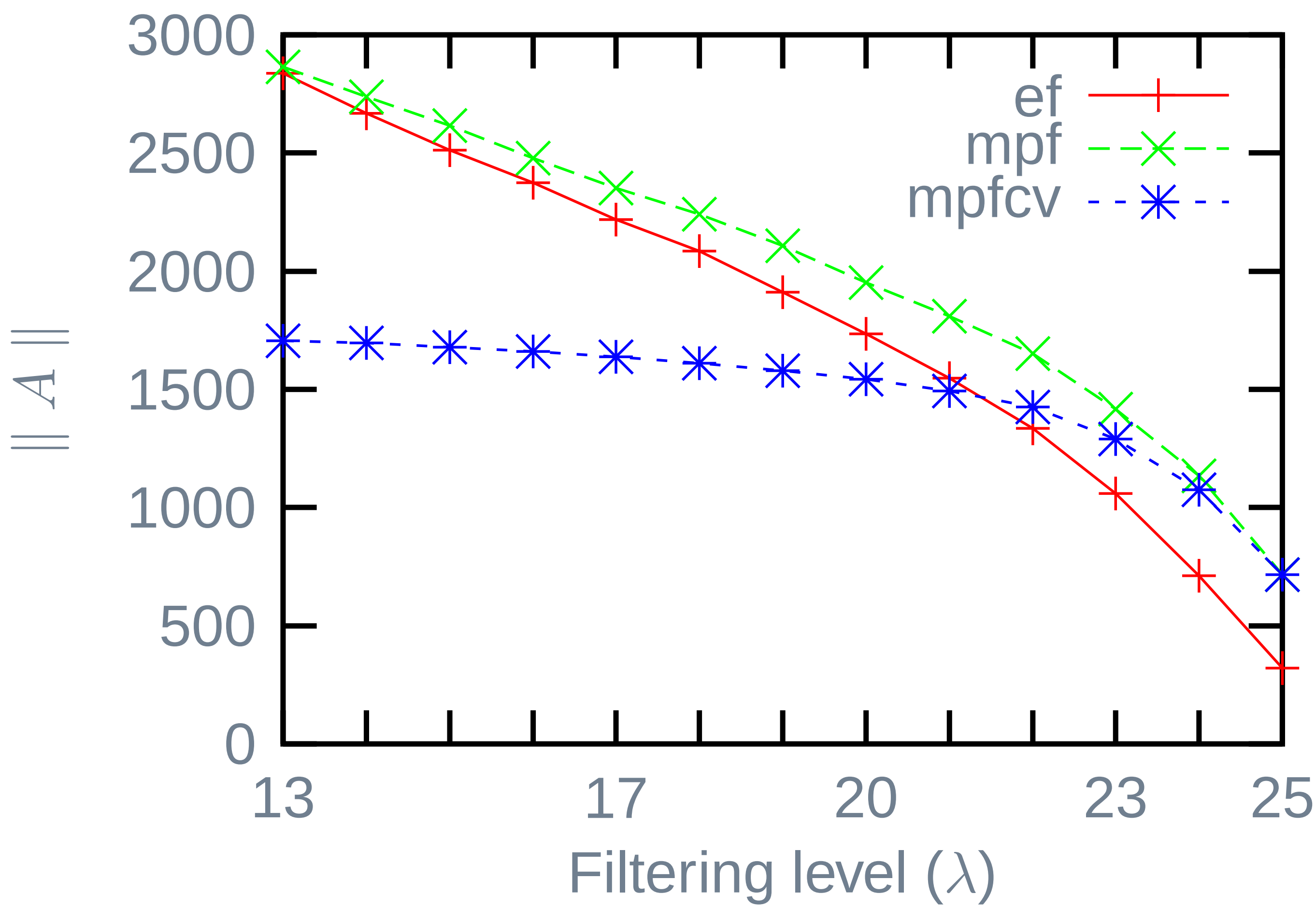
Most Aggressive Filters

Filters	Count	Proportion
ipfmaj-7	3131	35.38%
ipfmaj-6	3116	35.21%
ipfmaj-5	3107	35.11%
ipfmaj-4	3071	34.70%
ipfmaj-3	2979	33.66%
cvf	2879	32.53%
mpf-13	2864	32.36%
ef-13	2837	32.06%
⋮	⋮	⋮

Most Conservative Filters

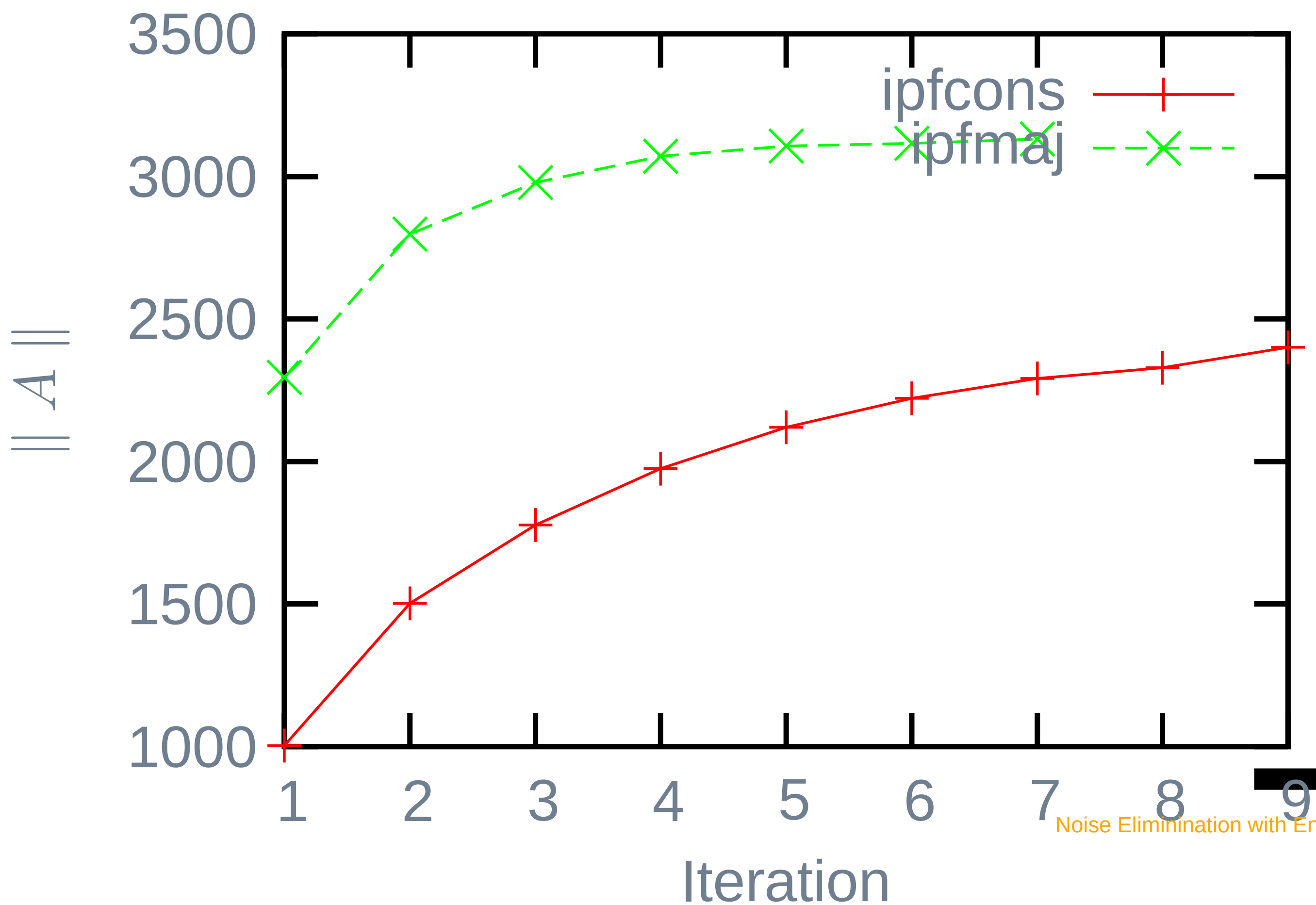
Filters	Count	Proportion
⋮	⋮	⋮
mpf-24	1135	12.82%
mpfcv-24	1076	12.16%
ef-23	1059	11.97%
ipfcons-1	1004	11.34%
mpf-25	717	8.10%
mpfcv-25	717	8.10%
ef-24	711	8.03%
ef-25	321	3.63%

At a Given Filtering Level



Iterative-Partitioning Filter

- $n = 5$
- $m = 1$ (J48)



Combination

	Filters	Combination count			
Combination	that agree	*nfp*	*fp*	Total	Proportion
<empty>	NA	2065	376	2441	27.58%
cvf	1/59	175	24	199	2.25%
All the filters	59/59	113	9	122	1.38%

Conclusion

- Two new filtering schemes
- Unified framework
- Cross-Validation Filtertoo aggressive
- Ensemble Filterat high filtering level is conservative

Questions?