Performance Measures for Machine Learning

Performance Measures

- Accuracy
- Weighted (Cost-Sensitive) Accuracy
- Lift
- Precision/Recall
  - F
  - Break Even Point
- ROC
  - ROC Area

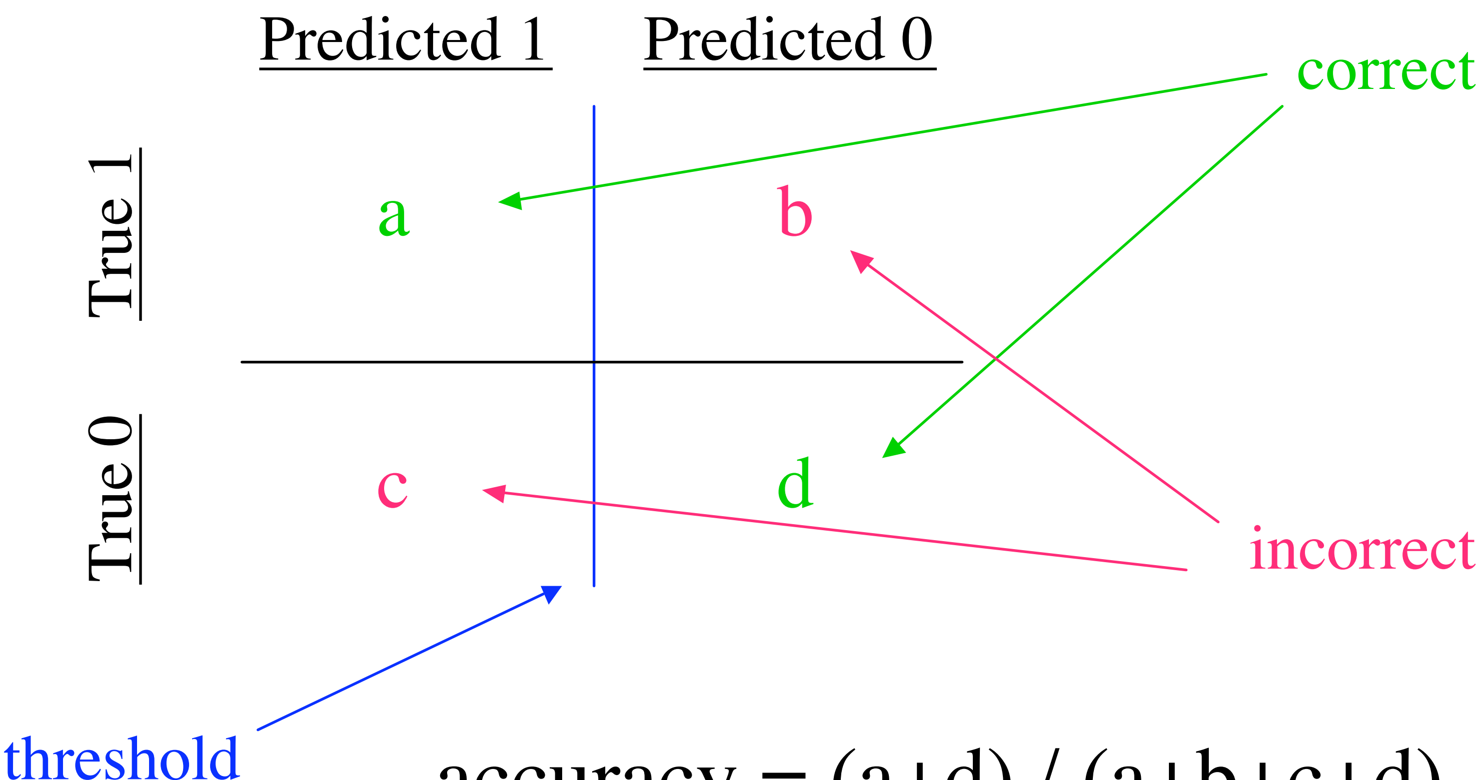
Accuracy

- Target: 0/1, -1/+1, True/False, …
- Prediction = f(inputs) = f(x): 0/1 or Real
- Threshold: f(x) > thresh => 1, else => 0
- threshold(f(x)): 0/1

$$accuracy = \frac{\sum_{i=1\ldots N} \left(1 - (target_i - threshold(f(\vec{x}_i)))\right)^2}{N}$$

- #right / #total
- p("*correct*"): p(threshold(f(x)) = target)

Confusion Matrix



$$\text{accuracy} = (a+d) \,/\, (a+b+c+d)$$

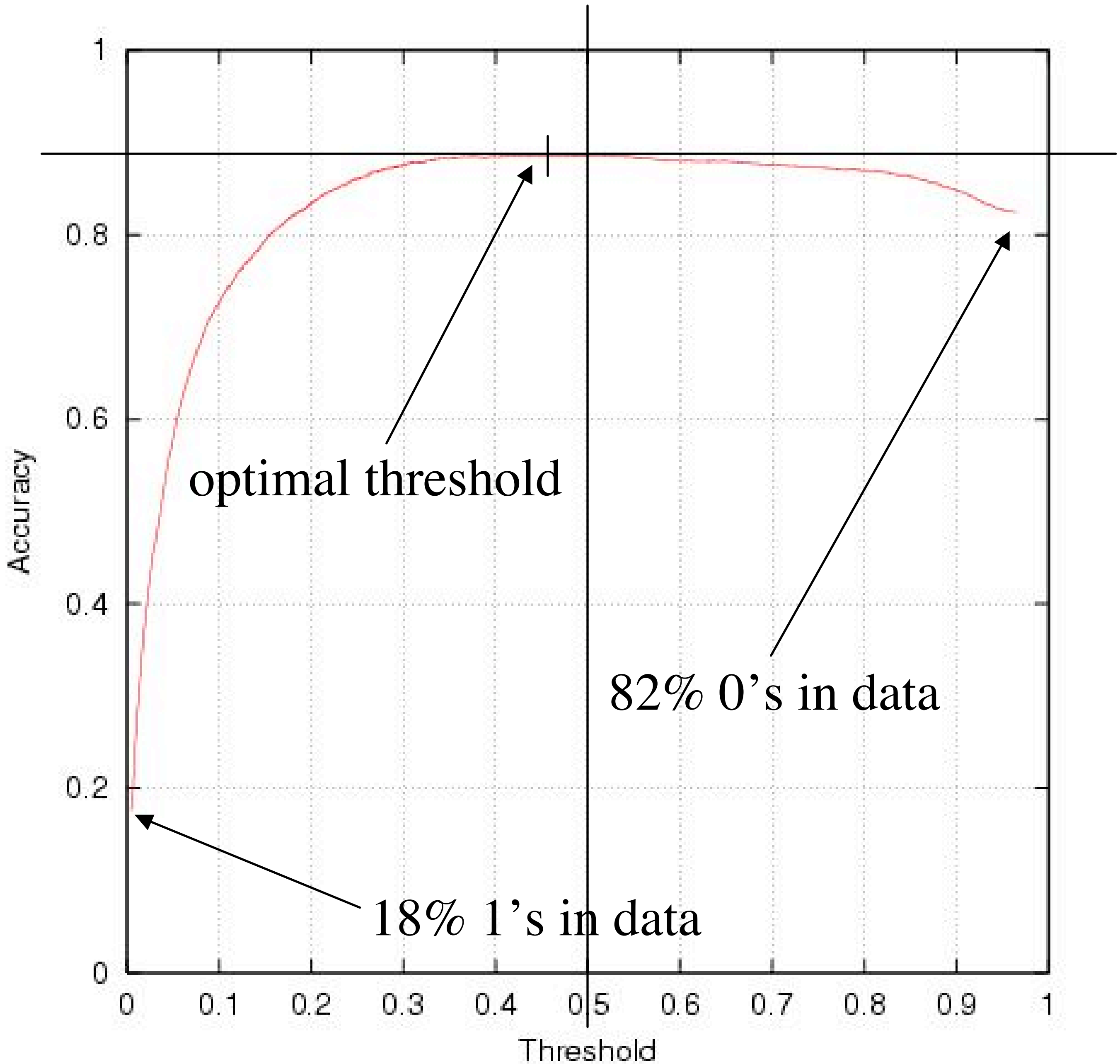
Prediction Threshold

	Predicted 1	Predicted 0
True 1	0	b
True 0	0	d

- threshold > MAX(f(x))
- all cases predicted 0
- (b+d) = total
- accuracy = %False = %0's

	Predicted 1	Predicted 0
True 1	a	0
True 0	c	0

- threshold < MIN(f(x))
- all cases predicted 1
- (a+c) = total
- accuracy = %True = %1's


1
0.8
0.6
0.4
0.2
0
Accuracy
optimal threshold
82% 0's in data
18% 1's in data
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Threshold

threshold demo

Problems with Accuracy

- Assumes equal cost for both kinds of errors
  - cost(b-type-error) = cost (c-type-error)

- is 99% accuracy good?
  - can be excellent, good, mediocre, poor, terrible
  - depends on problem
- is 10% accuracy bad?
  - information retrieval
- BaseRate = accuracy of predicting predominant class
  (on most problems obtaining BaseRate accuracy is easy)

Percent Reduction in Error

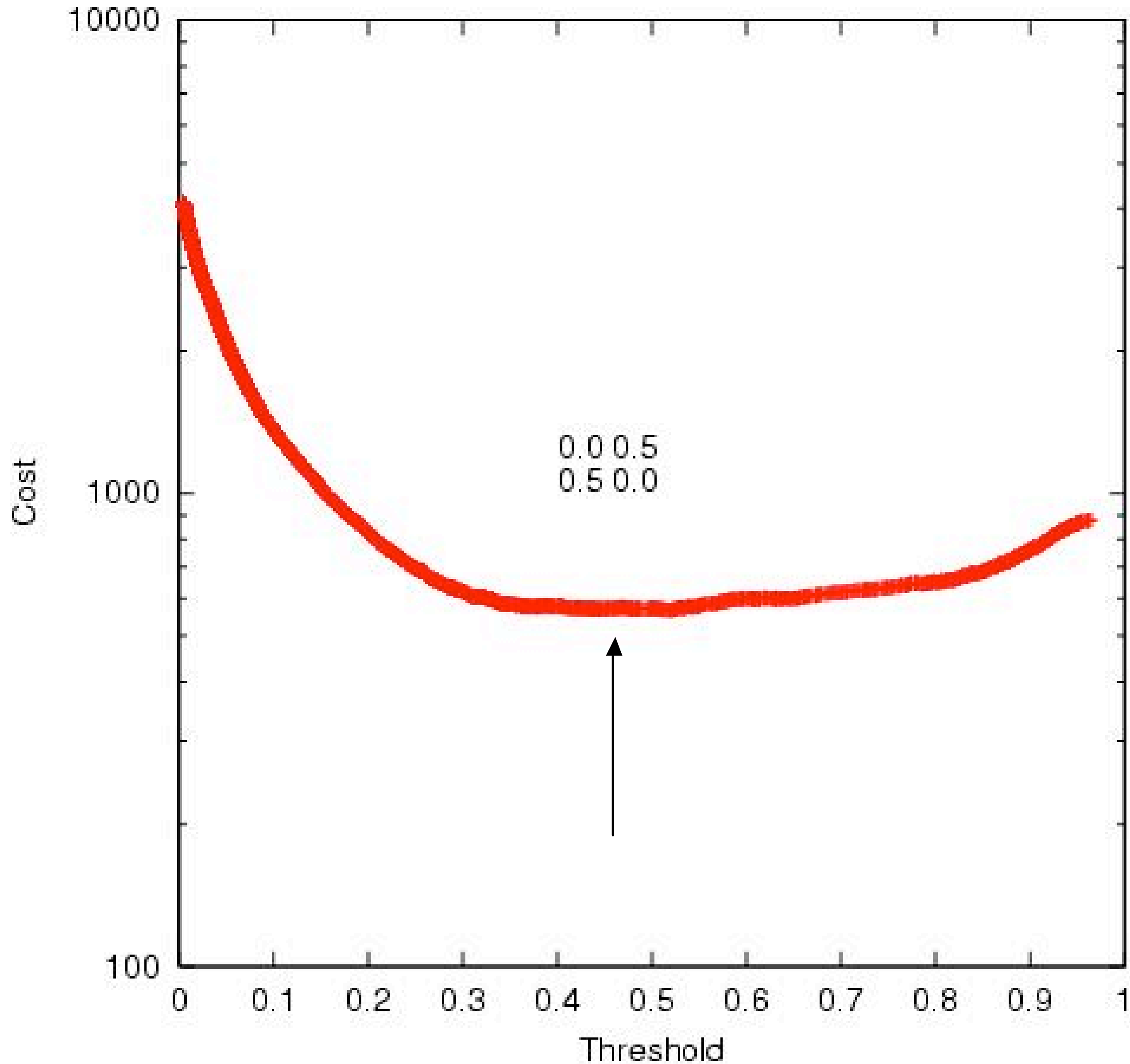
- 80% accuracy = 20% error
- suppose learning increases accuracy from 80% to 90%
- error reduced from 20% to 10%
- 50% reduction in error

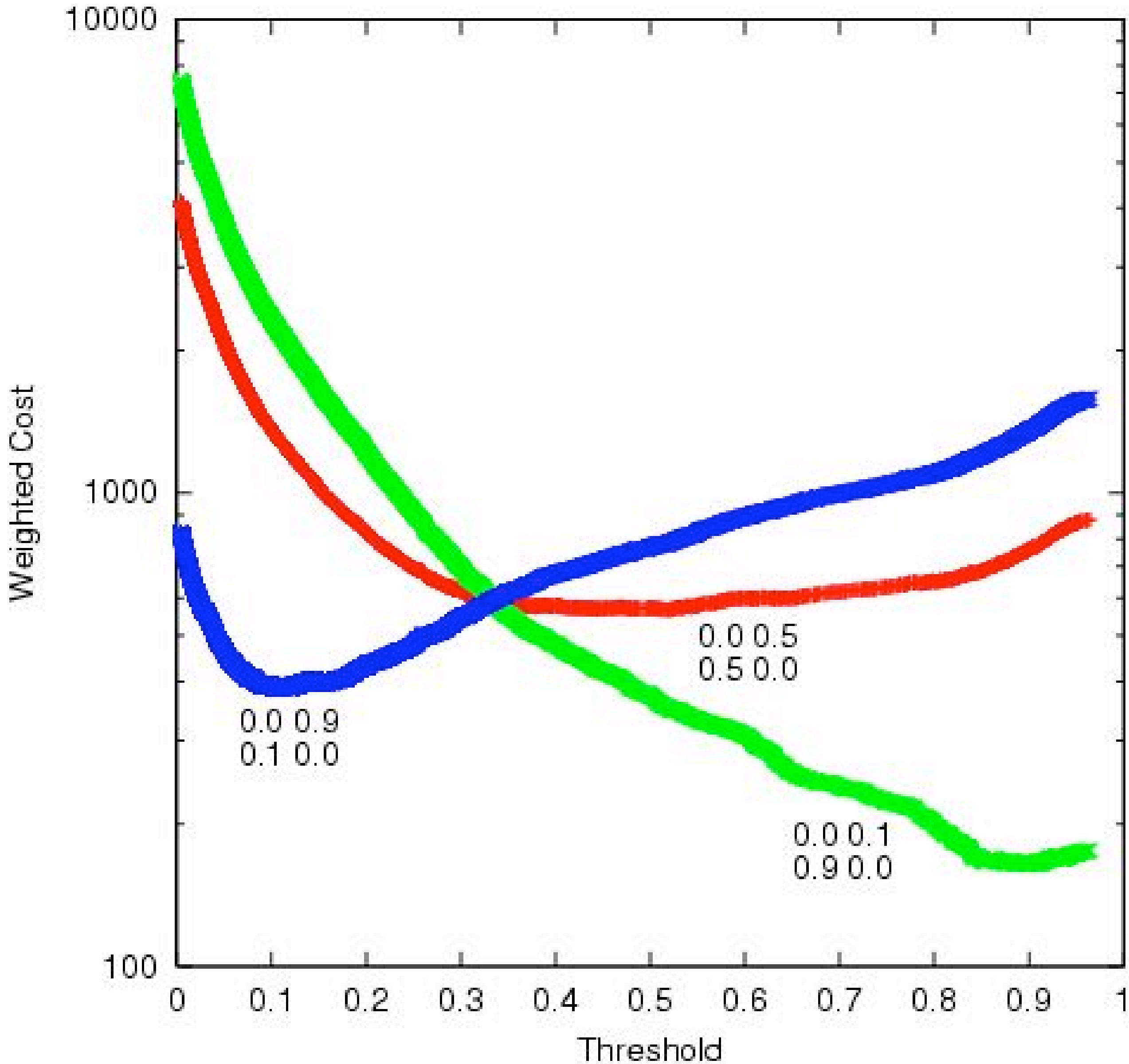
- 99.90% to 99.99% = 90% reduction in error
- 50% to 75% = 50% reduction in error
- can be applied to many other measures

Costs (Error Weights)

	Predicted 1	Predicted 0
True 1	w_a	w_b
True 0	w_c	w_d

- Often $W_a = W_d =$ **zero** and $W_b \neq W_c \neq$ **zero**


10000
1000
100
Cost
0.0 0.5
0.5 0.0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Threshold


10000
1000
100
Weighted Cost
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Threshold
0.0 0.5
0.5 0.0
0.0 0.9
0.1 0.0
0.0 0.1
0.9 0.0

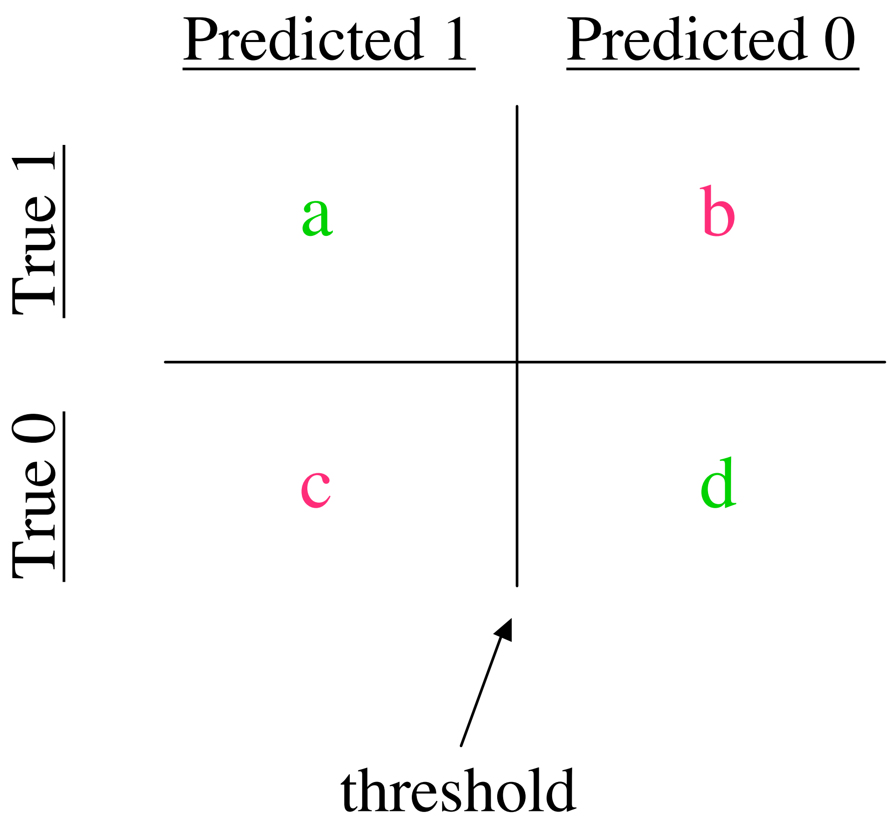
Lift

- not interested in accuracy on entire dataset
- want accurate predictions for 5%, 10%, or 20% of dataset
- don't care about remaining 95%, 90%, 80%, resp.
- typical application: marketing

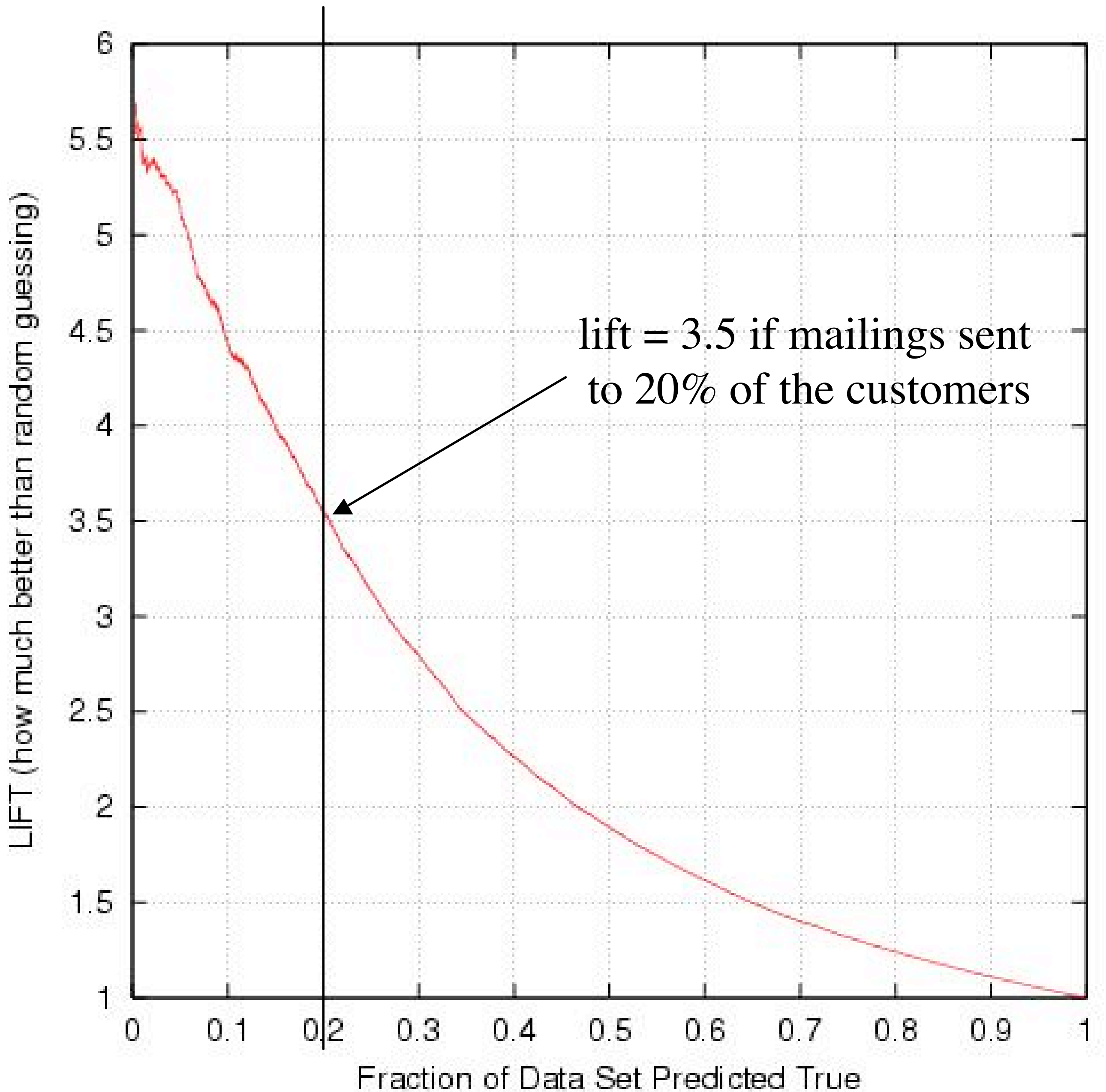
$$lift(threshold) = \frac{\%\,positives > threshold}{\%\,dataset > threshold}$$

- how much better than random prediction on the fraction of the dataset predicted true (f(x) > threshold)

Lift

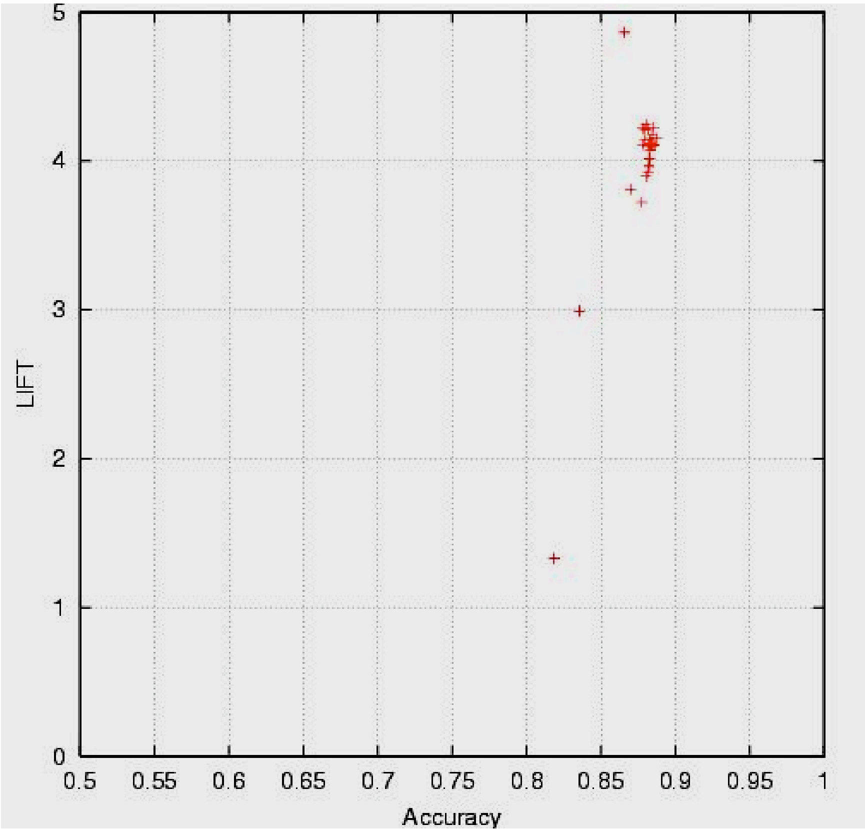


$$lift = \frac{a/(a+b)}{(a+c)/(a+b+c+d)}$$

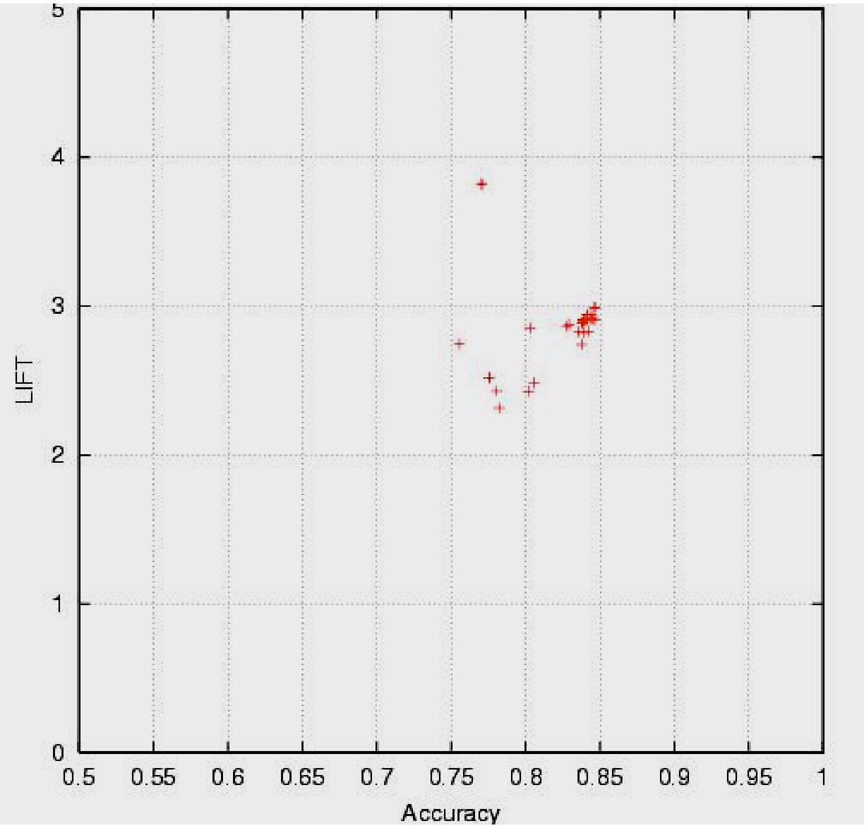

6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
LIFT (how much better than random guessing)
Fraction of Data Set Predicted True
lift = 3.5 if mailings sent
to 20% of the customers

Lift and Accuracy do not always correlate well

Problem 1



Problem 2



(thresholds arbitrarily set at 0.5 for both lift and accuracy)

Precision and Recall

- typically used in document retrieval
- Precision:
  - how many of the returned documents are correct
  - precision(threshold)
- Recall:
  - how many of the positives does the model return
  - recall(threshold)
- Precision/Recall Curve: sweep thresholds

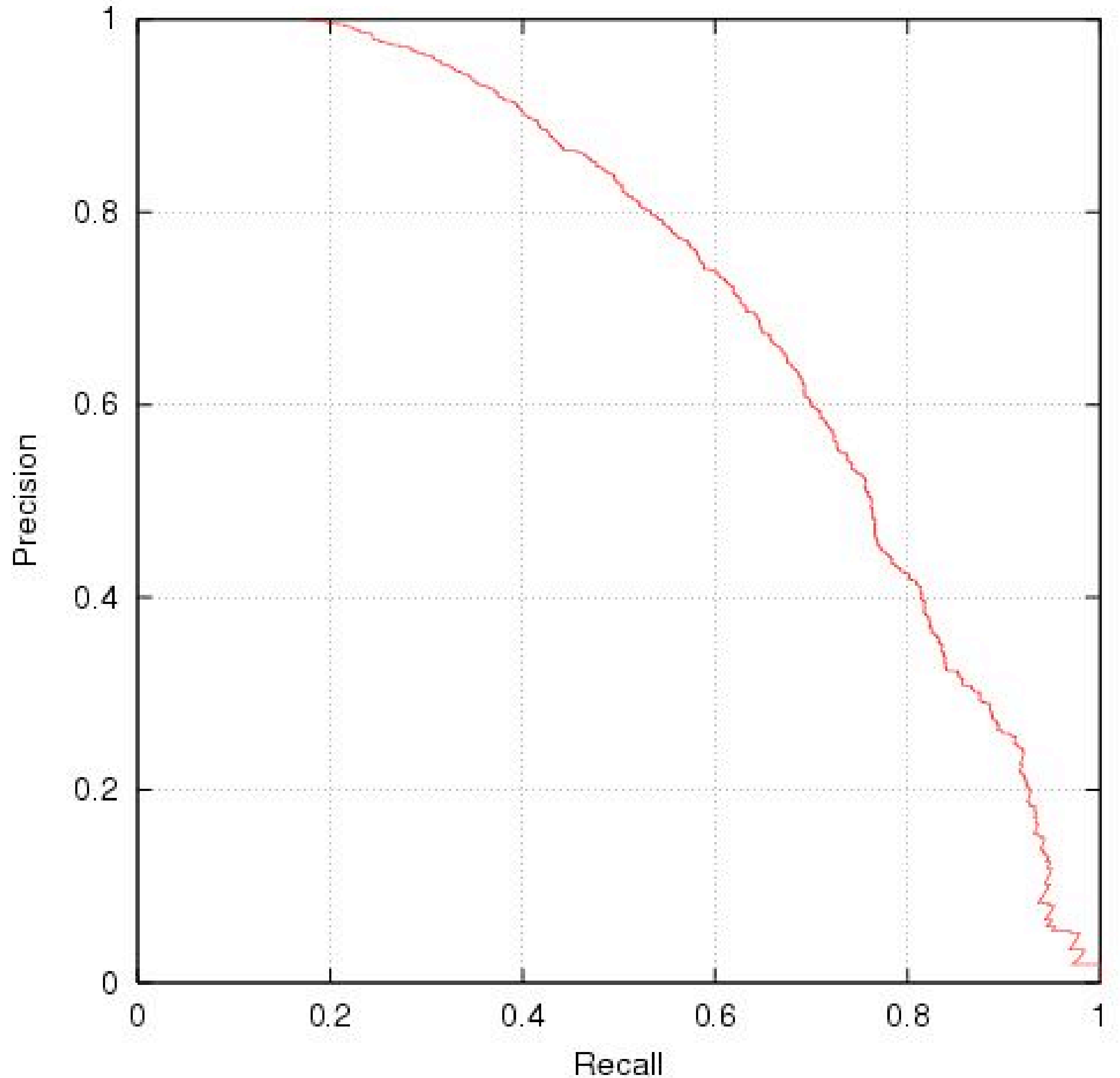
Precision/Recall

	Predicted 1	Predicted 0
True 1	a	b
True 0	c	d

threshold

$$PRECISION = a/(a+c)$$

$$RECALL = a/(a+b)$$


1
0.8
0.6
0.4
0.2
0
Precision
0
0.2
0.4
0.6
0.8
1
Recall

Summary Stats: F & BreakEvenPt

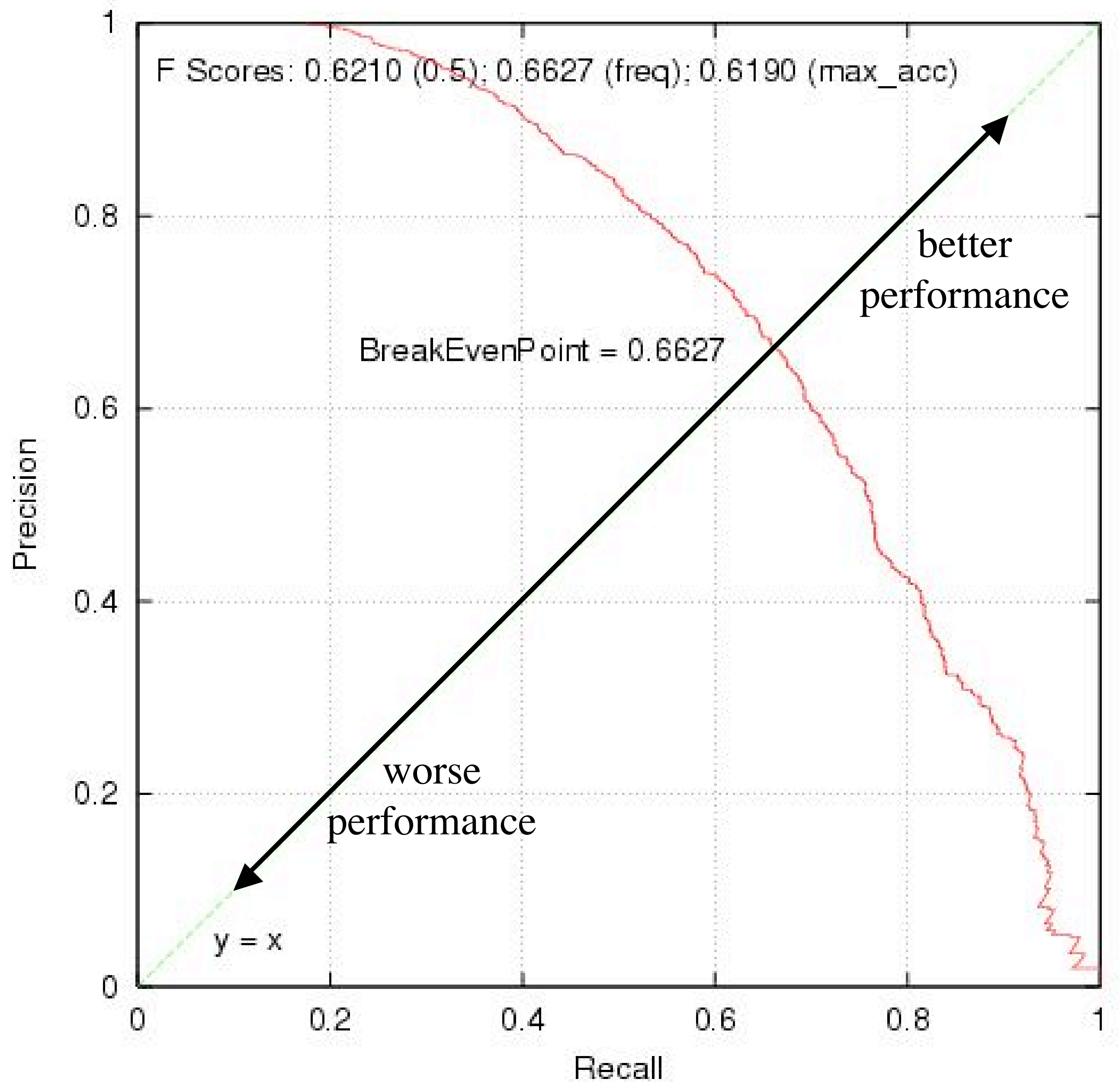
$$PRECISION = a/(a+c)$$

$$RECALL = a/(a+b)$$

$$F = \frac{2*(PRECISION \times RECALL)}{(PRECISION + RECALL)}$$

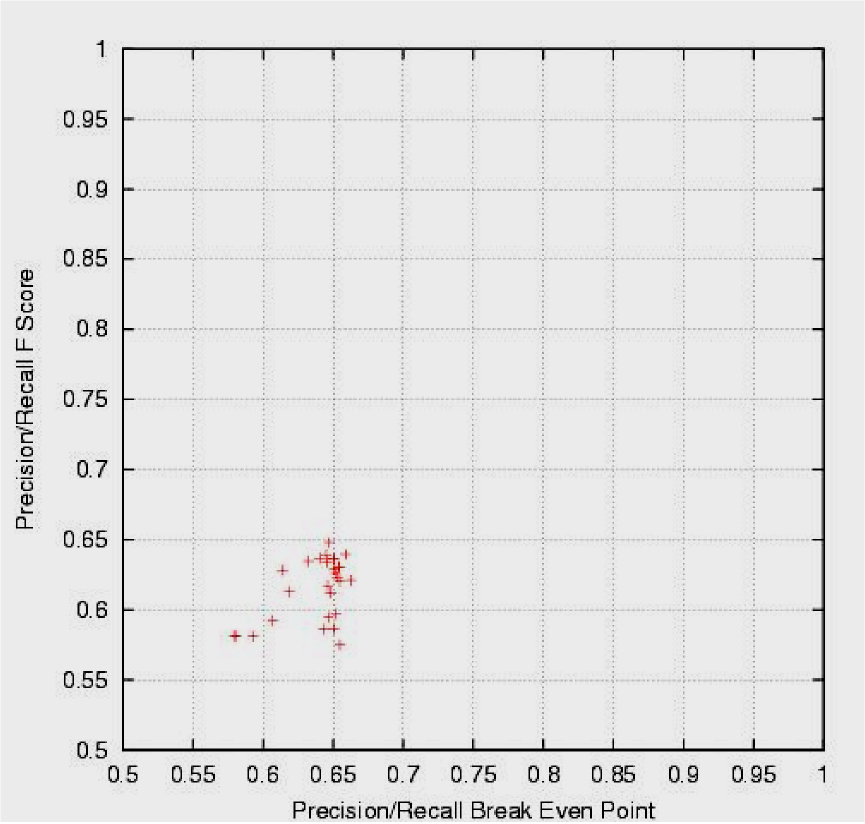
harmonic average of precision and recall

$$BreakEvenPoint = PRECISION = RECALL$$

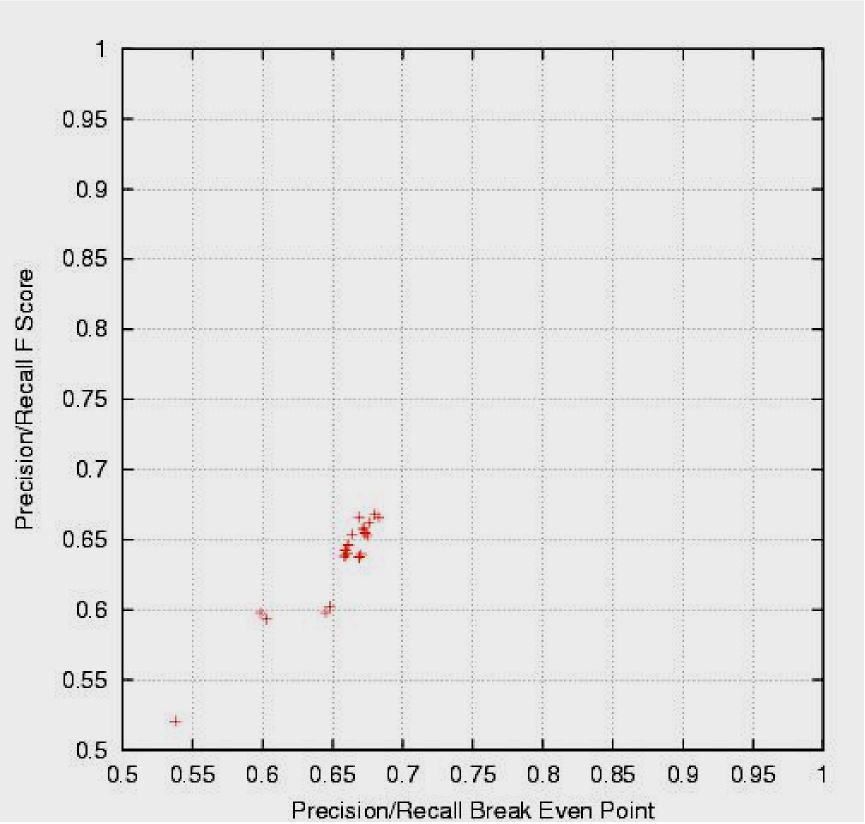

F Scores: 0.6210 (0.5); 0.6627 (freq); 0.6190 (max_acc)
better
performance
BreakEvenPoint = 0.6627
worse
performance
y = x
Precision
Recall
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

F and BreakEvenPoint do not always correlate well

Problem 1



Problem 2



	Predicted 1	Predicted 0
True 1	true positive	false negative
True 0	false positive	true negative

	Predicted 1	Predicted 0
True 1	TP	FN
True 0	FP	TN

	Predicted 1	Predicted 0
True 1	hits	misses
True 0	false alarms	correct rejections

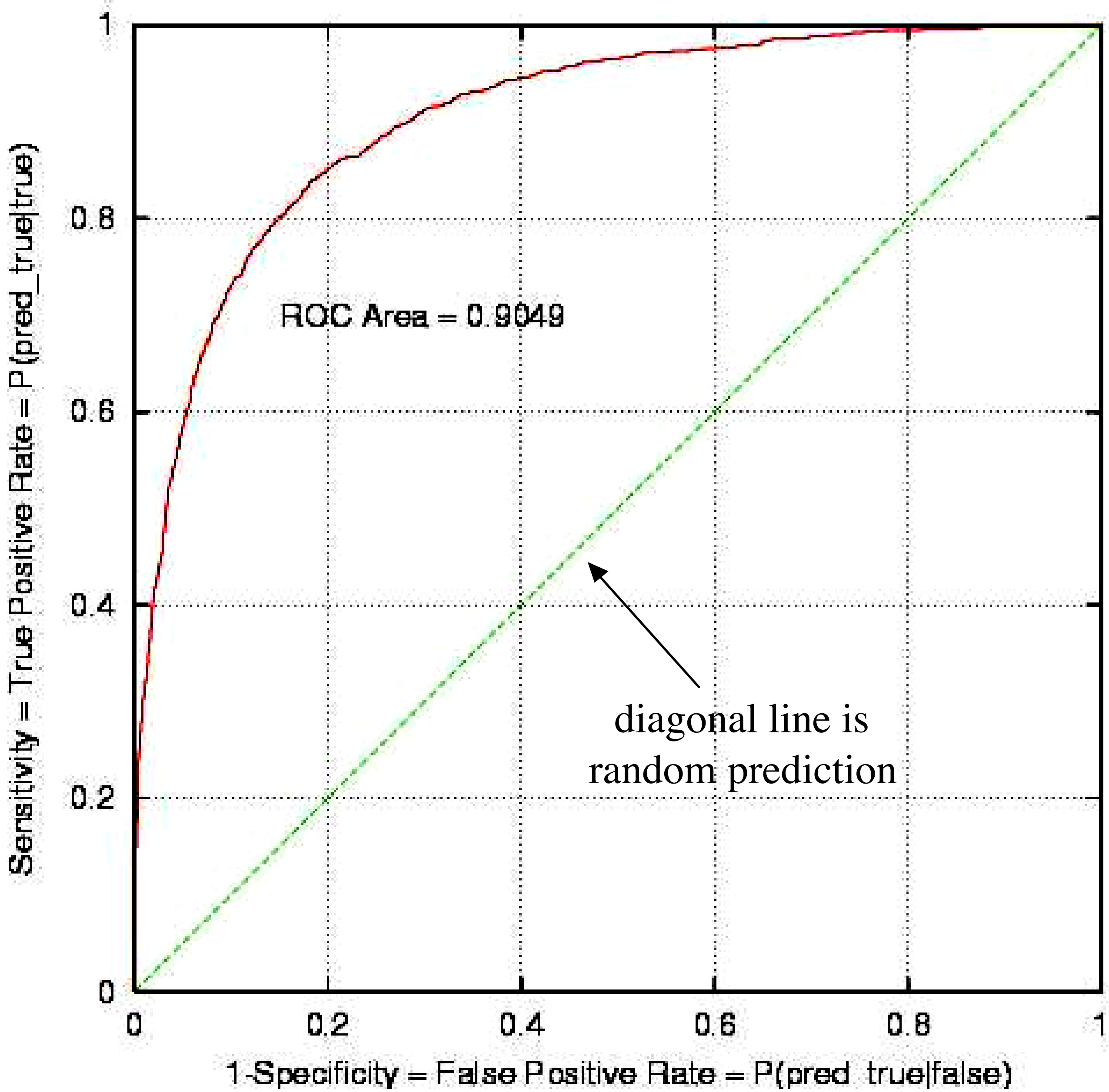
	Predicted 1	Predicted 0
True 1	P(pr1|tr1)	P(pr0|tr1)
True 0	P(pr1|tr0)	P(pr0|tr0)

ROC Plot and ROC Area

- Receiver Operator Characteristic
- Developed in WWII to statistically model false positive and false negative detections of radar operators
- Better statistical foundations than most other measures
- Standard measure in medicine and biology
- Becoming more popular in ML

ROC Plot

- Sweep threshold and plot
  - TPR vs. FPR
  - Sensitivity vs. 1-Specificity
  - P(true|true) vs. P(true|false)
- Sensitivity = a/(a+b) = Recall = LIFT numerator
- 1 - Specificity = 1 - d/(c+d)


ROC Area = 0.9049
diagonal line is random prediction
Sensitivity = True Positive Rate = P(pred_true|true)
1-Specificity = False Positive Rate = P(pred true|false)
0
0.2
0.4
0.6
0.8
1

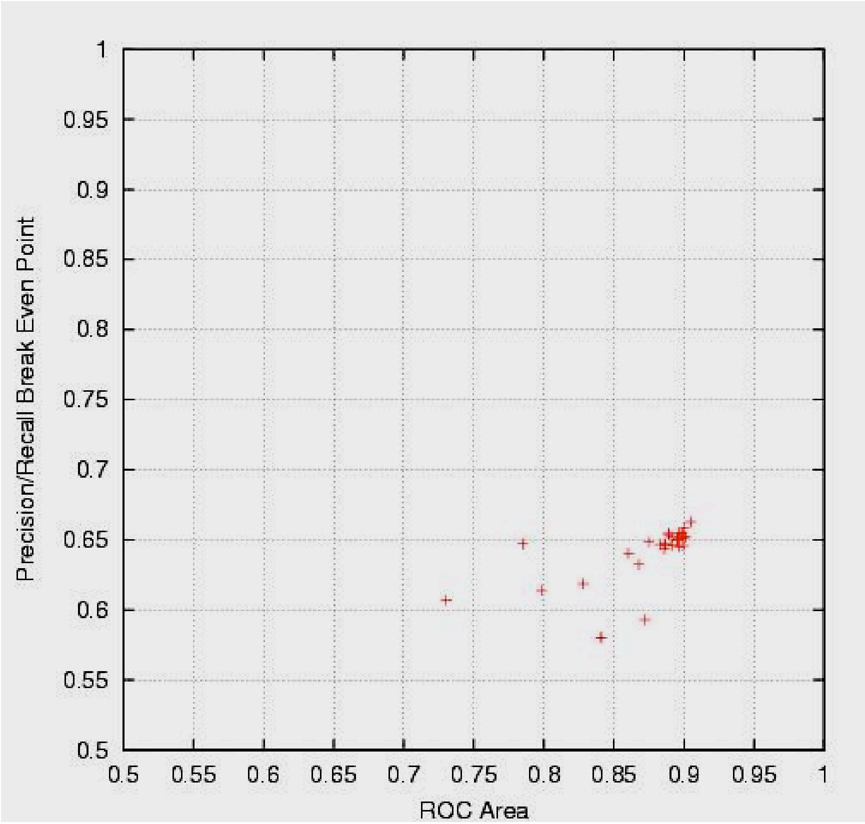
Properties of ROC

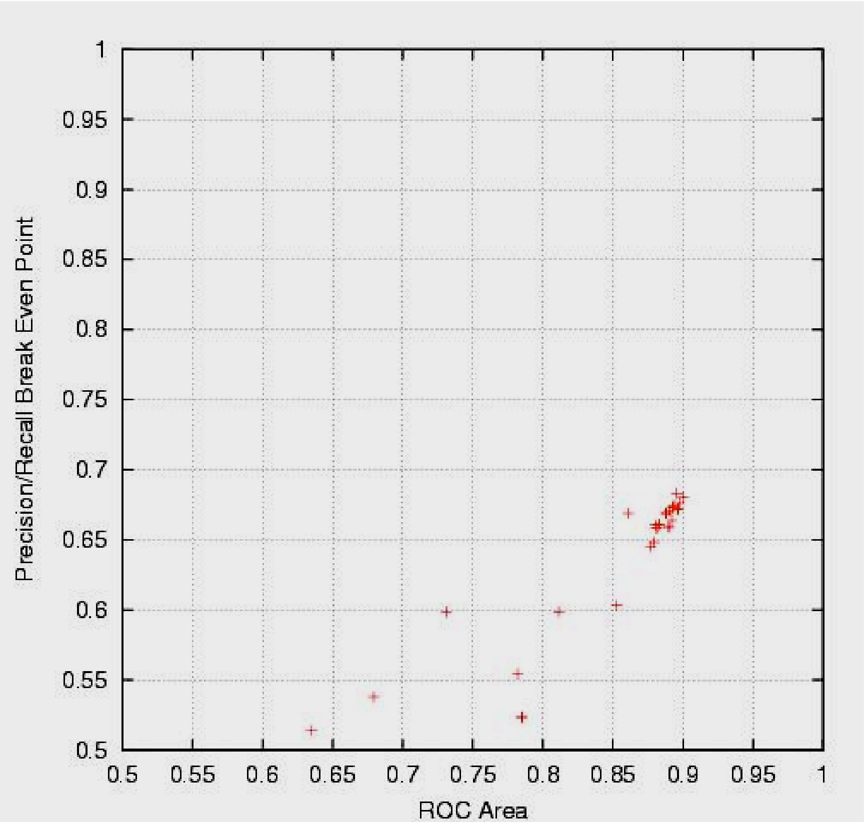
- ROC Area:
  - 1.0: perfect prediction
  - 0.9: excellent prediction
  - 0.8: good prediction
  - 0.7: mediocre prediction
  - 0.6: poor prediction
  - 0.5: random prediction
  - <0.5: something wrong!

Properties of ROC

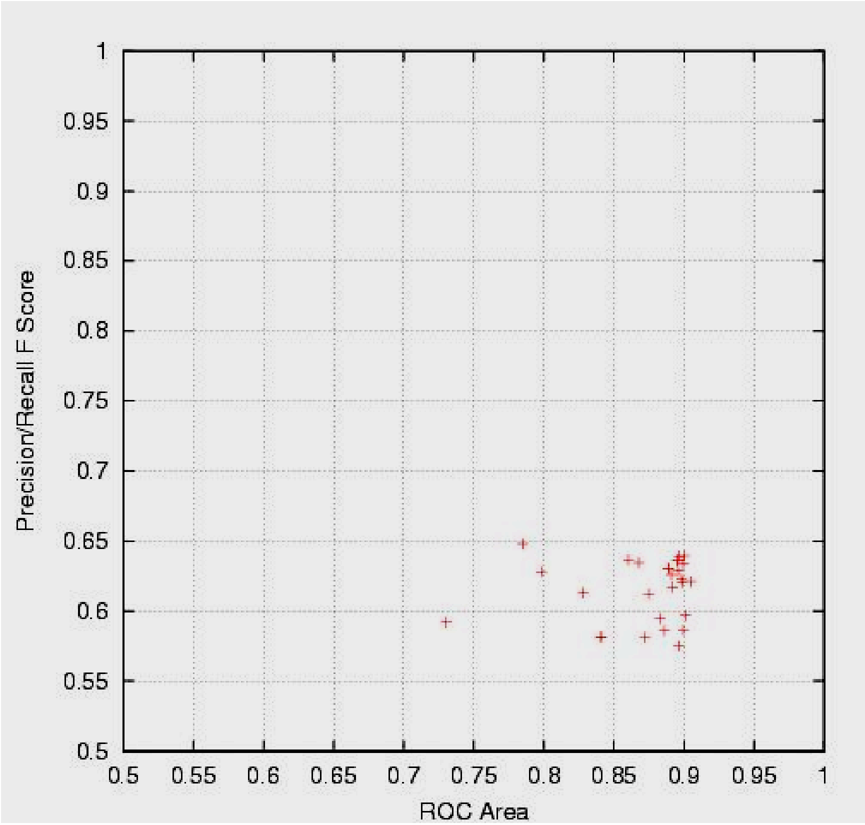
- Slope is non-increasing
- Each point on ROC represents different tradeoff (cost ratio) between false positives and false negatives
- Slope of line tangent to curve defines the cost ratio
- ROC Area represents performance averaged over all possible cost ratios
- If two ROC curves do not intersect, one method dominates the other
- If two ROC curves intersect, one method is better for some cost ratios, and other method is better for other cost ratios

Problem 1	Problem 2

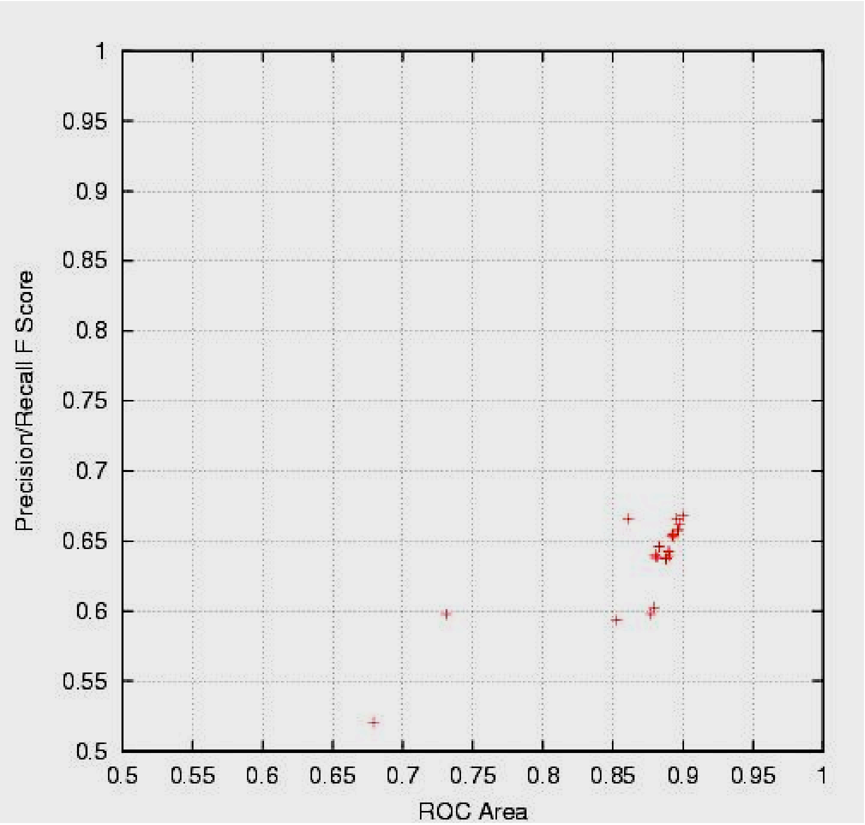




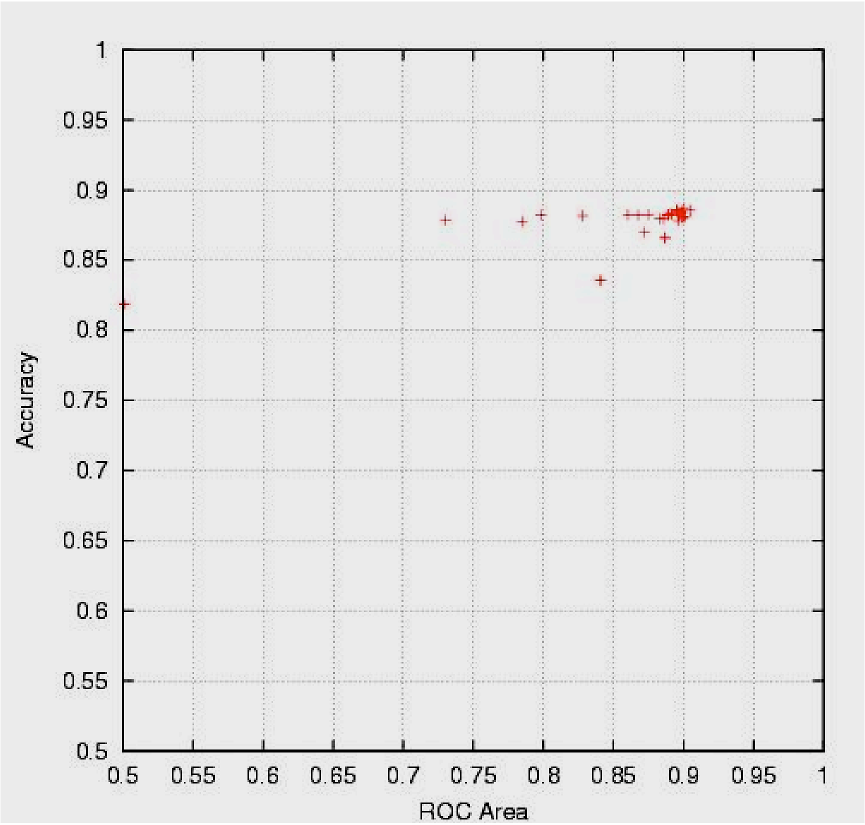
Problem 1



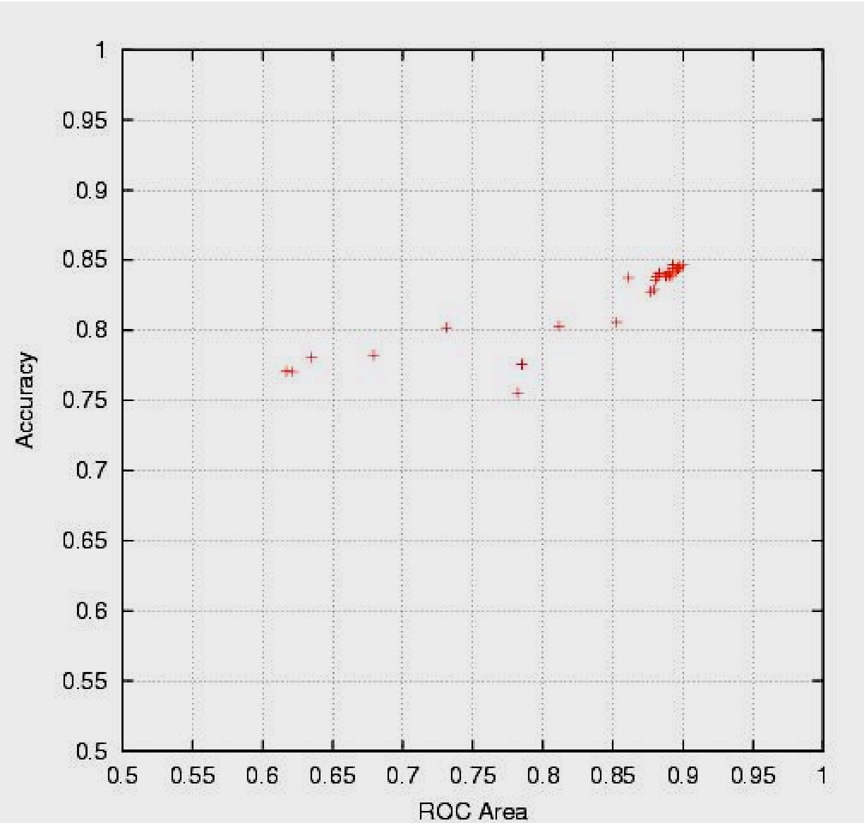
Problem 2



Problem 1



Problem 2



Summary

- the measure you optimize to makes a difference
- the measure you report makes a difference
- use measure appropriate for problem/community
- accuracy often is not sufficient/appropriate
- ROC is gaining popularity in the ML community
- only accuracy generalizes to >2 classes!