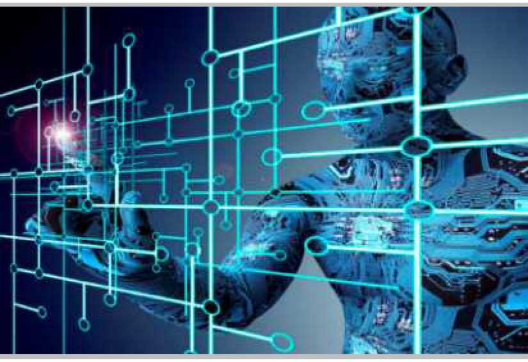
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2018-8365C



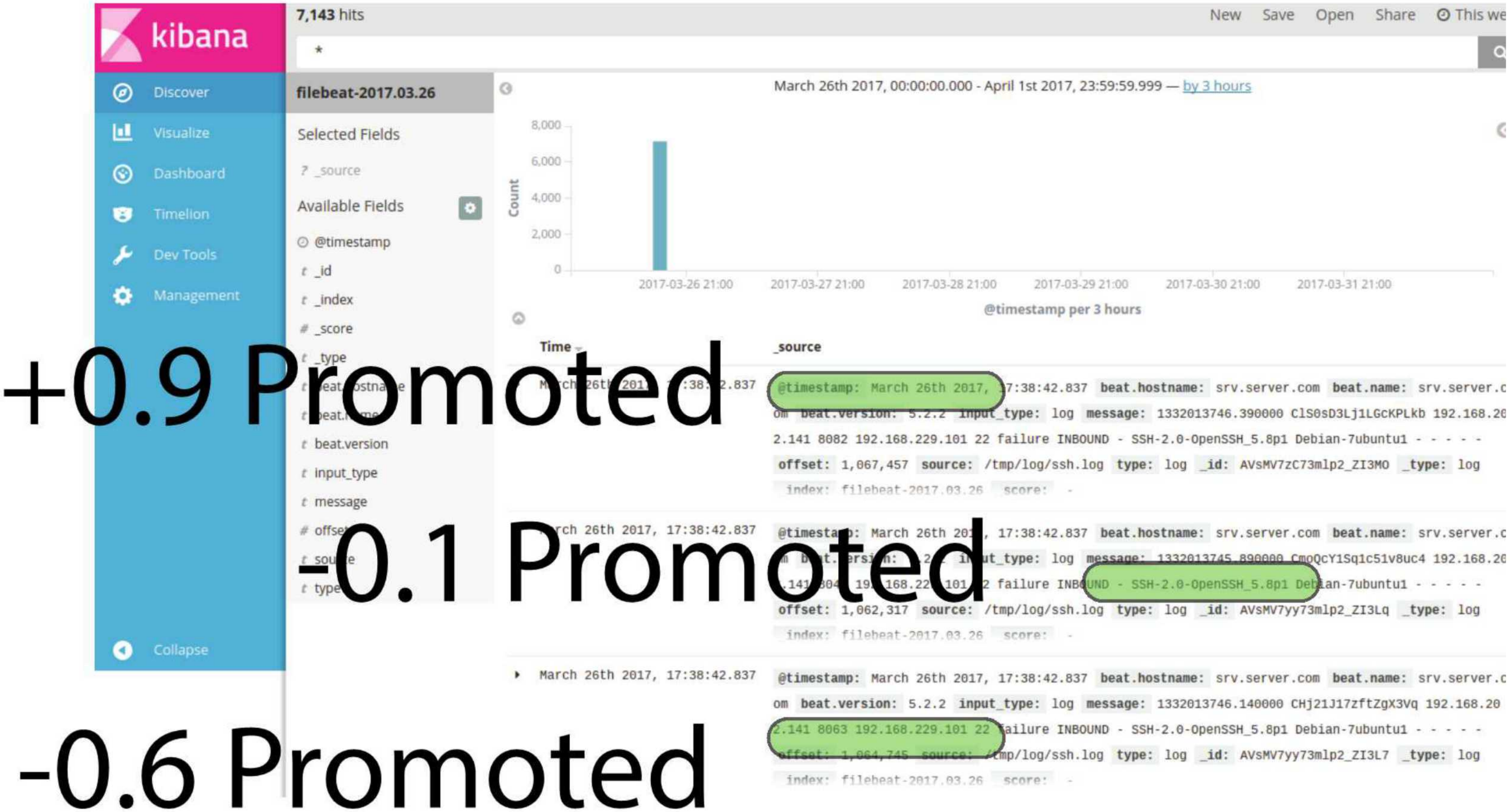
Designing for Interpretability and Adaptability by Using Weighted Averages

Sapan Agarwal, Andrew De La Cruz, Corey Hudson

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Want to Highlight Key Features in an Alert

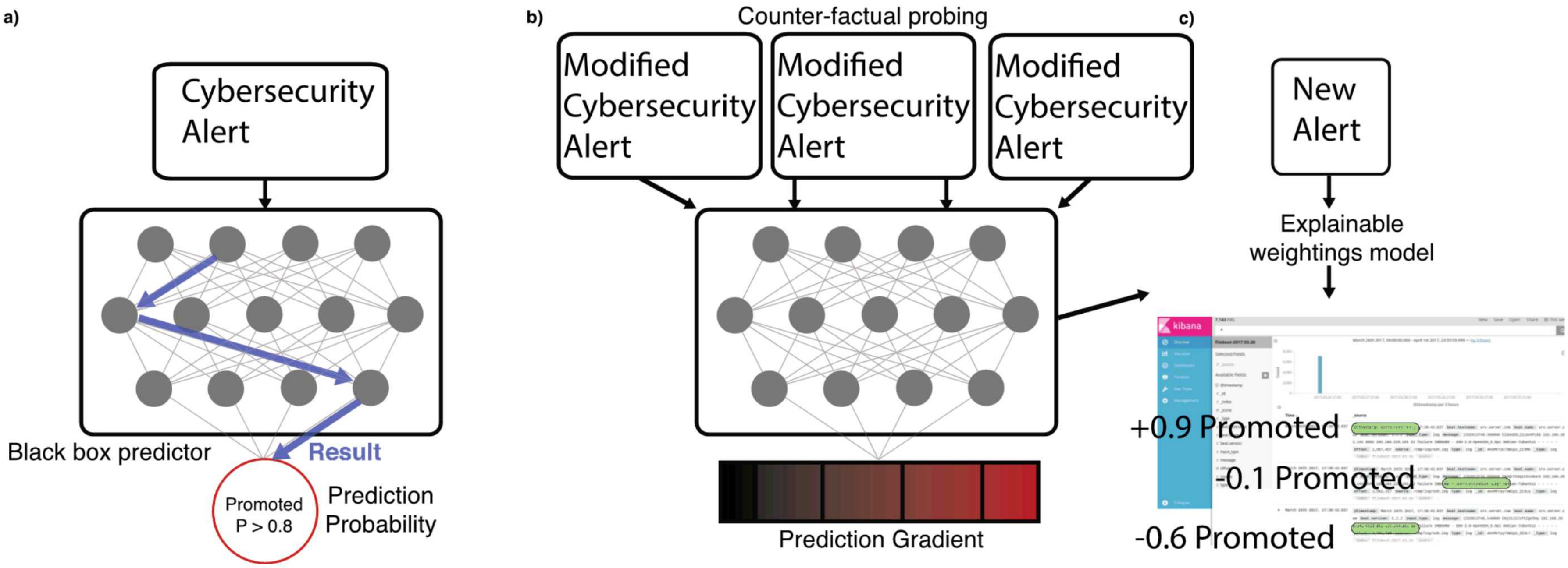
Use machine learning to classify or prioritize alerts and indicate what parts a human analyst should focus on



Explainability background

- Explainability and accuracy are frequently at odds in classification and ranking problems
- Logistic regression is imminently explainable
- $\boldsymbol{logit(C) = \beta_0 + \sum_{i=1}^{n} \beta_i X_i}$
- However…for even trivially simple examples, like MNIST (handwriting), LR scores about 93% accuracy vs record accuracies around 99.8%

State of the art: Counterfactual probing



Problem with Counterfactual Probing

1. Typically requires the model be mapped onto a linear model
2. Decision and explainability models are separate
3. Not scalable!!
4. Requires a robust sense of what randomization means for alerts

Ideal classifier

1. Highly accurate
2. Explains each instance
3. Streaming, modifiable on the fly
4. Scalable

Use our new classifier:
AWE-ML: Averaged Weights for Explainable Machine Learning

Want a Fast, Adaptable, Explainable Classifier

Find the probability of outcome $Y = 1$
given measured features $x_1 = x_1', x_2 = x_2', x_2 = x_3'$

Estimate $P(Y = 1 | \, x_1 = x_1', x_2 = x_2', x_3 = x_3')$

From training data, measure the following probabilities:

$$P(Y = 1) = 0.8$$
$$P(Y = 1 | \, x_1 = x_1') = 0.001$$
$$P(Y = 1 | \, x_2 = x_2') = 0.9$$
$$P(Y = 1 | \, x_3 = x_3') = 0.7$$
$$P(Y = 1 | \, x_1 = x_1', x_2 = x_2') = 0$$

How do we combine these to get the best overall probability estimate?

No Good 1st Principles Method

- Bayesian methods degenerate to values of 0 and 1
- One route to determine the importance of features is by averaging the probabilities

$$P(Y = 1 |\, x_1 = x_1{}', x_2 = x_2{}', x_3 = x_3{}') \approx \frac{\sum_i P(Y=1|\, x_i = x_i{}')}{\sum_i 1}$$

- Need to account for many effects:
  - Use combinations of features: $P\left(Y = 1 |\, x_i = x_i', x_j = x_j'\right)$
  - As a P approaches 0 or 1, it should be given more weight
  - Measured probabilities with less supporting data should be given less weight
  - Measured probabilities that give new information should possibly be given more weight
  - Account for class imbalance

$$P = \sum_i w_i \times P(Y = 1 |\, x_i = x_i{}') + \sum_{ij} w_{ij} \times P\left(Y = 1 |\, x_i = x_i', x_j = x_j{}'\right) + \ldots$$

Use a Hierarchical Weighted Average

Using Categorical / Binned Data

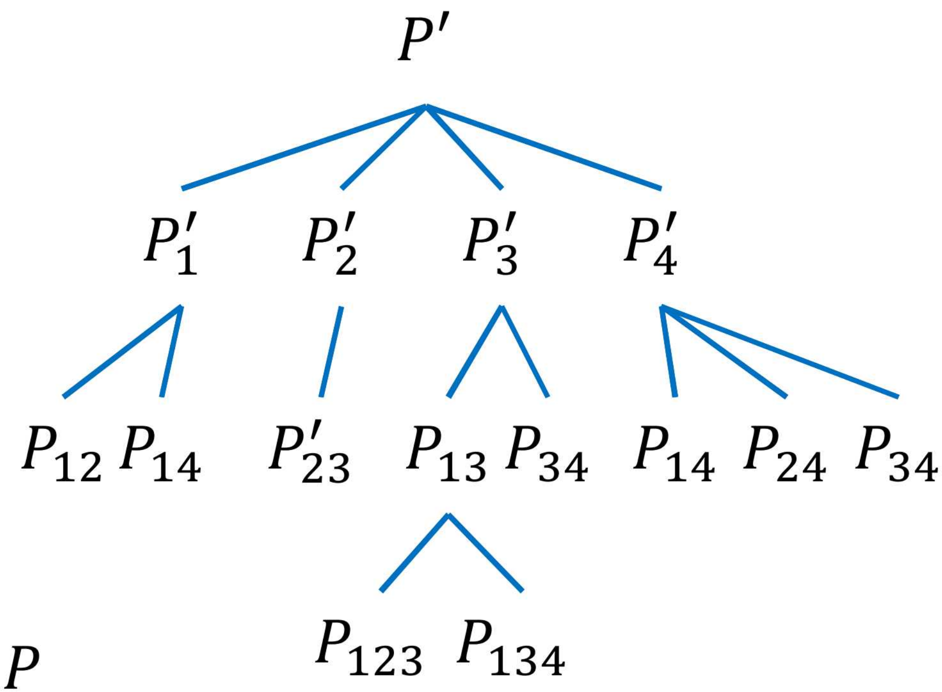
- AWE-ML requires data to be binned into categorical variables
- Binned data allows classifications to be directly tied to specific bins in the training data
- Random forests directly use continuous data averaged over different cuts of the data, preventing identification of the specific subset of the training data that is relevant to classification.
- Binning may result in a small loss in accuracy, but a gives a large improvement in explainability and adaptability.

Hierarchically Average the Probability

$$P' = \frac{\sum_i P_i'}{\sum_i 1}$$

$$P_i' = \frac{\sum_{j\neq i} P_{ij}'}{\sum_{j\neq i} 1}$$

$$P_{ij}' = \frac{\sum_{k\neq j\neq i} P_{ijk}}{\sum_{k\neq j\neq i} 1}$$



$P = P(Y = 1)$, P'= estimate of P

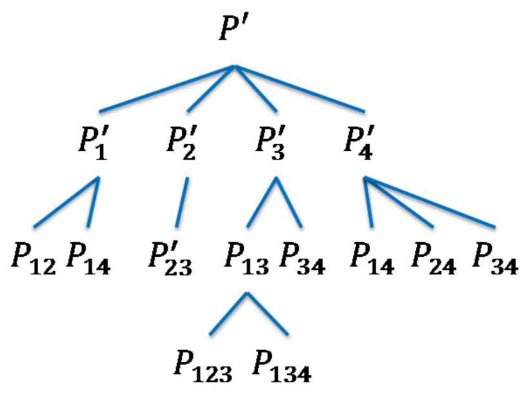
$$P_i = P(Y = 1 |\, x_i = x_i{}')$$

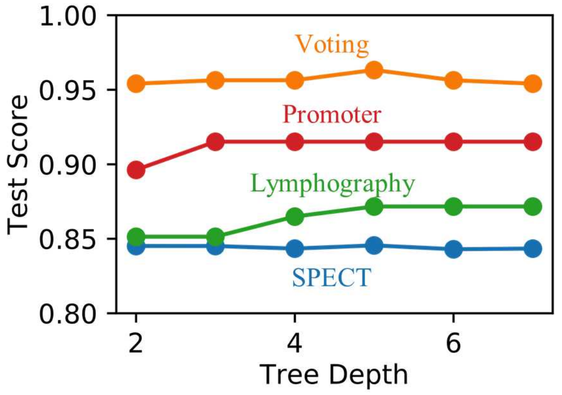
$$P_{ij} = P\left(Y = 1 |\, x_i = x_i', x_j = x_j'\right)$$

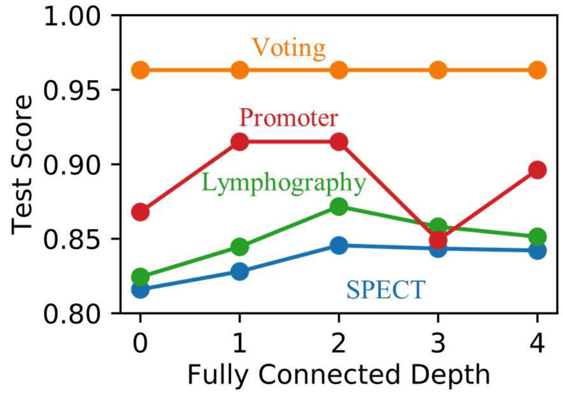
$$P_{ijk} = P\left(Y = 1 |\, x_i = x_i', x_j = x_j', x_k = x_k'\right)$$

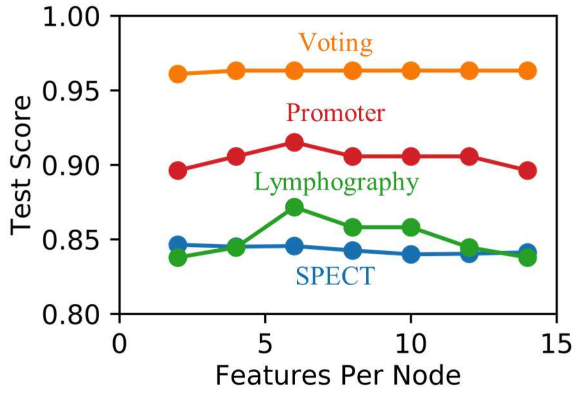
Limit Tree Size to Limit Computational Time

- Increasing the tree depth beyond 6 levels does not increase the accuracy on the tested data sets
- Only the first couple layers need to be fully connected
- After that, only the top 6 to 10 features that minimize entropy/gini coeficient need to be kept at each node







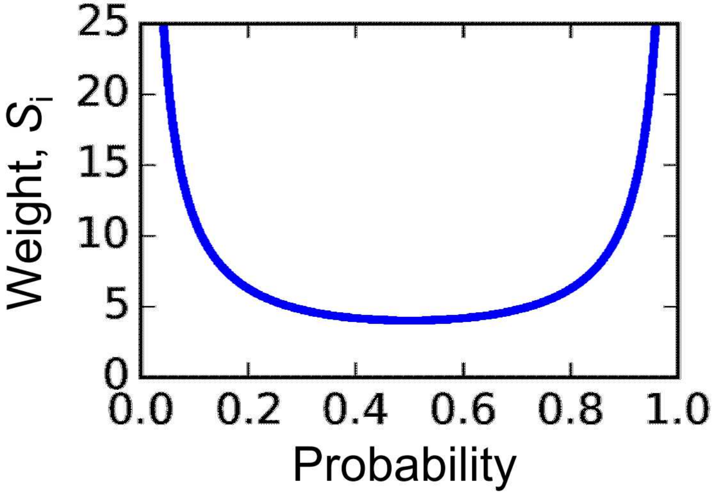


Weight P=0 and P=1 More Strongly

Features that predict Y′ more strongly should be weighted more.

$$P' = \frac{\sum_i P_i' \times S_i}{\sum_i S_i}$$

$$S_i = \frac{1}{P_i' \times (1 - P_i')}$$



Tried logarithmic scaling, but empirically does not work as well

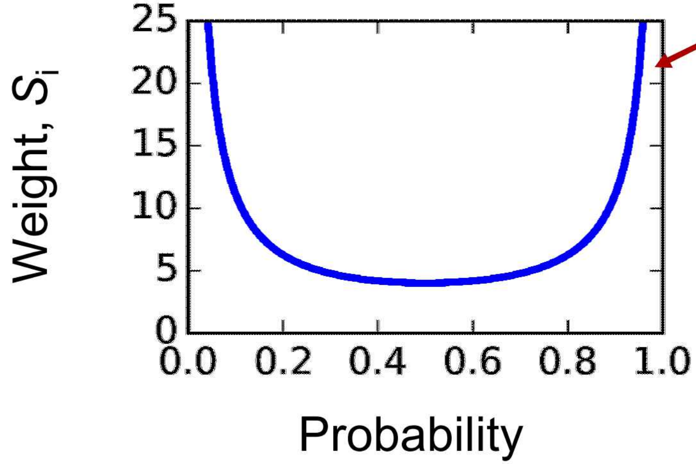
$$S_i = \frac{\log(P_i'/(1 - P_i'))}{P_i' - 0.5}$$

$P = P(Y = 1)$

$P_i = P(Y = 1 | x_i = x_i')$, P_i'=estimate of P_i

Add Weights for Noise, Class Imbalance and New Information

- Account for noise / limited data counts
  - Limit S_i
  - Scale sqrt(N_{data})



Need to avoid large weights due to noise, i.e., P_i=1 based on 1 data point

- Account for class imbalance
- Weight features that give new information / significantly change the estimated probability

Can Express as a Simple Weighted Average

$$P = \sum_i w_i \times P(Y = 1|\, x_i = x_i{}') +$$
$$\sum_{ij} w_{ij} \times P\left(Y = 1|\, x_i = x_i', x_j = x_j{}'\right) +$$
$$\sum_{ijk} w_{ijk} \times P\left(Y = 1|\, x_i = x_i', x_j = x_j{}', x_k = x_k{}'\right) +$$
$$\ldots$$

Can dynamically update model by updating probabilities.
Weights are entirely derived from the probabilities

Fitting the model means optimizing 7 hyperparameters

- nbins: Number of bins for continuous valued data
- TreeDepth, FullyConnectedDepth and FeaturesPerNode
  - Tree creation metaparameters
- n_{max}, n_{err}: Noise related metaparameters
- UsefulnessModel: usefulness model

AWE-ML is as accurate as conventional methods

Dataset	AWE-ML	Linear SVM	SVM (RBF)	Logistic Regression	Random Forests
Bank	$90.1\% \pm 0.3\%$	$90.1\% \pm 0.4\%$	$89.2\% \pm 0.1\%$	$90.1\% \pm 0.3\%$	$90.4\% \pm 0.2\%$
Breast Cancer	$96.7\% \pm 1.9\%$	$96.5\% \pm 1.4\%$	$96.9\% \pm 2.5\%$	$96.7\% \pm 2.3\%$	$95.1\% \pm 1.9\%$
Credit	$82.2\% \pm 0.3\%$	$79.7\% \pm 0.6\%$	$82.0\% \pm 0.6\%$	$81.1\% \pm 0.3\%$	$81.6\% \pm 0.4\%$
Customer	$92.0\% \pm 2.3\%$	$91.1\% \pm 2.3\%$	$92.5\% \pm 3.9\%$	$90.2\% \pm 4.9\%$	$92.2\% \pm 2.5\%$
Iris	$98.0\% \pm 3.0\%$	$96.0\% \pm 6.1\%$	$96.0\% \pm 4.4\%$	$95.3\% \pm 5.2\%$	$96.0\% \pm 4.4\%$
Lymphography	$85.5\% \pm 11.9\%$	$85.1\% \pm 12.9\%$	$83.7\% \pm 10.5\%$	$83.7\% \pm 7.8\%$	$84.4\% \pm 3.9\%$
Promoter	$94.3\% \pm 6.4\%$	$93.3\% \pm 6.0\%$	$92.4\% \pm 5.7\%$	$92.4\% \pm 7.0\%$	$91.5\% \pm 10.8\%$
Spect	$83.0\% \pm 6.0\%$	$82.0\% \pm 4.9\%$	$79.4\% \pm 1.3\%$	$79.4\% \pm 1.3\%$	$81.2\% \pm 5.7\%$
Splice	$96.4\% \pm 0.9\%$	$95.0\% \pm 1.2\%$	$86.4\% \pm 1.1\%$	$95.9\% \pm 0.9\%$	$94.6\% \pm 1.1\%$
Transfusion	$78.0\% \pm 2.7\%$	$77.1\% \pm 2.0\%$	$76.0\% \pm 0.3\%$	$76.7\% \pm 2.4\%$	$73.1\% \pm 2.7\%$
Voting Records	$96.7\% \pm 1.8\%$	$94.4\% \pm 3.5\%$	$95.1\% \pm 2.1\%$	$95.4\% \pm 3.0\%$	$95.1\% \pm 2.1\%$
Average	90.3%	89.1%	88.1%	88.8%	88.65%

Dataset	AWE-ML	Random Forests
SCOT Alerts	96.6% ± 1.3%	97.4% ± 1.0%
System Call Counts	94.5% ± 0.7%	95.0% ± 0.7%*

Hyper parameters for all classifiers optimized using 4 fold cross validation

* System Call Counts Accuracy using binned data with random forests is 94.6%

Analyze a Misclassified SCOT Instance

Predicted the alert should be closed with 83% probability

Feature 1	Feature 2	Counts Closed	Counts Promoted	Probability Closed	Weight
topic_2 = 1	time_month = 2	2166	145	94%	2.1%
time_weekday = 0	time_month = 2	1381	144	91%	1.8%
time_weekday = 0	topic_2 = 1	1149	54	96%	1.8%
time_weekday = 0	topic_3 = 1	1667	147	92%	1.7%
topic_3 = 1	time_month = 2	2817	233	92%	1.7%
time_weekday = 0	topic_1 = 1	1705	143	92%	1.7%
topic_1 = 1	time_month = 2	1987	147	93%	1.7%
time_weekday = 0	topic_0 = 1	1102	122	90%	1.5%
topic_0 = 1	time_month = 2	1702	202	89%	1.5%
corr_class_1 = 0	topic_0 = 1	6201	226	96%	1.3%
splunk_search_97 = 0	topic_0 = 1	6201	226	96%	1.3%

…

- Top feature combinations are dominated by date based features!
  - Not helpful for predicting attacks in the future
- No particular combination of features dominates

Analyze Individual Features

Feature	Weighted Probability	Original Probability	Weight
time_month = 2	83%	97%	10.94%
time_weekday = 0	82%	97%	10.74%
topic_0 = 1	76%	96%	8.90%
topic_4 = 9	68%	96%	7.87%
time_day = 17	57%	88%	7.77%
topic_2 = 1	92%	98%	7.71%
topic_3 = 1	87%	97%	7.65%
topic_1 = 1	86%	98%	7.46%
corr_class_0 = 0	92%	99%	5.20%
time_year = 0	92%	99%	5.19%
corr_class_unknown = 0	92%	99%	5.19%
splunk_search_97 = 0	93%	99%	5.19%
corr_class_1 = 0	93%	99%	5.13%
corr_class_None = 0	94%	99%	5.06%

Probability given all feature combinations containing the specified feature

Probability given a single feature

Eliminate Date Based Features and Analyze a Correctly Classified Instance

Feature 1	Feature 2	Feature 3	Count Closed	Counts Promoted	Weight
http_user_agent_perl = 0			2240	0	66.5%
http_method_post = 0	topic_3 = 3		1508	0	6.6%
http_method_post = 0	topic_1 = 2		917	0	5.9%
http_method_post = 0	topic_2 = 2		1158	0	5.0%
http_method_post = 0	topic_0 = 7		1114	0	5.0%
http_method_post = 0	topic_4 = 2		813	0	4.5%
http_response_body_length = 4			1038	0	3.8%
http_request_body_length = 6			622	0	2.2%
topic_1 = 2	http_status_code_404 = 0	topic_3 = 3	441	0	0.3%

…

A single feature dominated the classification and we can see why, all 2,240 times it appeared in the training data, the alert was closed

Analyze another misclassified instance

Incorrectly Predicted the alert should be closed with 99.5% probability

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Counts Closed	Counts Promoted	Prob. Closed	Weight
topic_1 = 1	http_status_code_200 = 0	topic_4 = 1	topic_2 = 1	topic_3 = 1	665	1	99.8%	7.5%
topic_1 = 1	http_status_code_200 = 0	topic_4 = 1	topic_3 = 1	topic_0 = 8	726	1	99.9%	5.0%
topic_4 = 1	corr_class_None = 0				3219	7	99.8%	2.7%
splunk_search_97 = 0	topic_4 = 1				3219	7	99.8%	2.7%
corr_class_1 = 0	topic_4 = 1				3219	7	99.8%	2.7%
topic_4 = 1	corr_class_unknown = 0				3219	7	99.8%	2.7%
corr_class_0 = 0	topic_4 = 1				3219	7	99.8%	2.7%
topic_1 = 1	http_status_code_200 = 0	topic_4 = 1	topic_3 = 1	corr_class_unknown = 0	732	1	99.9%	2.3%
http_method_post = 0	corr_class_unknown = 0				9133	24	99.7%	2.1%

…

There is always at least one case promoted for every feature combination. Could we create new metrics to find rare events?

Analyze Misclassified FARM System Call

Incorrectly Predicted Malware with 78.9% probability

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Count Good	Count Bad	Weight
CreateEvent=1	FreeVirtMem=1	OpenSection=2	SetInfoFile=2			0	3	8.4%
CreateEvent=1	QryDirFile=0	QrySysInfo=1	SetInfoThread=1	UnmapViewOfSec=4	UsrRegWindMsg= 1	84	0	6.6%
CreateEvent=1	QryDirFile=0	QryInfoFile=4	QrySysInfo=1	SetInfoThread=1	UsrRegWindMsg=1	50	0	4.1%
CreateSection=4	QryAttributesFile=4	RqstWaitReplyPrt=4	WaitForMultObj=1			0	283	4.1%
CreateThread=0	OpenSection=2	QrySection=3	SetInfoFile=2	UsrRegWindMsg=1		0	25	3.2%
OpenSection=2	ReadFile=3	SetInfoFile=2	SysCall#240=0	SysCall#326=0		0	8	2.9%
OpenSection=2	ReadFile=3	SetInfoFile=2	SysCall#240=0	SysCall#264=0		0	8	2.6%
CreateEvent=1	QryDefaultLocale=1	QryInfoJobObj=0	QryKey=0	SysCall#92=1	SetInfoFile=2	0	5	2.5%
OpenSection=2	OpenThreadTkn=1	QryInfoJobObj=0	SetInfoFile=2	SysCall#172=0		0	2	2.5%
CreateFile=3	CreateSection=4	RqstWaitReplyPrt=4	WaitForOneObj=0			0	5	2.4%
FreeVirtMem=1	FsControlFile=2	RqstWaitReplyPrt=4	SysCall#249=1			0	8	2.3%

…

There is contradictory information resulting in the misclassification

Summary

- Matches state of the art machine learning accuracy
  - Identical on open datasets, within 1% on cyber datasets
- Fully explainable
- Coded in Cython for speed
- Compatible with Sci-kit Learn

$$P = \sum_i w_i \times P(Y = 1 |\ x_i = x_i{}') + \\ \sum_{ij} w_{ij} \times P\left(Y = 1 |\ x_i = x_i', x_j = x_j{}'\right) + \\ \cdots$$

- To do:
  - Further improve accuracy, automate meta parameter settings
  - Simplify explanations
  - Handle continuous valued data
  - Parallelize code
  - Adapt to new data without retraining
  - Extend to Regression

Backup/Extra Slides

Lets look at the importance by feature

Incorrectly Predicted Malware with 78.9% probability

Feature	Weighted Probability	Original Probability	Weight
NtSetInformationFile = 2	96.7%	19.7%	11.1%
NtOpenSection = 2	99.8%	27.4%	8.4%
NtCreateEvent = 1	60.4%	29.7%	5.3%
NtReadFile = 3	95.3%	20.6%	3.9%
NtFreeVirtualMemory = 1	96.6%	48.6%	3.9%
NtUserRegisterWindowMessage = 1	46.2%	16.5%	3.7%
NtSetInfoThread = 1	47.9%	27.4%	3.6%
NtRequestWaitReplyPort = 4	99.3%	76.2%	3.2%
NtQueryDirectoryFile = 0	35.5%	43.0%	2.9%
NtFsControlFile = 2	91.1%	21.7%	2.9%
NtQuerySystemInformation = 1	3.8%	26.7%	2.0%
NtOpenThreadToken = 1	98.0%	31.2%	1.9%
NtCreateSection = 4	95.1%	67.8%	1.9%
NtQueryInformationFile = 4	54.1%	22.1%	1.8%
NtUnmapViewOfSec = 4	31.5%	34.0%	1.7%
NtCreateFile = 3	67.1%	22.3%	1.6%
SysCall#326 = 0	86.3%	65.1%	1.6%
NtMapViewOfSec = 3	69.5%	32.3%	1.6%
SysCall#240 = 0	98.0%	39.7%	1.5%

…

Probability given all feature combinations containing the specified feature

Top 20 most important of 384 features

May be able to map the key system calls back to the part of the file that is malware

Accounting for feature combinations significantly changes the probability

Analyze Correctly Classified FARM System Call

Correctly Predicted Malware with 99.9998% probability

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Count Good	Count Bad	Weight
FsControlFile=4	MapViewOfSec=4	OpenProcTokn=1	QryAttribFile=4	SetInfoProcess=4	WaitForMultObj=0	0	288	13.1%
MapViewOfSec=4	OpenKey=4	SysCall#92=1	SetInfoThread=2	WaitForOneObj=3		0	169	3.8%
OpenKey=4	OpenSection=3	SysCall#92=1	SetInfoThread=2			0	133	3.8%
CreateFile=4	CreateMutant=3	FsControlFile=4	MapViewOfSec=4	UnmapViewOfSec=4	WaitForMultObj=0	0	125	3.4%
FlushInstructCache=1	FsControlFile=4	MapViewOfSec=4	QryAttribFile=4	WaitForOneObj=3		0	334	2.7%
SysCall#0=0	AccessCheck=0	CreateSection=4	SetInfoThread=2			0	315	2.5%
CreateSection=4	OpenProcTokn=1	SetInfoThread=2	UnmapViewOfSec =4	WaitForMultObj=0	UsrValidateSecure=0	0	216	2.5%
CreateSection=4	OpenDirObject=2	OpenProcTokn=1	SetInfoThread=2	WaitForMultObj=0	UsrValidateSecure=0	0	219	2.3%
SysCall#0=0	CreateSection=4	SetInfoThread=2	WaitForOneObj=3	UserSysParamInfo=0		0	333	2.2%
CreateFile=4	MapViewOfSec=4	QueryValueKey=4	WriteVirtMem=0			0	328	1.9%
MapViewOfSec=4	OpenKey=4	SysCall#92=1	SetInfoThread=2	WaitForMultObj=0		0	79	1.8%

By Feature

Sandia National Laboratories

Correctly Predicted Malware with 99.9998% probability

Feature	Weighted Probability	Original Probability	Weight
NtMapViewOfSection = 4	100.0%	82.3%	8.8%
NtWaitForMultipleObjects = 0	100.0%	37.4%	8.1%
NtFsControlFile = 4	100.0%	66.6%	7.2%
NtSetInformationThread = 2	100.0%	88.5%	7.2%
NtWaitForSingleObject = 3	100.0%	40.8%	5.2%
NtOpenKey = 4	100.0%	51.4%	5.1%
NtOpenProcessToken = 1	100.0%	34.4%	4.4%
NtQueryAttributesFile = 4	100.0%	55.3%	3.4%
SysCall#92 = 1	100.0%	32.0%	3.2%
NtSetInformationProcess = 4	100.0%	59.9%	3.2%
NtCreateSection = 4	100.0%	67.8%	2.7%
NtCreateFile = 4	100.0%	28.5%	2.4%
NtQuerySection = 4	100.0%	78.3%	2.1%
NtUserSystemParametersInfo = 0	100.0%	71.0%	2.0%
NtUnmapViewOfSection = 4	100.0%	34.0%	1.8%
NtCreateEvent = 4	100.0%	72.4%	1.6%
SysCall#0 = 0	100.0%	29.5%	1.4%
NtOpenSection = 3	100.0%	44.0%	1.4%
NtOpenDirectoryObject = 2	100.0%	49.8%	1.3%

…

Probability given all feature combinations containing the specified feature

Probability given a single feature

Top 20 most important of 384 features

Use Model to Analyze a Misclassified Result

1984 Congressional Voting Records Dataset

The classifier predicted with 70% probability that this Member of Congress would be a Republican when they are a Democrat.

Features	Counts Rep.	Counts Dem.	Probability Rep.	Weight	Cumulative Weight
Immigration-Y, South Africa Export Act-N	16	0	100%	19.4%	19.4%
Doc Fee-Y, Mx Missile-N, Immigration-Y, Duty Free-N	47	0	100%	4.8%	24.2%
Doc Fee-Y, Contras aid-N, Immigration-Y, Duty Free-N	47	0	100%	4.7%	28.9%
Adopt Budget-Y, Synfuels cutback-Y, Education-N	0	58	0%	4.0%	32.9%
Water-Y, Adopt Budget-Y, Synfuels cutback-Y,	0	38	0%	3.1%	36.0%

Feature #	Probability Republican	Weight	Feature #	Probability Republican	Weight
Immigration-Y	95%	17%	Mx missile-N	86%	4%
Doc fee freeze-Y	94%	16%	Nicaraguan contra aid – N	83%	4%
South Africa Export Admin Act-N	94%	14%	Anti-satellite test ban-N	29%	3%
Adopt Budget-Y	14%	8%	Handicapped infants-N	78%	3%
Synfuels corporation cutback-Y	23%	8%	Crime-Y	55%	2%
Education spending-N	21%	6%	El Salvador aid-Y	69%	2%
Duty free exports-N	89%	5%	Superfund right to sue-Y	52%	2%
Water project cost sharing-Y	53%	4%	Religious groups in schools-Y	56%	2%

Republican features have higher certainty and therefore higher weight

Probability given all feature combinations containing the specified feature

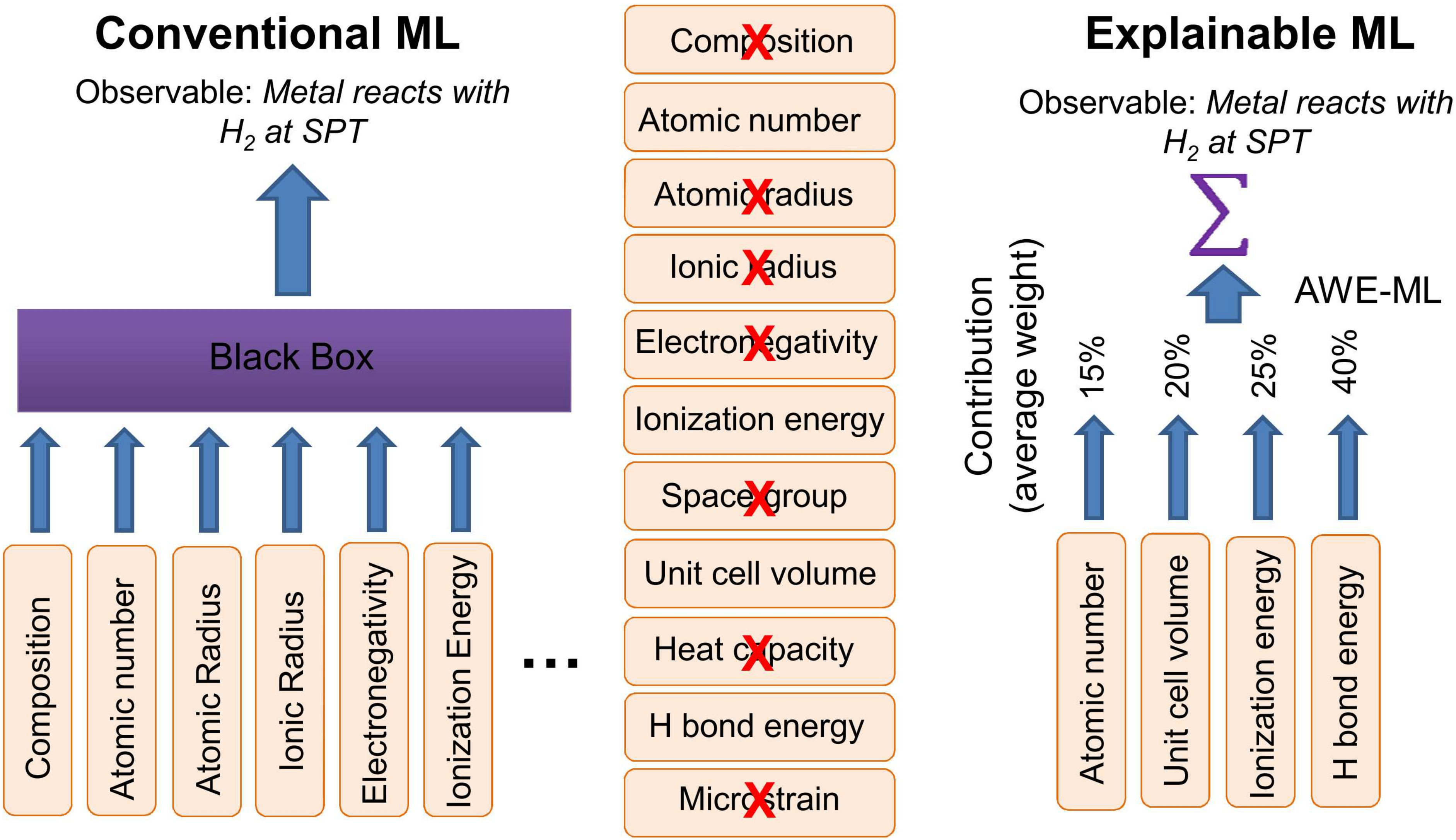
Analyze a Correctly Classified Result

1984 Congressional Voting Records Dataset

The classifier correctly predicted this member of congress is a republican

Features	Counts Rep.	Counts Dem.	Probability Rep.	Weight	Cumulative Weight
Budget -N, Doc Fee-Y, Immigration-Y	54	0	100%	18.3%	18.3%
Doc Fee-Y, Immigration-Y, Education –Y	57	0	100%	16.1%	34.4%
Immigration-Y, Education-Y, SA Export-N	16	0	100%	5.0%	39.4%
Water-N, Doc Fee-Y, Immigration-Y	31	0	100%	3.0%	42.4%
Water-N, Immigration-Y, SA Export-N	6	0	100%	0.9%	43.3%
Handicap-N, Budget-N, Doc Fee-Y, MxMissile-N, Superfund-Y	76	1	98.7%	0.5%	43.8%

Conventional vs explainable machine learning

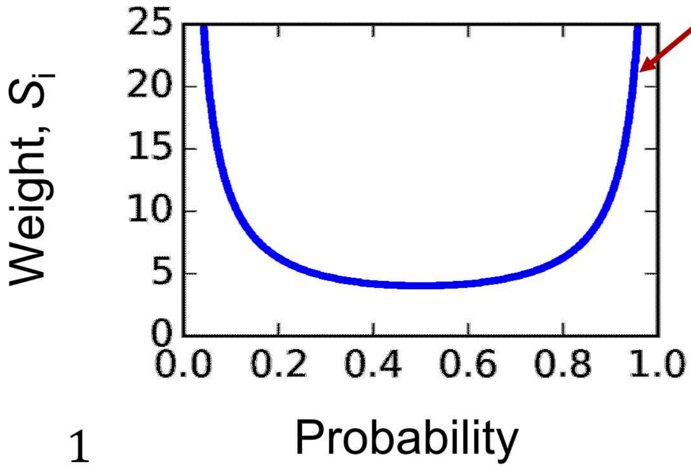


⇒ ***The EML framework has the potential to address exceedingly complex interrelationships among features that defy scientific intuition to extract structure-property relationships***

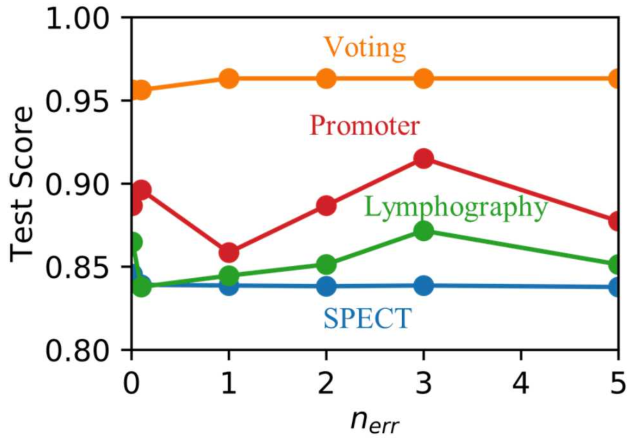
Account for Noise: Regularization

- Each probability estimate, P_i, P_{ij}, etc is backed by some number of training examples, n.
- Limit P=0/1 scaling based on n.
  - Limit P_i used to calculate S_i to $\left[\frac{n_{err}}{n}, 1 - \frac{n_{err}}{n}\right]$
    - n_{err} is a meta-parameter around 1, represents average number of examples that might be wrong due to noise
    - If n<n_{err}, set $P_i = 0.5$

Need to avoid large weights due to noise, i.e., $P_i = 1$ based on 1 data point



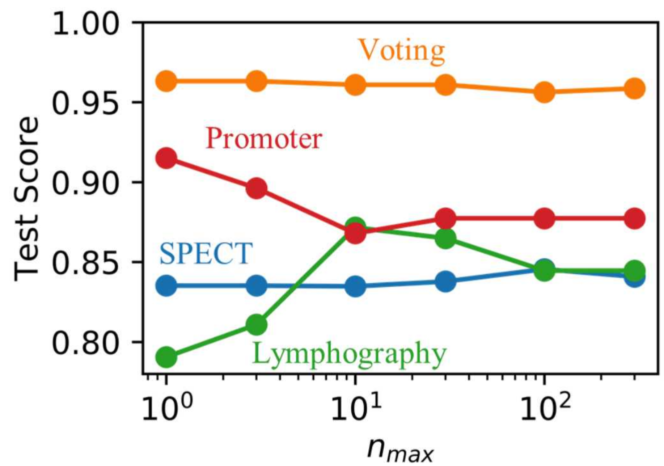
$$S_i = \frac{1}{P_i' \times (1 - P_i')}$$



Additionally add a weight based on the number of samples

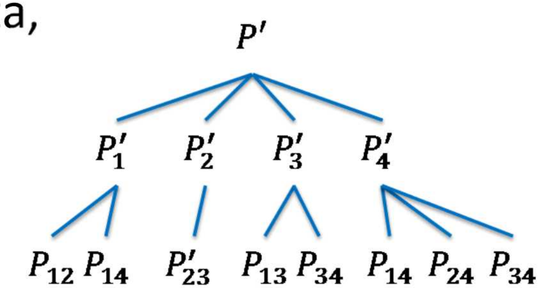
- Each probability estimate, P_i, P_{ij}, etc is backed by a some number of training examples, n.
- Determine a weight based on n
  - Define n_{max} = # of training examples to needed to overcome noise
  - Scaling factor = $N_i = \min(\sqrt{n_i}, \sqrt{n_{max}})$
  - Noise tends to scale with variance, proportional to sqrt (N)

$$P' = \frac{\sum_i P_i' \times S_i \times N_i}{\sum_i S_i \times N_i}$$



- If none of the probabilities are backed by significant data, use the parent probability:

$$P' = \frac{\max_i(\mathrm{N_i})}{N} \times \frac{\sum_i P_i' \times S_i \times N_i}{\sum_i S_i \times N_i} + \left(1 - \frac{\max_i(\mathrm{N_i})}{N}\right) \times P$$



Account For Class Imbalance

- Allow rare classes to have larger weights to compensate for lower counts

Previously

Limit P_i used to calculate S_i to $\left[\frac{n_{err}}{n}, 1 - \frac{n_{err}}{n}\right]$

$$P' = \frac{\sum_i P_i' \times S_i}{\sum_i S_i}$$

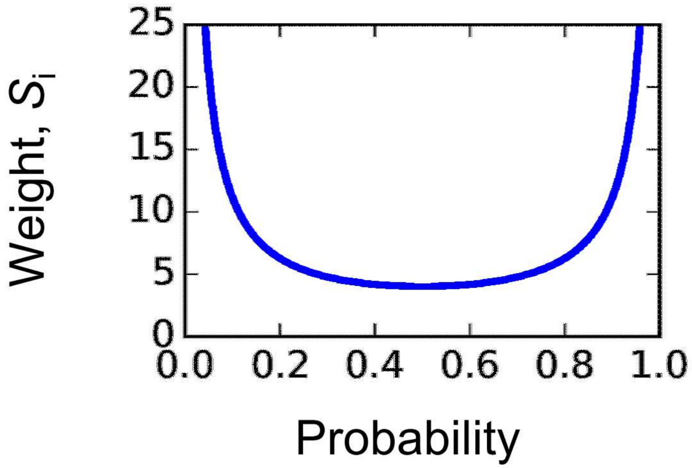
$$S_i = \frac{1}{P_i' \times (1 - P_i')}$$

Multiply by $1 - P(Y = Y')$ Multiply by $P(Y = Y')$

Now

$$P_{i,max} = \max\left(1 - \left(P(Y = Y')\frac{n_{err}}{n}\right), 0.5\right)$$

$$P_{i,min} = \min\left((1 - P(Y = Y')) \times \frac{n_{err}}{n}, 0.5\right)$$



Add a weight based on usefulness of a feature

- Estimate usefulness by how much each feature changes the probability around the decision function. Three possible models for usefulness are:

None

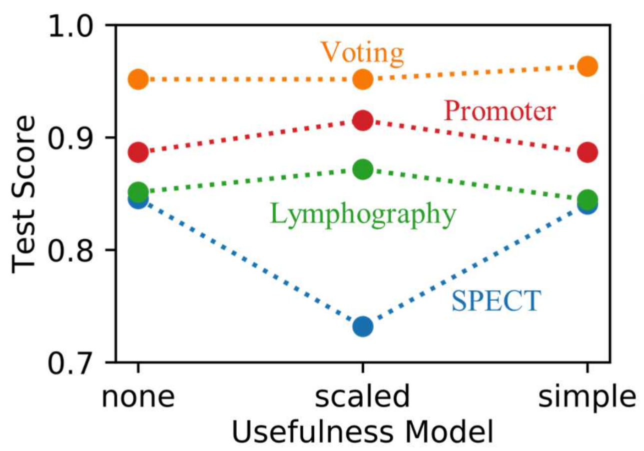
$$U_i = 1$$

Simple

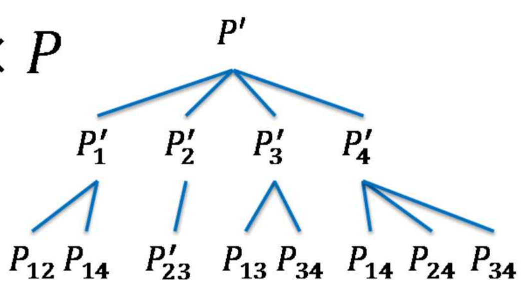
$$U_i = \frac{P_i' - P}{P_i' + P}$$

Scaled

$$U_i = |S_i \times (P_i' - 0.5) - S_o \times (P - 0.5)|$$



$$P' = \frac{\max(\mathrm{N_i})}{N} \times \frac{\sum_i P_i' \times S_i \times N_i \times U_i}{\sum_i S_i \times N_i \times U_i} + \left(1 - \frac{\max(\mathrm{N_i})}{N}\right) \times P$$



Thus the overall model is:

$$P' = \frac{\max_{i}(\mathrm{N_i})}{N} \times \frac{\sum_i P_i' \times S_i \times N_i \times U_i}{\sum_i S_i \times N_i \times U_i} + \left(1 - \frac{\max_{i}(\mathrm{N_i})}{N}\right) \times P$$

$$P_i' = \frac{\max_{j}(\mathrm{N_{ij}})}{N_i} \times \frac{\sum_{j \neq i} P_{ij}' \times S_{ij} \times N_{ij} \times U_{ij}}{\sum_{j \neq i} S_{ij} \times N_{ij} \times U_{ij}} + \left(1 - \frac{\max_{j}(\mathrm{N_{ij}})}{N_i}\right) \times P_i$$

$$P_{ij}' = \frac{\max_{k}(\mathrm{N_{ijk}})}{N_{ij}} \times \frac{\sum_{k \neq j \neq i} P_{ijk}' \times S_{ijk} \times N_{ijk} \times U_{ijk}}{\sum_{k \neq j \neq i} S_{ijk} \times N_{ijk} \times U_{ijk}} + \left(1 - \frac{\max_{k}(\mathrm{N_{ijk}})}{N_{ij}}\right) \times P_{ij}$$

Where:

$$P = P(Y = 1)$$

$$P_i = P(Y = 1 |\, x_i = x_i')$$

$$P_{ij} = P\left(Y = 1 |\, x_i = x_i', x_j = x_j'\right)$$

$$P_{ijk} = P\left(Y = 1 |\, x_i = x_i', x_j = x_j', x_k = x_k'\right)$$

$$S_i = \frac{1}{P_i' \times (1 - P_i')}$$

$$U_i = \frac{P_i' - P}{P_i' + P}$$

$$\mathrm{N_i} = \min(\sqrt{n_i}, \sqrt{n_{max}})$$