Learning to Knowledge Discovery to Action in Distribution Sensitive Scenarios

Nitesh V. Chawla
University of Notre Dame
http://www.nd.edu/~nchawla
nchawla@nd.edu

Data, Inference, Analysis
and Learning Lab @ ND

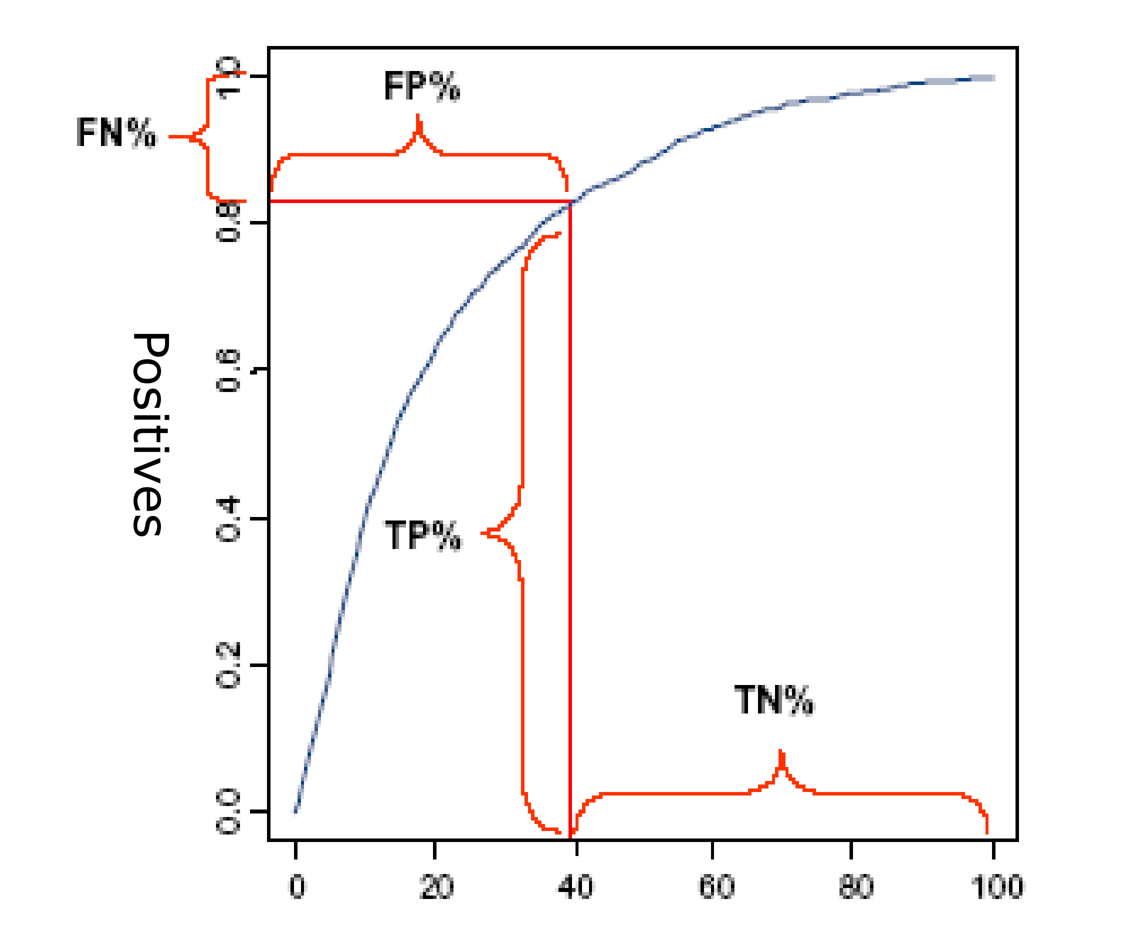
That needle in the Haystack

The statistic of rare event

Data set is imbalanced, if the classes are unequally distributed

Class of interest (minority class) is often much infrequent or rarer

But, the cost of error on the minority class can have a bigger bite



	Actual Negative	Actual Positive
Predict Negative	TN	FN
Predict Positive	FP	TP

Typical Prediction Model

Cost and Benefits

	Actual Negative	Actual Positive
Predict Negative	b00	b01
Predict Positive	b10	b11

	Actual Negative	Actual Positive
Predict Negative	TN	FN
Predict Positive	FP	TP

$$B_N = (1 - P_k)b_{00} + P_k b_{01}$$

$$B_P = (1 - P_k)b_{10} + P_k b_{11}$$

Costs

Benefit of Non-Default

$$b_{00}(k,x)(1-P_k) > (1-P_k)b_{10} + P_k b_{11} - P_k b_{01}(x)$$

$$b_{00}(k,x) > \frac{(1-P_k)b_{10} + P_k b_{11} - P_k b_{01}(x)}{(1-P_k)}$$

$$\boxed{\therefore NPV = (1-P_k)b_{00} - (1-P_k)b_{01} + P_k b_{11} - P_k b_{10}}$$

$$\equiv (1-P_k)b(TN) - (1-P_k).C(FP) + P_k.b(TP) - P_k.C(FN)$$

Liu and Chawla, "Benefit Scoring for Pricing," *KDD* 2007

Paradox of False Positive

- Imagine a disease that has a prevalence of 1 in a mllion people. I invent a test that is 99% accurate. I am obviously excited. But, when applied to a million, it returns positive for 10,000 (remember, it is 99%accurate). Priors tell us otherwise. There is one in a million infected --- 99% accurate test is inaccurate 9,999 times out of 10,0000.

The one in a 100, one in a 1000, one in 100,000, and one in a million event

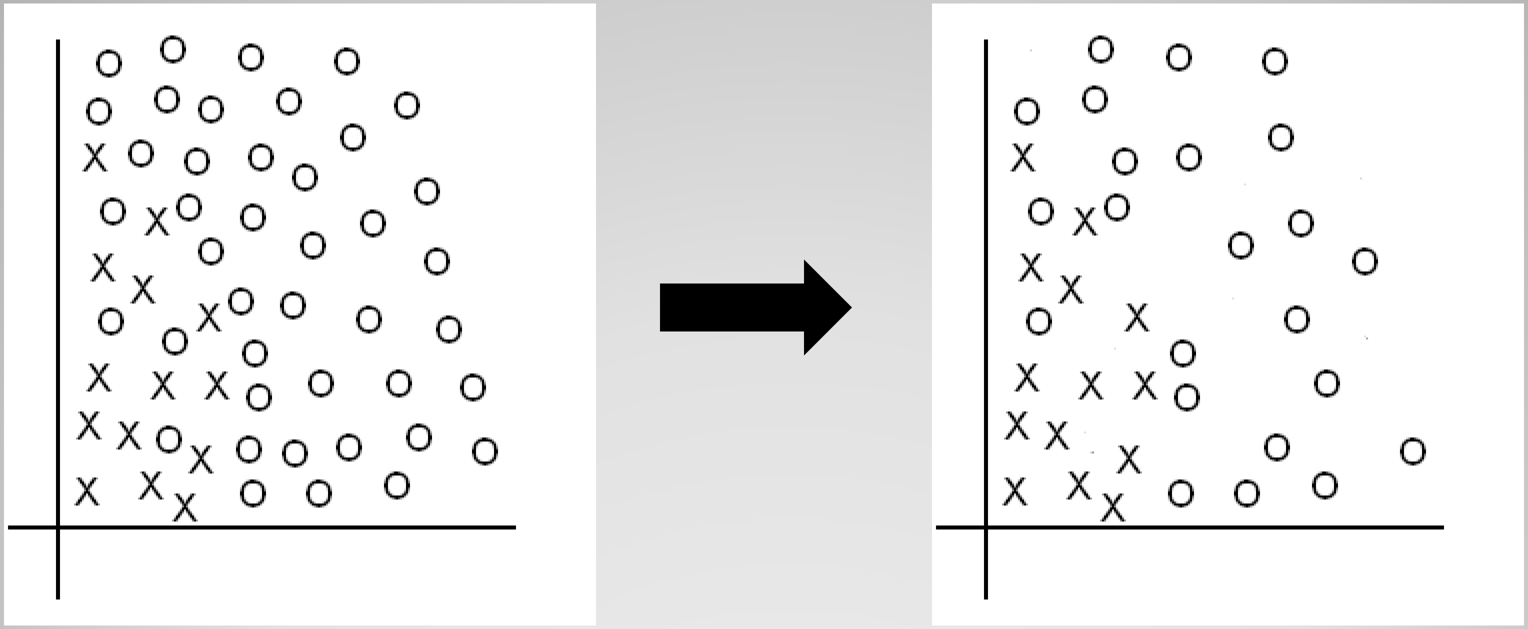
- Real-world has abundance of scenarios with such imbalance in class distributions
  - Fraud detection
  - Fault detection and prediction
  - Failures
  - Disease prediction
  - Intrusion detection
  - Text categorization
  - Bioinformatics
  - Direct marketing
  - Terrorism
  - Physics simulations

Typical Solutions

- Sampling Methods
- Moving Decision Threshold
- Classifiers' Objective Functions

Undersampling

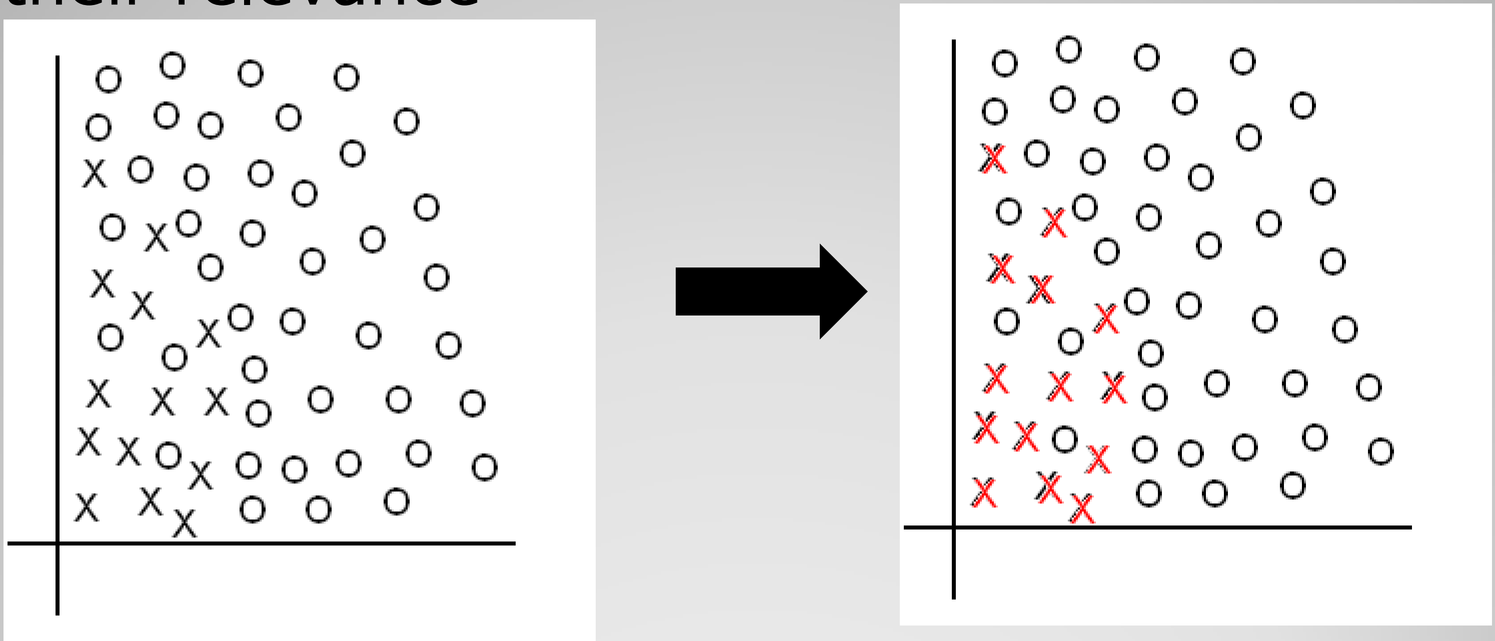
- **Randomly remove majority class examples**



Risk of losing potentially important majority class examples, that help establish the discriminating power

Oversampling

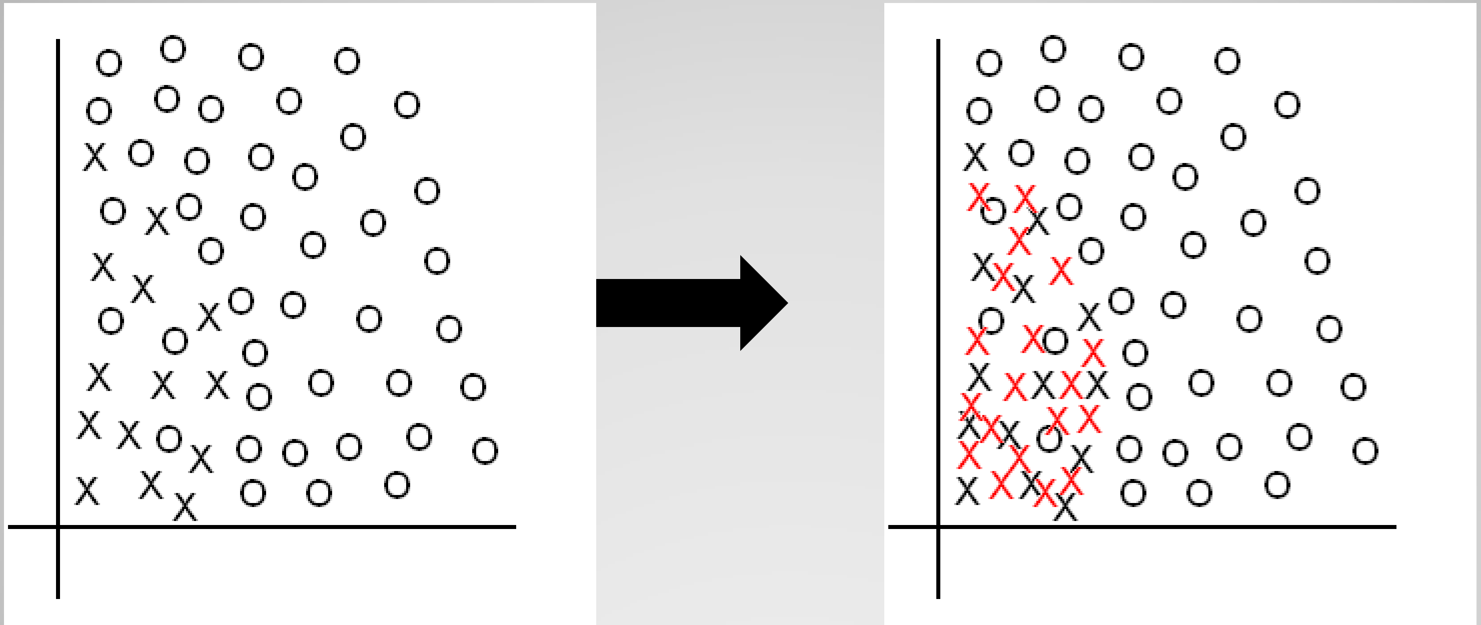
- Replicate the minority class examples to increase their relevance



But no new information is being added. Hurts the generalization capacity.

Instead of replicating, let us invent some new instances

- SMOTE: Synthetic Minority Over-sampling Technique



- Conclusions from Sampling Work:
  - When faced with the problem of class imbalance, SMOTE and undersampling, is generally the preferred combination.
  - Using a wrapper can effectively discover the potentially optimally amounts of sampling.
  - Effectively countering imbalance counters misclassification costs issues

Chawla, et al., "SMOTE: Synthetic Minority Oversampling Technique, *Journal of Artificial Intelligence Research,*

Cieslak, Chawla, "Start Globally, Optimize Locally, and Predict Globally: Improving Performance on Imbalanced Data," *IEEE International Conference on Data Mining (ICDM),* 2007

Chawla et al., "Automatically countering class imbalance and its empirical relationship to cost, *Data Mining and Knowledge Discovery Journal*, 2009

- Sampling approaches can be computationally expensive
- Outstanding Question: *Can we improve baseline classifier performance?*

Beyond Sampling

- Traditional decision tree splitting criteria are typically class skew sensitive
  - Almost always need some sampling or threshold moving
  - Ensemble methods can potentially mitigate but can be limited

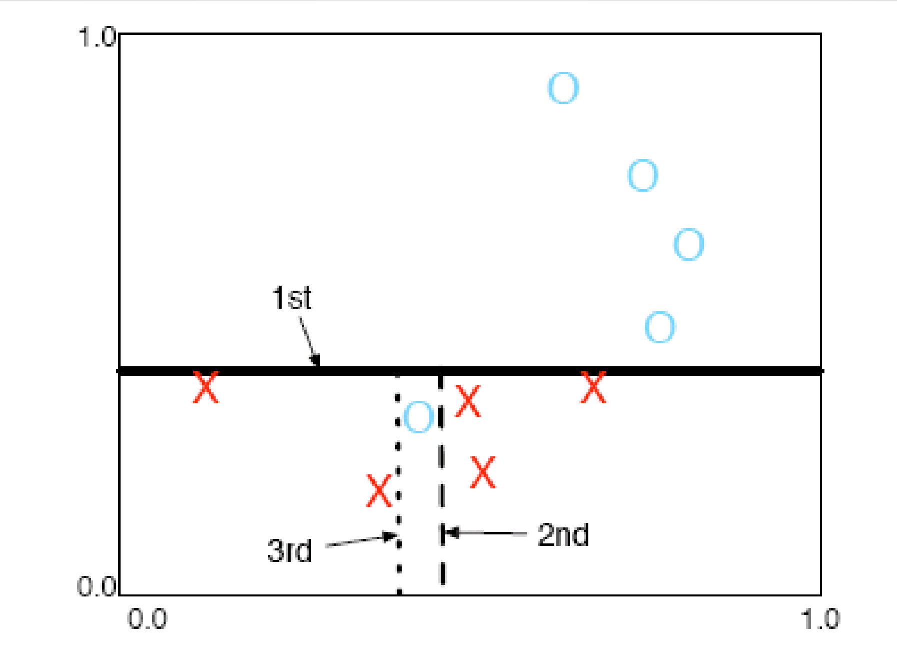
Looking at Decision Trees

Decision Trees

- A popular choice when combined with sampling or moving threshold to counter the problem of class imbalance
- The leaf frequencies converted to probability estimates (Laplace or m-estimate smoothing applied, typically)
  - Suggested use is as a PET – Probability Estimation Trees (unpruned, no-collapse, and Laplace)

Decision tree (im)purity metrics

Partition feature space to maximize purity at leaves. Recurse



Entropy (Information Gain) as an impurity

(Q,W) classes of interest

N = number of samples

N_i = number of samples in class i

N^S = number of samples in L/R

N_i^S = number of samples in class i isL/R split

$$E = \sum_{i \in (W,Q)} -\frac{N_i^L}{N^L} \log_2 \frac{N_i^L}{N^L} + \sum_{i \in (W,Q)} -\frac{N_i^R}{N^R} \log_2 \frac{N_i^R}{N^R}$$

Consider a skew insensitive criterion

- Hellinger Distance
  - distance between probability measures independent of the dominating parameters

Properties of Hellinger Distance

$$d_H(P,Q) = \sqrt{\int_\Omega (\sqrt{P} - \sqrt{Q})^2 d\lambda}$$

$$d_H(P,Q) = \sqrt{\sum_{\phi \in \Phi} (\sqrt{P(\phi)} - \sqrt{Q(\phi)})^2}$$

- Measures countable space Φ
- Ranges from 0 to √2
- Symmetric: $d_H(P,Q)=d_H\,(Q,P)$
- Lower bounds KL divergence

Hellinger as decision tree splitting criterion

$$d_H = \sqrt{(\sqrt{P(L|+)} - \sqrt{P(L|-)})^2 + (\sqrt{P(R|+)} - \sqrt{P(R|-)})^2}$$

$$d_H = \sqrt{2 - 2\sqrt{P(L|+)P(L|-)} - 2\sqrt{P(R|+)P(R|-)}}$$

Inf. Gain vs. Hellinger distance

(Q, W) classes of interest

N

N_i = number of samples in class i

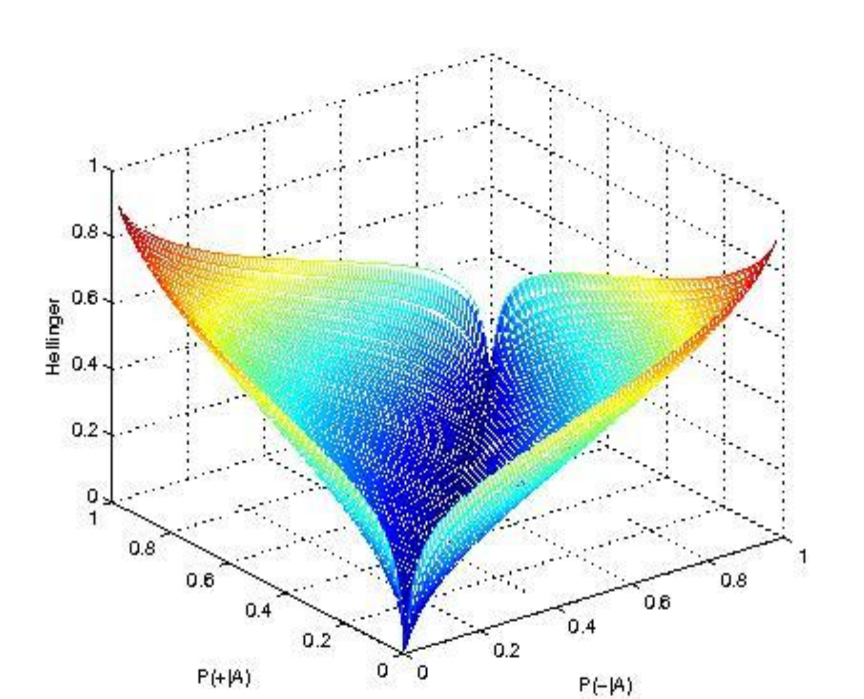
N^S = number of samples in L/R

N_i^S = number of samples in class i is L/R split

$$E = \sum_{i \in (W,Q)} -\frac{N_i^L}{N^L}\log_2\frac{N_i^L}{N^L} + \sum_{i \in (W,Q)} -\frac{N_i^R}{N^R}\log_2\frac{N_i^R}{N^R}$$

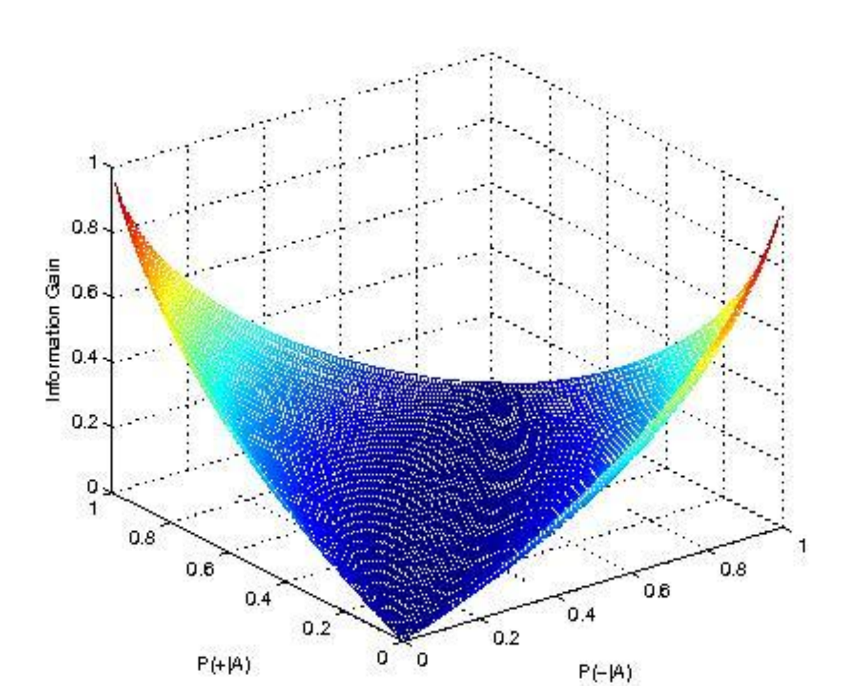
$$H = \sqrt{\left\{\sqrt{\frac{N_Q^L}{N_Q}} - \sqrt{\frac{N_W^L}{N_W}}\right\}^2 + \left\{\sqrt{\frac{N_Q^R}{N_Q}} - \sqrt{\frac{N_W^R}{N_W}}\right\}^2}$$

Comparing Value Surfaces



P(x|+) P(x|-)

Hellinger Distance

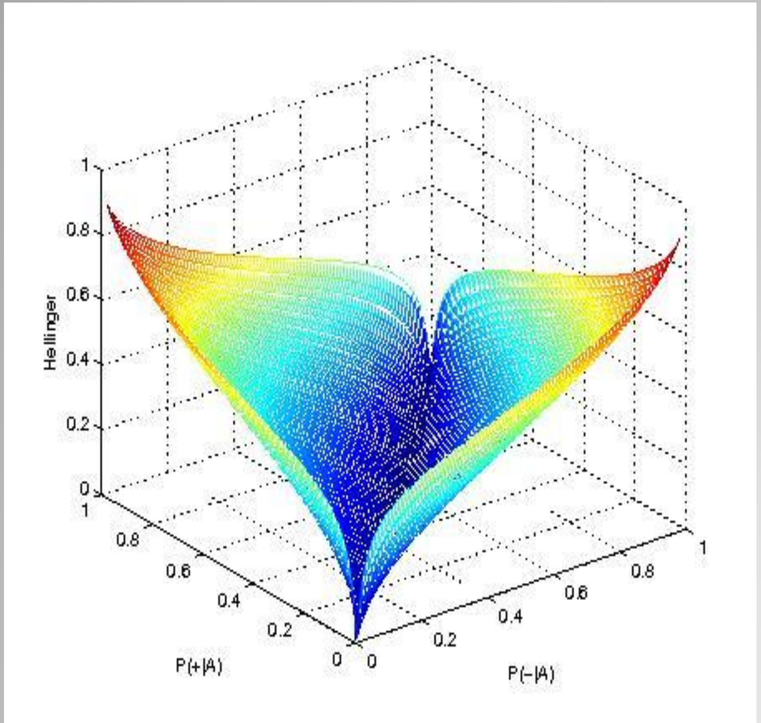


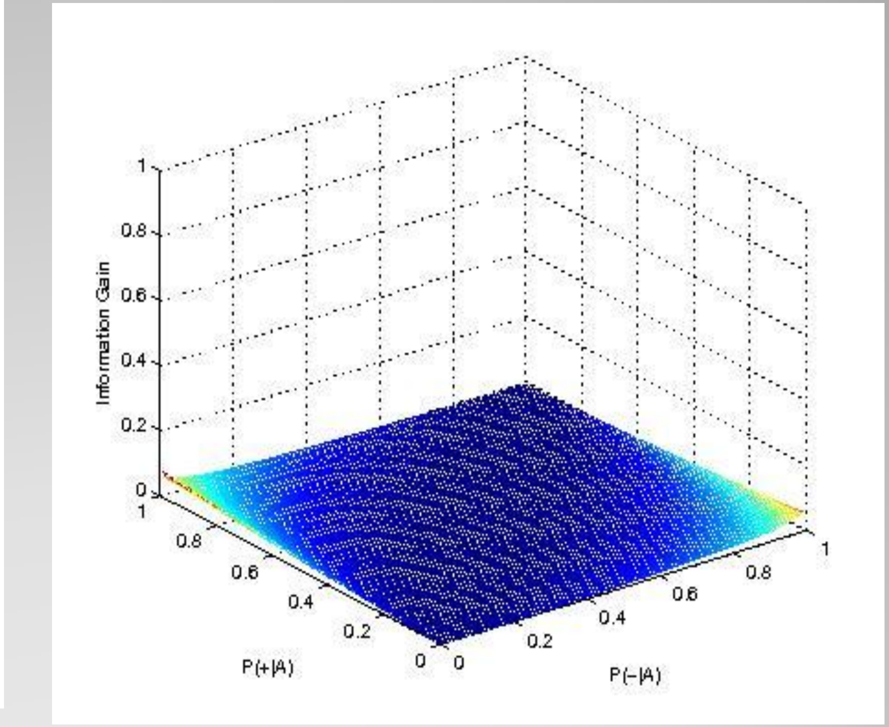
P(x|+) P(x|-)

Information Gain

Class ratio +:- = 1:1

Comparing Value Surfaces





Hellinger Distance

Information Gain

Class ratio +:- = 1:100

HDDT Results

	Base			Sampling		
	C4.5	Gini (CART	HDDT	C4.5	Gini	HDDT
Avg Rank	5.61	7.42	2.50	4.00	6.18	3.79
Friedman 95% conf	√	√	--		√	

- Single Hellinger distance decision trees compete with and surpass sampling classifiers

Conclusions v1.0

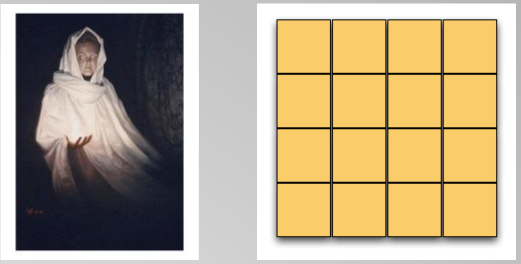
If you are learning on imbalanced data, use

Hellinger Distance Decision Trees.

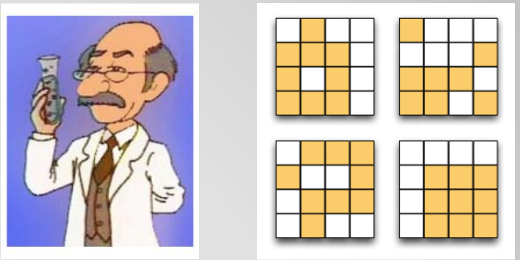
But More Can Be Better

- Traditional: Use 100% of training data to build a sage.
- Ensemble: Randomize training data to build many voted experts (“bagging”).
- Boosting: Emphasize difficult instances in future iterations

One sage sees all the data



Many experts see 2/3’s of the data



Experts outperform the sage!

Imbalanced Data

Determined AUROC for each method on 38 unbalanced datasets.

Key

T: Single Tree

B: Bagging

BT: Boosting

SE: SMOTE 50/50

SMX: SMOTE Wrapper optimized using X

Hellinger Distance (HD) AUROC Ranks

	B	T	Bt	SE	SWT	SWB	SWBt
Average Rank	5.10	14.95	7.16	15.25	16.86	8.62	8.45
90% Confidence		√		√	√		
95% Confidence		√		√	√		
99% Confidence		√		√	√		

Information Gain (IG) AUROC Ranks

	Bt	T	B	SE	SWT	SWB	SWBt
Average Rank	6.23	16.21	6.86	15.50	16.46	8.71	7.64
90% Confidence		√		√	√		
95% Confidence		√		√	√		
99% Confidence		√		√	√		

Imbalanced Data

Which bagging wins?

	HD+B	IG+B
Dataset Wins	16	4
Rank Sum	163	27
Wilcoxon Winner at 95%	√	

Confirmed hypothesis: "Hellinger distance with bagging statistically significantly performs best on unbalanced datasets."

Conclusions v1.1

If you are learning on imbalanced data, use bagged Hellinger Distance Decision Trees.

Balanced Data

Determined Accuracy for each method on 29 balanced datasets.

	HD+Bt	HD+B	IG+Bt	IG+B
Average Rank	2.16	3.03	2.12	3.03
90% Confidence				
95% Confidence				
99% Confidence				

Confirmed hypothesis: "Hellinger distance with bagging does not perform statistically significantly worse on balanced datasets."

Conclusions

If you are learning on imbalanced data, use bagged Hellinger Distance Decision Trees.

If you are learning on balanced data, you may also use bagged Hellinger Distance Decision Trees.

Cieslak and Chawla, "Learning Hellinger Distance Decision Trees for Imbalanced Data," *European Conference on Machine Learning, 2008*

Cieslak and Chawla, "Learning robust and skew insensitive decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),* UNDER REVIEW.

- Add to it "predictive uncertainties"

But, what about the actual talk title: A framework for evaluating models

So, what can I really say about the performance of my favorite model.

Optimal decisions, while they can maximize performance in static environments, can result in fragility for complex, uncertain, and rapidly changing problems.

Often a disagreement between performance evaluation criterion, (perhaps) the learning objective function, and how the model may be deployed.

Ideally, want models agnostic to performance estimates.

Really, that rarely happens.

- Manage the Tipping Point: Prepare for, React to, Manage the Predictive Uncertainties

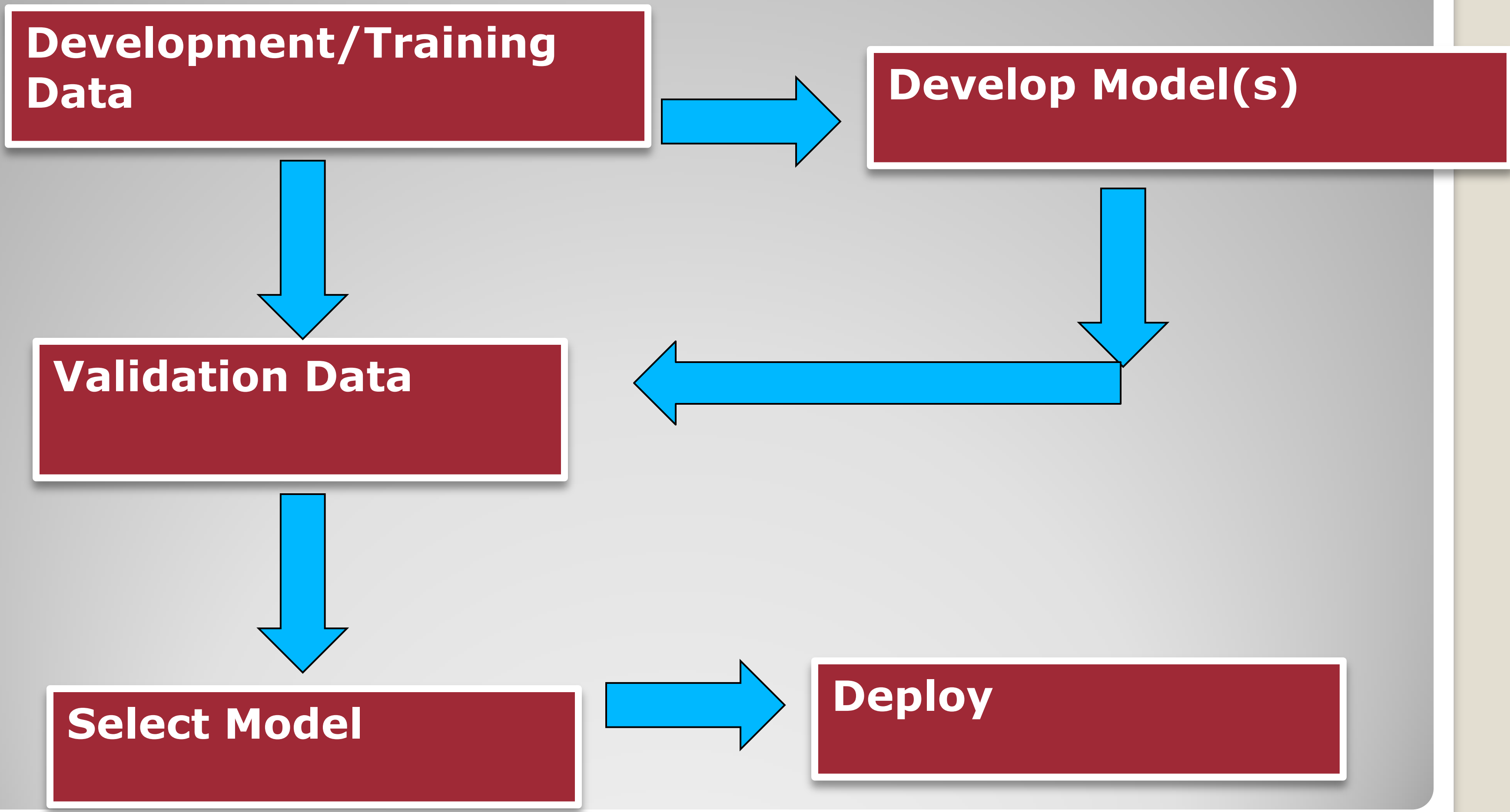
The test sample is supposed to represent the population to be encountered in the future. But in reality, it is usually a random sample of the current population. High performance on the test sample does not guarantee high performance on future samples, things do change.

Grand Challenge Problem

Tipping Point Grand Challenge

- Can we anticipate the impact of potential changes in distribution?
- Can we gauge the impact of those to different performance estimates?
- Can we appropriately weigh and select models for use?

First, let us consider some common steps of model development



Development/Training Data

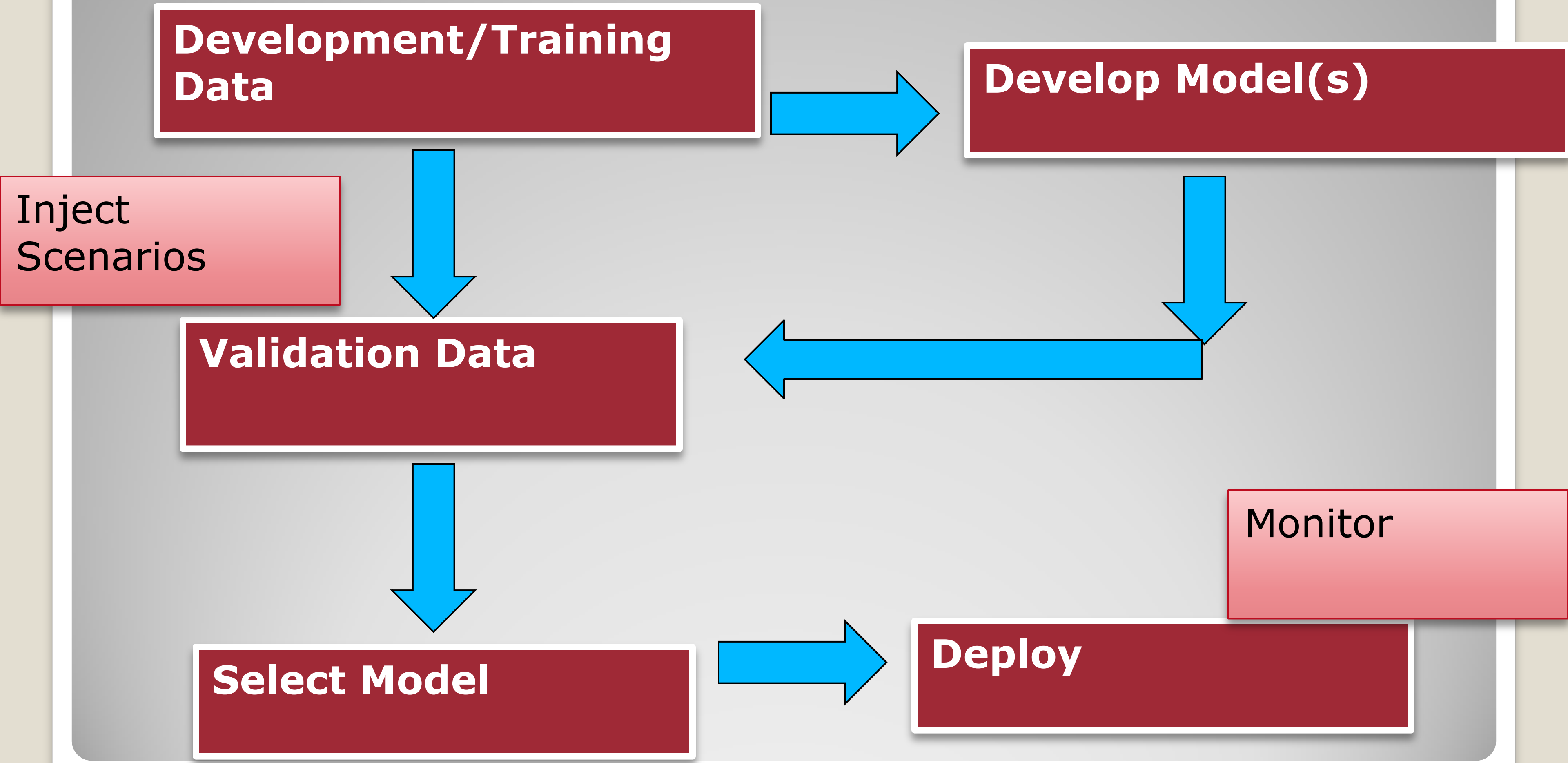
Develop Model(s)

Validation Data

Select Model

Deploy

Let us change this framework.



Development/Training Data

Develop Model(s)

Inject Scenarios

Validation Data

Monitor

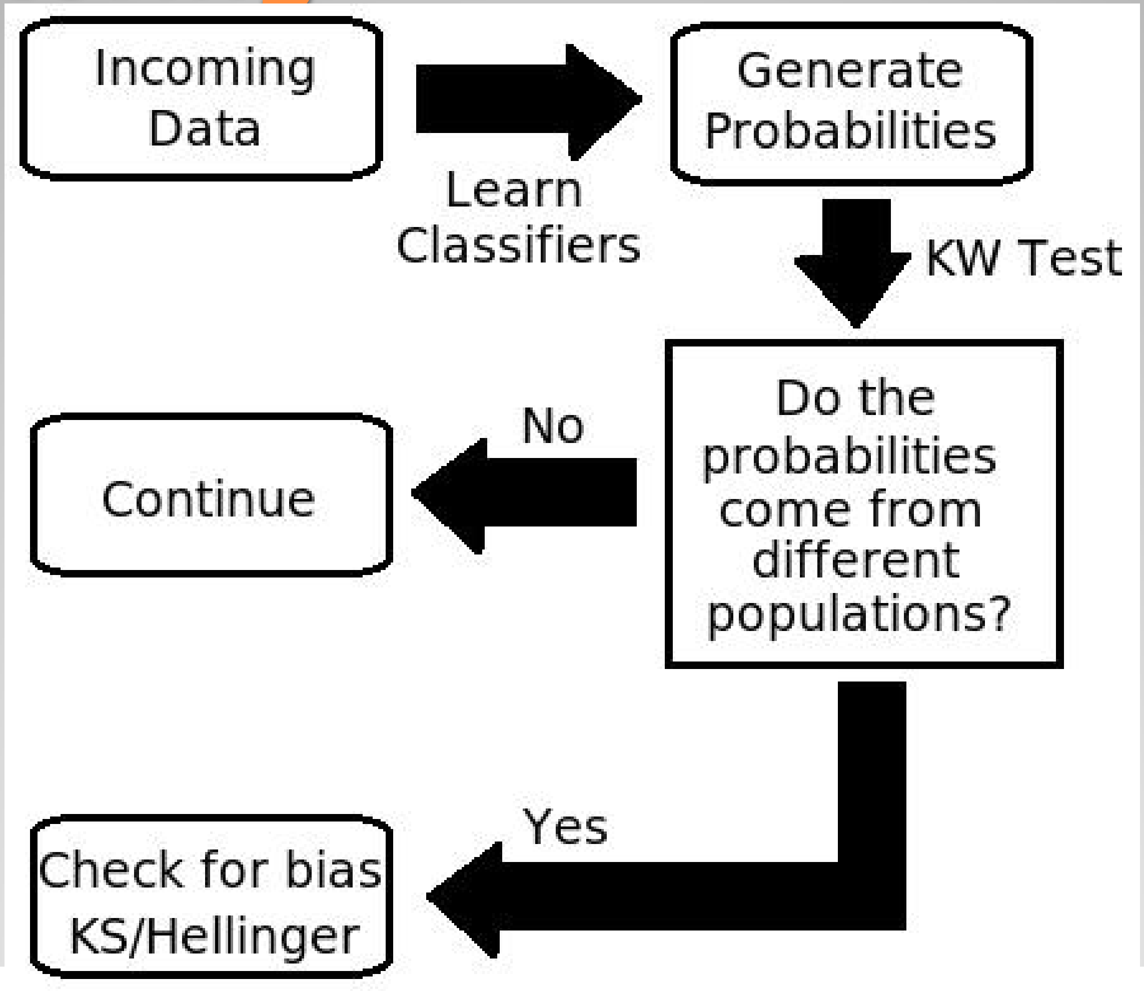
Select Model

Deploy

Possible Scenarios

- Sample Selection Bias
  - Missing Not At Random (MNAR)
- Missing At Random (MAR)
- Shifting Class Priors
- Covariate Shift
- Noise

Monitoring


Incoming Data
Learn Classifiers
Generate Probabilities
KW Test
Do the probabilities come from different populations?
No
Continue
Yes
Check for bias KS/Hellinger

Step 1: Detecting a Fracture

- Learn classifiers and generate probabilities on both validation and testing sets
- Use Kruskal-Wallis on these populations

$$K = (N-1)\frac{\sum_{i=1}^{g} n_i(\bar{r}_i - \bar{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}\left(r_{ij} - \bar{r}\right)^2}$$

Where g is the number of groups, n_i is the size of group *i*, r_{ij} is the rank of observation *j* in group *i*

Step 1: Detecting a Fracture

- Learn classifiers and generate probabilities on both validation and testing sets
- Use Kruskal-Wallis on these populations

$$K = (N-1)\frac{\sum_{i=1}^{g} n_i(\bar{r}_i - \bar{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}\left(r_{ij} - \bar{r}\right)^2}$$

Where g is the number of groups, n_i is the size of group *i*, r_{ij} is the rank of observation *j* in group *i*

Step 2: Isolating Change

If the Kruskal-Wallis results indicate bias, then examine the feature space:

- The **Kolmogorov-Smirnov Test** quantifies the value gap, no distributional assumption
- The **Hellinger Distance** measures the distributional divergence

Detail is in the Design of Experimentation

- Model Monitor Evaluating and Monitoring Models
- You can download from http://www.nd.edu/~dial

Cieslak, Chawla, "Detecting Fractures in Classifier Performance," *IEEE International Conference on Data Mining (ICDM),* 2007

Cieslak, Chawla, "A Framework for Monitoring Classifiers' Performance: When and Why Failure Occurs?," *Knowledge and Information Systems Journal,* 2008

Raeder, Chawla, "Model Monitor: Evaluating, Comparing and Monitoring Models," *Journal of Machine Learning Research,* 2009

Let neither measurement without theory
Nor theory without measurement dominate
Your mind but rather contemplate
A two-way interaction between the two
Which will your thought processes stimulate
To attain syntheses beyond a rational expectation!

Contributed by A. Zellner.

Summary

Workshops, Tutorials on this topic

- Chawla et al., Workshop on Learning from Imbalanced Datasets, *International Conference on Machine Learning*, 2003
- Chawla et al., Special Issue on Learning from Imbalanced Datasets, *SIGKDD Explorations,* 2004
- Chawla et al., Workshop on Mining when rare events matter more, and errors have costs, *PAKDD 2009*
- Chawla, Tutorial: Mining When Classes are Imbalanced, Rare Events Matter More, and Errors Have Costs Attached, *SIAM,* 2009

Thank you

- Questions?
- For papers and software
  - http://www.nd.edu/~dial
  - nchawla@nd.edu