Learning from Imbalanced Classes: Problem Statement & Methods

Nikos Nikolaou

EPSRC Doctoral Prize Fellow,
Machine Learning & Optimization Group
School of Computer Science, University of Manchester

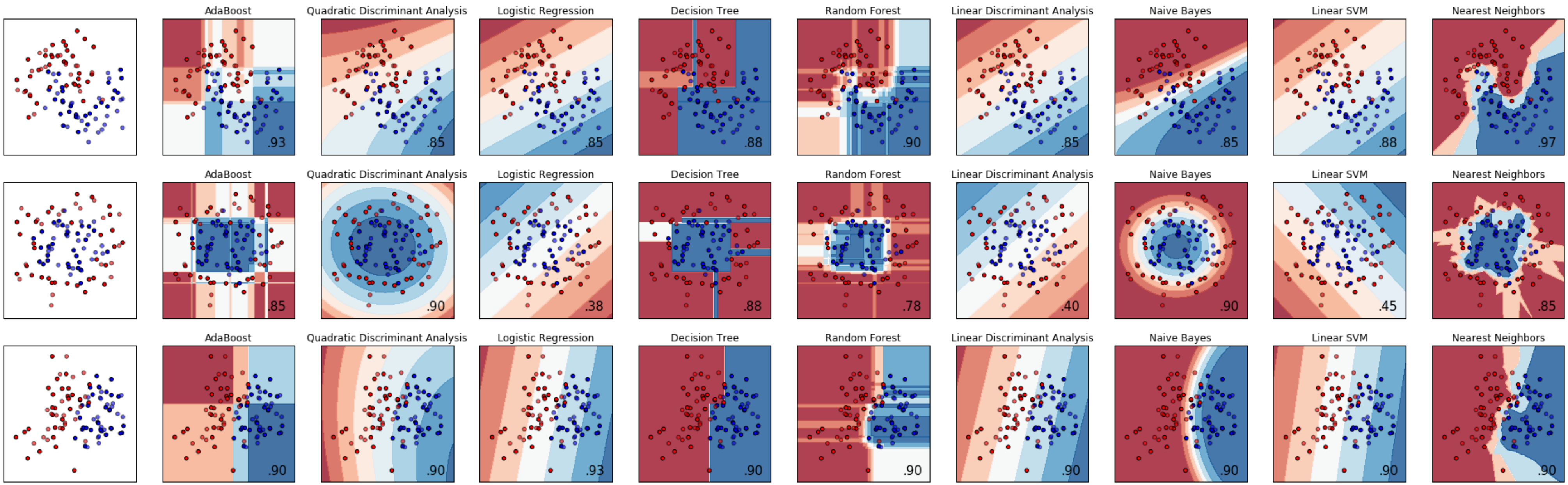
nikolaos.nikolaou@manchester.ac.uk

Engineering and Physical Sciences Research Council

The University of Manchester

Classification

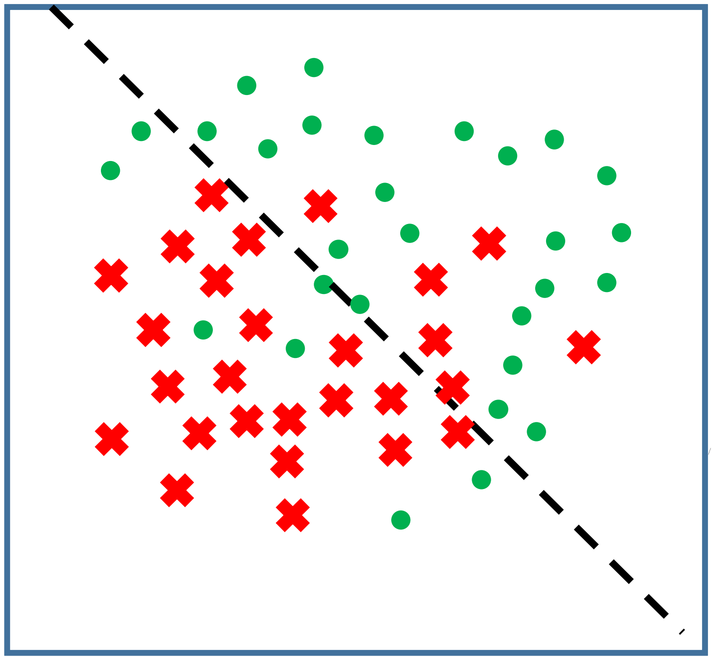
Given a set of points in some space belonging to different classes…



…learn a **decision surface** that **'best' separates classes**

Many **learning algorithms** each with its own **assumptions** (statistical, probabilistic, mathematical, geometrical, …)

Balanced vs. imbalanced class data



Imbalance often significant

Rare class often much **more important**

Standard algorithms & **evaluation measures treat both classes equally**

Imbalanced class learning: set of techniques for amending this

Outcomes of (binary) classification

Confusion matrix (contingency table)

	Truth	
Prediction	Positive	Negative
Positive	True Positive (TP)	False Positive (FP) Type I Error
Negative	False Negative (FN) Type II Error	True Negative (TN)

Can extend to multiclass classification...

Convention: **Rare class = Positive**

Can use **entries** to calculate various **evaluation measures**

↓

KNOW WHAT YOU WANT YOUR CLASSIFIER TO DO!!!

I. Defining the problem

- Ensure as many of Pos predictions are indeed Pos
- Ensure as many of Pos examples are predicted as Pos
- Achieve a (weighted) balance of the above
- Achieve good performance across classes
- Minimize expected cost (risk) of classifications
- Maximize TPR for a given maximum FPR
- And more…

Popular evaluation measures

	Truth	
Prediction	Positive	Negative
Positive	TP	FP
Negative	FN	TN

$$G - mean = \sqrt{Recall * Specificity}$$

Geometric mean of Recall & Specificity

$$F_\beta - measure = \frac{(1+\beta^2) * Precision * Recall}{\beta^2 Precision + Recall}$$

Weighted harmonic mean of Precision & Recall
(Common special case: $\beta = 1$, equal weight)

$$Precision = \frac{TP}{TP + FP}$$ $(Positive\ Predictive\ Value)$

% of Pos predictions that are indeed Pos

$$Recall = \frac{TP}{TP + FN}$$ $(Sensitivity, True\ Positive\ Rate)$

% of Pos that are indeed predicted as Pos

$$Specificity = \frac{TN}{TN + FP}$$ $(True\ Negative\ Rate)$

% of Neg that are indeed predicted as Neg

Other evaluation measures...

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

specificity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}} = 1-\mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = 1-\mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TP}} = 1-\mathrm{PPV}$$

false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TN}} = 1-\mathrm{NPV}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{P+N} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

F1 score

is the harmonic mean of precision and sensitivity

$$F_1 = 2\cdot\frac{\mathrm{PPV}\cdot\mathrm{TPR}}{\mathrm{PPV}+\mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$$

Matthews correlation coefficient (MCC)

$$\mathrm{MCC} = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

Informedness or Bookmaker Informedness (BM)

$$\mathrm{BM} = \mathrm{TPR}+\mathrm{TNR}-1$$

Markedness (MK)

$$\mathrm{MK} = \mathrm{PPV}+\mathrm{NPV}-1$$

And many many more...

Positive Likelihood Ratio

$$\mathrm{LR+} = \frac{\mathrm{TPR}}{\mathrm{FPR}}$$

Dominance

Dominance = $TPR - TNR$

Negative Likelihood Ratio

$$\mathrm{LR-} = \frac{\mathrm{FNR}}{\mathrm{TNR}}$$

Index of Balanced Accuracy

$IBA_a = (1 + a \times dominance)$ACC

(can also define for other metrics than ACC)

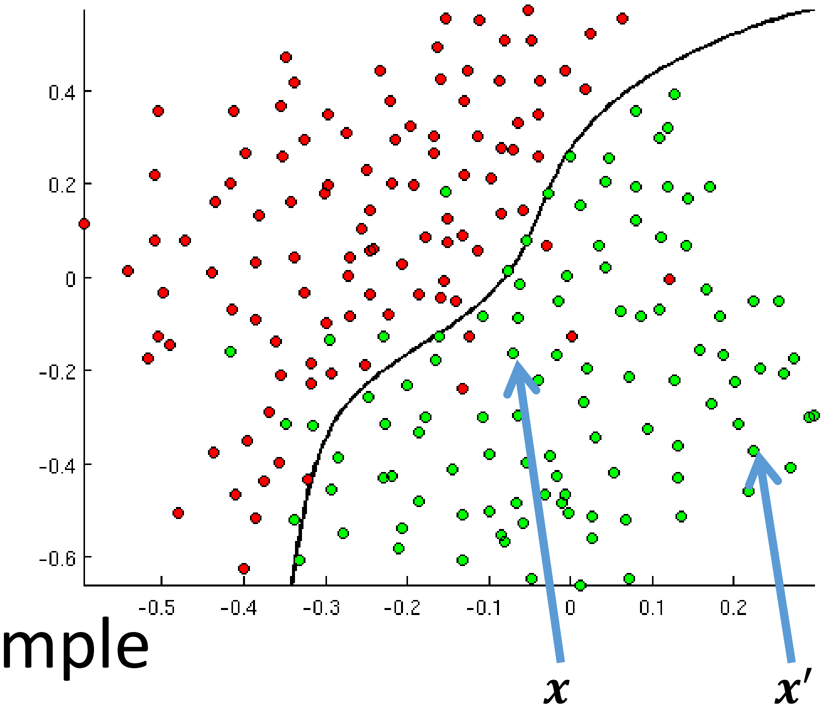
Diagnostic Odds Ratio

$$DOR = \frac{\mathrm{LR+}}{\mathrm{LR-}}$$

Precision at n

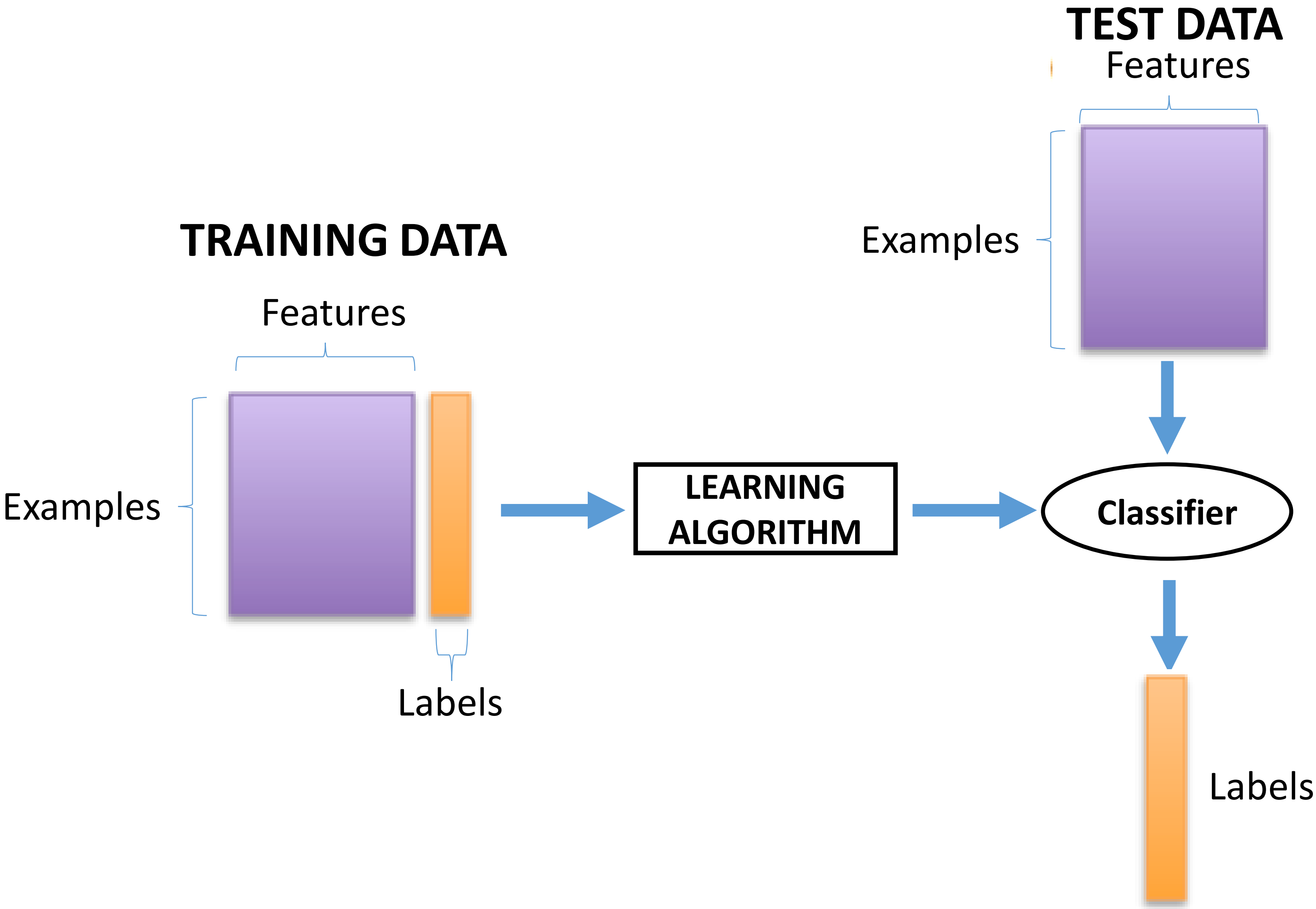
(as precision but for n top-ranked datapoints)

'Classifiers' can do many things...



- **Classify** examples
  - Is x positive?
- **Rank** examples
  - Is x 'more positive' than x'?
- Output a **score** for each example
  - 'How positive' is x?
- Output a **probability estimate** for each example
  - What is the (estimated) probability that x is positive?

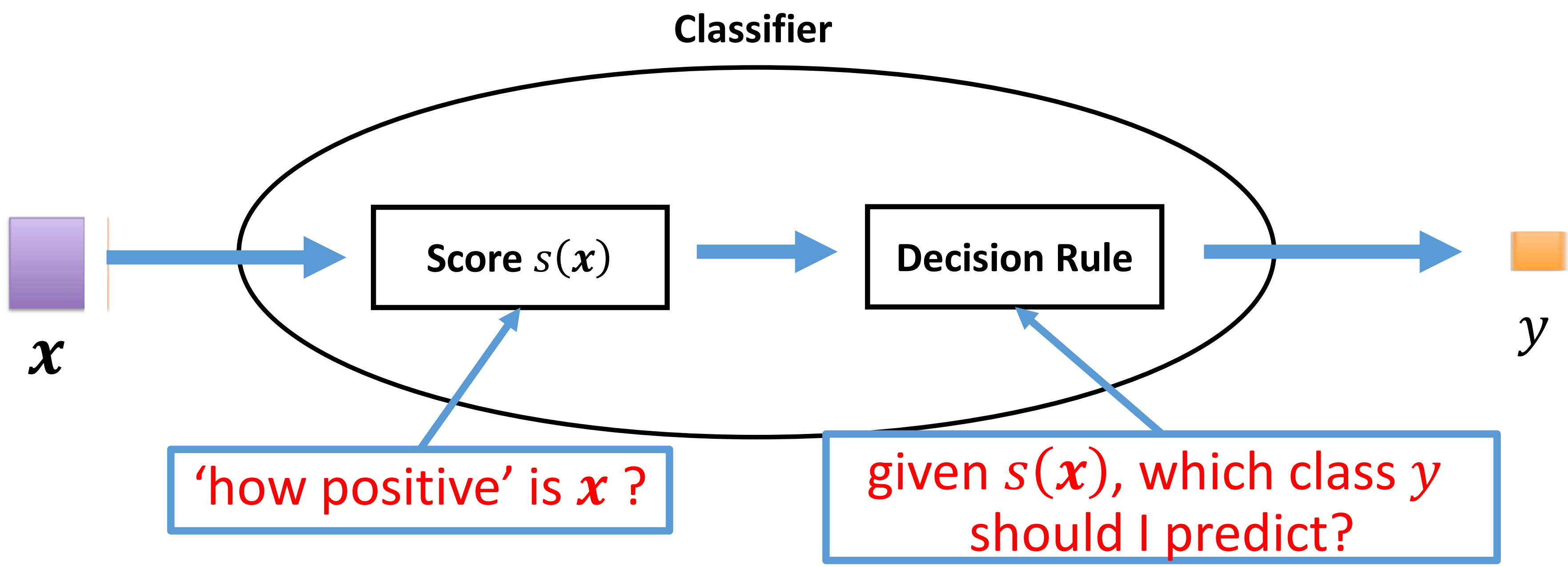
From examples to predictions



Peeking into the classifier

Scoring classifiers: quantify **'how positive'** they deem examples

...then use this number to **decide which class** to assign them



Normalized scores $s(\boldsymbol{x}) \epsilon [0,1]$ often treated as **'probability estimates'**

BUT BEWARE: most models produce biased scores!

A single model, many classifiers

A **decision rule** looks like:

$$\boxed{\begin{array}{c} IF\ s(\boldsymbol{x}) > t\ THEN\ predict\ y = Pos \\ IF\ s(\boldsymbol{x}) < t\ THEN\ predict\ y = Neg \end{array}}$$

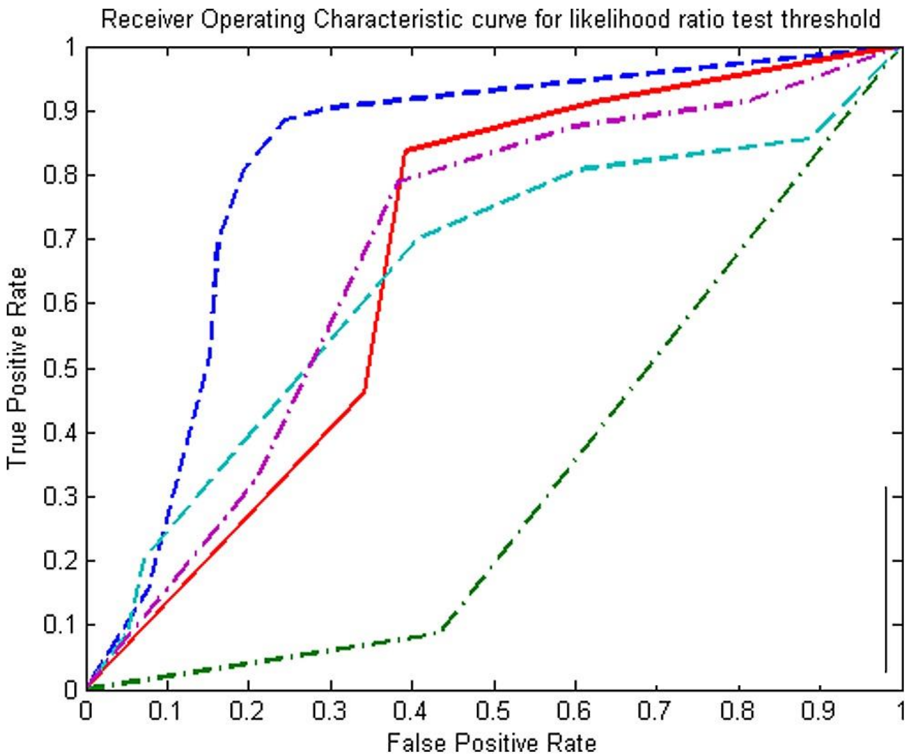
Decreasing threshold t → easier to classify examples as Pos

(inversely for increasing t)

ROC curves & AUC

$$IF\ s(\boldsymbol{x}) > t\ THEN\ predict\ y = Pos$$

Decreasing threshold t → easier to classify examples as Pos



→ **TPR (↑ or same)** $\frac{TP}{TP+FN}$

FPR (↑ or same) $\frac{FP}{TN+FP}$

Choose t offering **desired tradeoff**

Can **choose** among multiple **algorithms**

Can use **area under the curve** as **scalar evaluation measure**

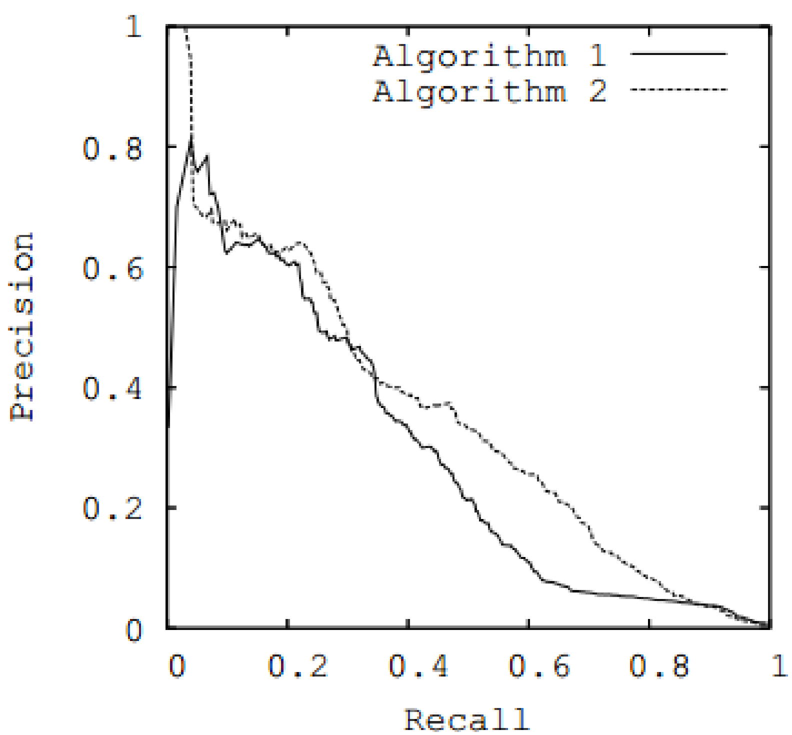
Precision-Recall curves & AUC

$$\boxed{IF\ s(\boldsymbol{x}) > \ t\ THEN\ predict\ y = \ Pos}$$

Decreasing threshold t → easier to classify examples as Pos

→ **Recall (↑ or same)** $\frac{TP}{TP+FN}$

Precision (?) $\frac{TP}{TP+FP}$



Choose t offering **desired tradeoff**

Can **choose** among multiple **algorithms**

Can use **area under the curve** as **scalar evaluation measure**

Expected cost (a.k.a. risk)

Can treat the **rarity of each class** as its **importance** (i.e. **cost of misclassifying**):

$$C_{FP} = 1/p_{NEG}$$
$$C_{FN} = 1/p_{POS}$$

(estimated on training set) $C_{TP} = C_{TN} = 0$

The goal then is to **minimize the expected cost**:

$$R = C_{FP} \text{ x FP} + C_{FN} \text{ x FN}$$

(expected FP, FN on test set)

Given a new example $\boldsymbol{x}$' this means:

$$\text{Predict } y = Pos \textit{ iff } \hat{p}(y = Pos|\boldsymbol{x}') > \frac{C_{FP}}{C_{FP} + C_{FN}}$$

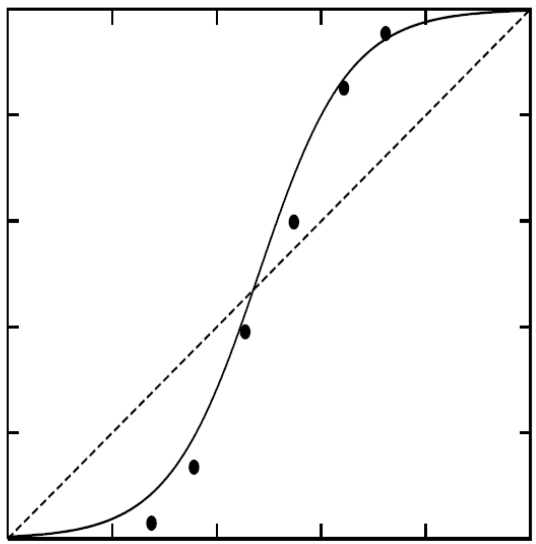
Threshold t known, but **need probability estimates**

Calibrating probability estimates

Using scores to make probabilistic decisions can be misleading!

Predict $y = Pos \text{ iff } \hat{p}(y = Pos|\boldsymbol{x}') > \dfrac{C_{FP}}{C_{FP} + C_{FN}}$

Model output (score)



Empirically Observed Probability

"Cost-sensitive boosting algorithms: Do we really need them?" Nikolaou, Edakunni, Kull, Flach, Brown. *Machine Learning. 2016*

I. Defining the problem

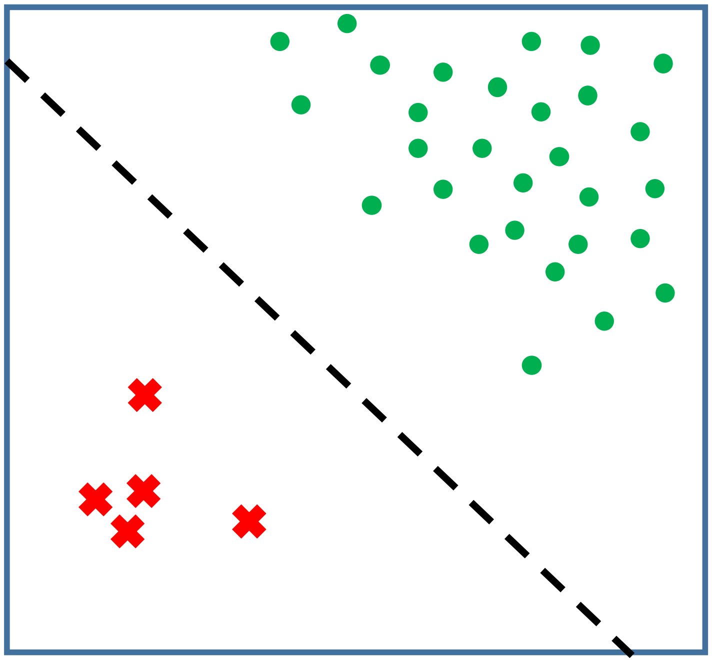
- Ensure as many of Pos predictions are indeed Pos
  Precision (PPV)
- Ensure as many of Pos examples are predicted as Pos
  Recall (a.k.a. TPR or Sensitivity)
- Achieve a (weighted) balance of the above
  F_β-measure; Precision-Recall Curve & AUC; …
- Achieve good performance across classes
  G-mean; ROC Curve & AUC; …
- Minimize expected cost (risk) of classifications
  Calibrate prob. estimates, then minimize risk; Cost Curves & AUC; …
- Maximize TPR for a given maximum FPR
  (Neyman-Pearson detection)

Most sensible for 'needle-in-a-haystack' problems

II. Solving the problem

- **Do nothing** special
- **Balance the dataset**
  - Oversample minority and/or undersample majority class
  - Synthetic examples
- **Modify algorithm** to favour rare class (cost-sensitive learning)
  - Pre-weight examples / modify loss function / shift decision threshold
  - Calibrate probability estimates
- **Devise a new algorithm** specifically for the problem at hand
- Treat as an **anomaly detection** problem
- **Get more minority class data** (might be infeasible / costly)

Imbalance might not be a problem



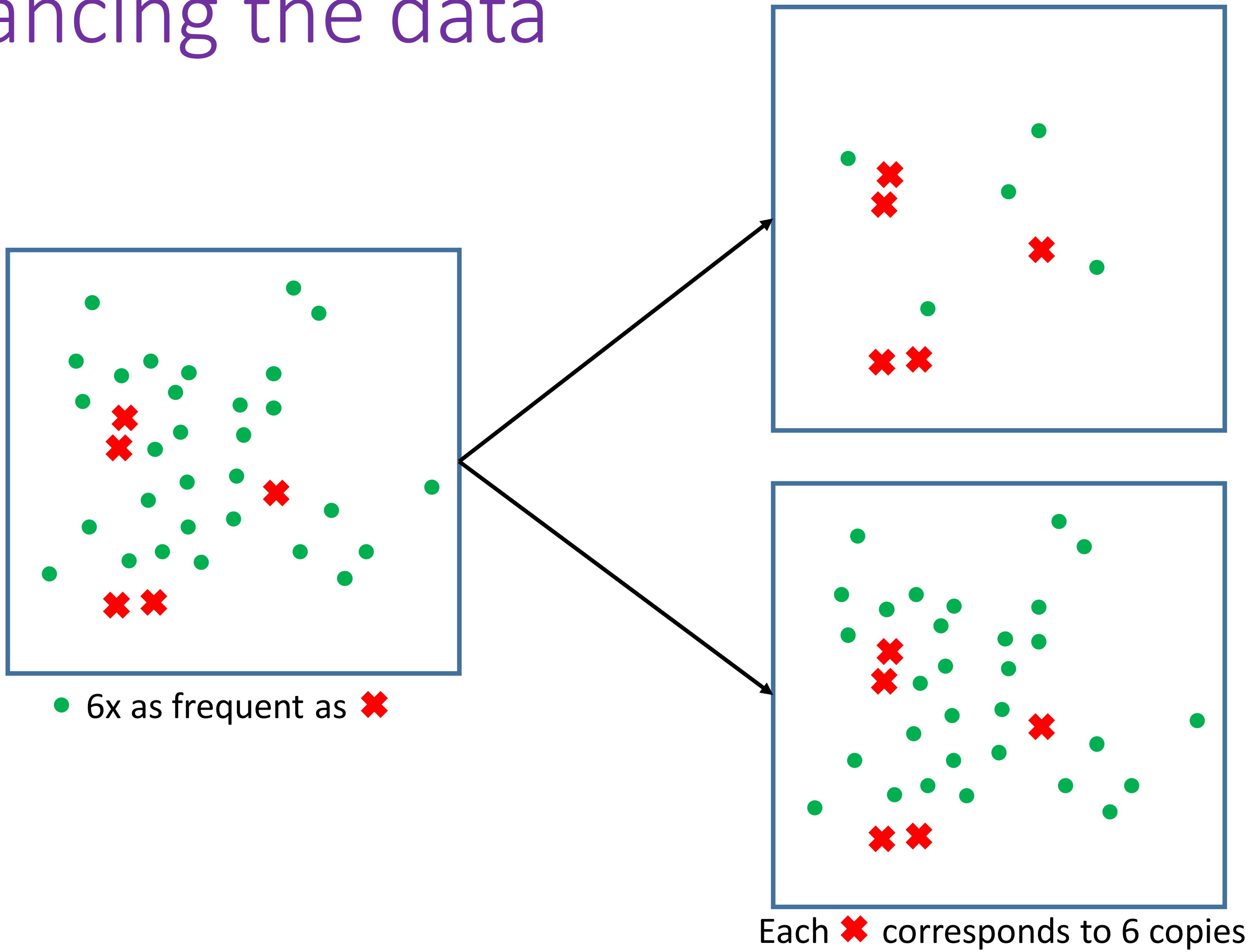
Data **separable** (not necessarily ‘linearly’) **by model**: no need to do anything!

So, before anything else try out **different models with different assumptions**

Might still want to bias the decision boundary in favour of minority class

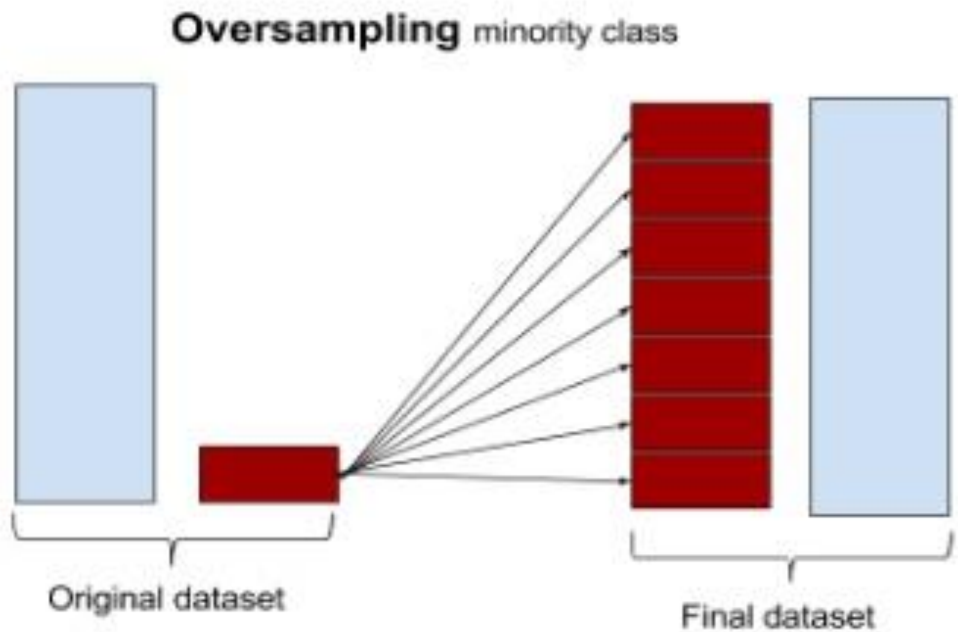
Problems start when we are forced to misclassify examples!

Balancing the data



Oversampling minority class

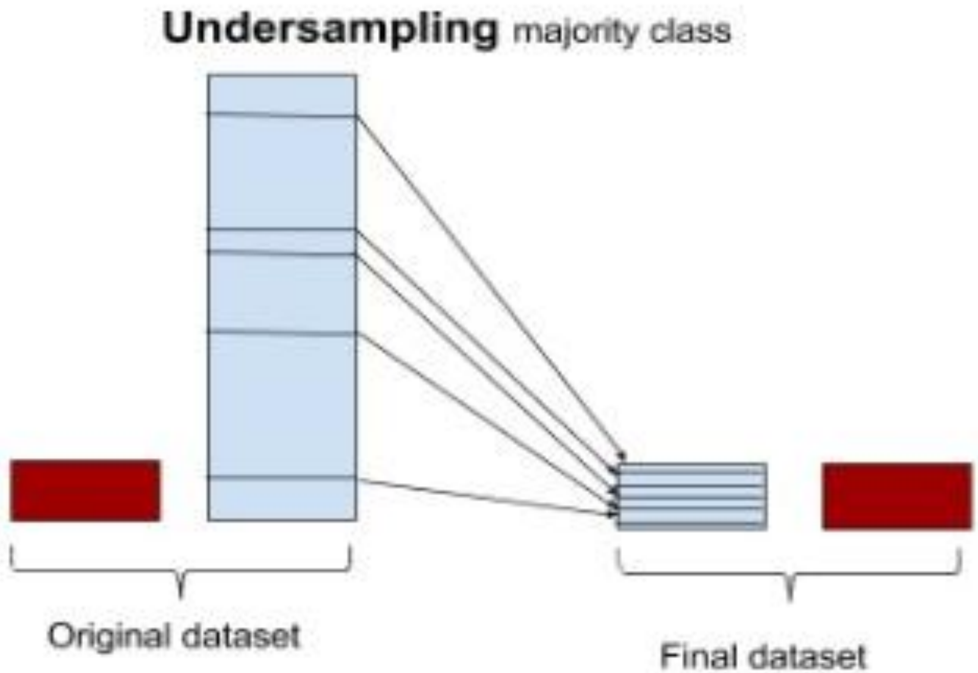
Create balanced dataset by **replicating minority examples**



- Cons: **variables appear to have lower variance** than they do
- Pros: **replicates errors** -if classifier A commits 1 FN on orig. data & minority data replicated x6, A will make 7 FNs on new set

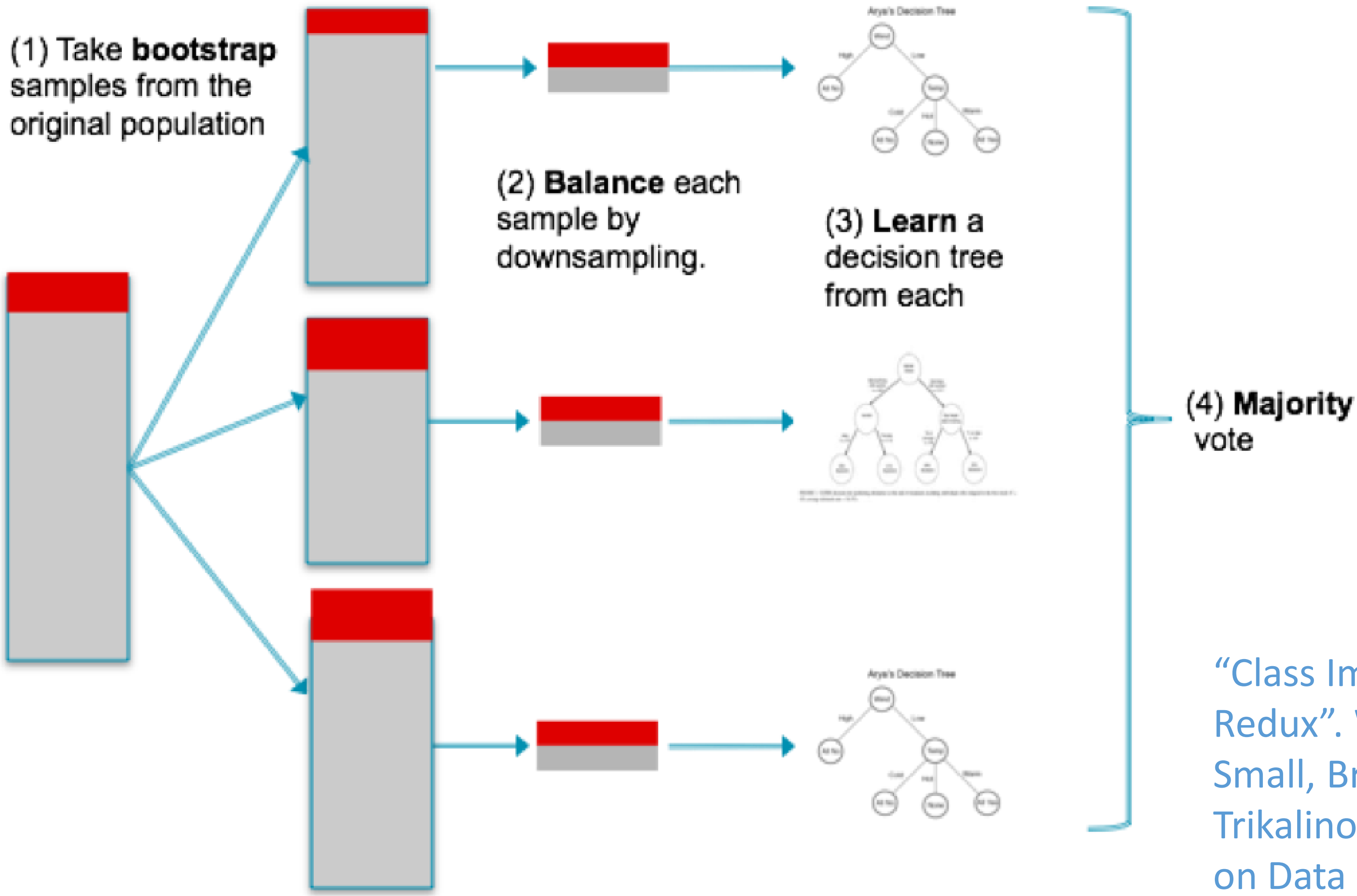
Undersampling majority class

Balance dataset by randomly **discarding majority examples**



- Cons: **variables appear to have higher variance** than they do; '**data is lost**'
- Pros: Can alleviate cons with **bagging**

Bagged undersampling (Blagging)

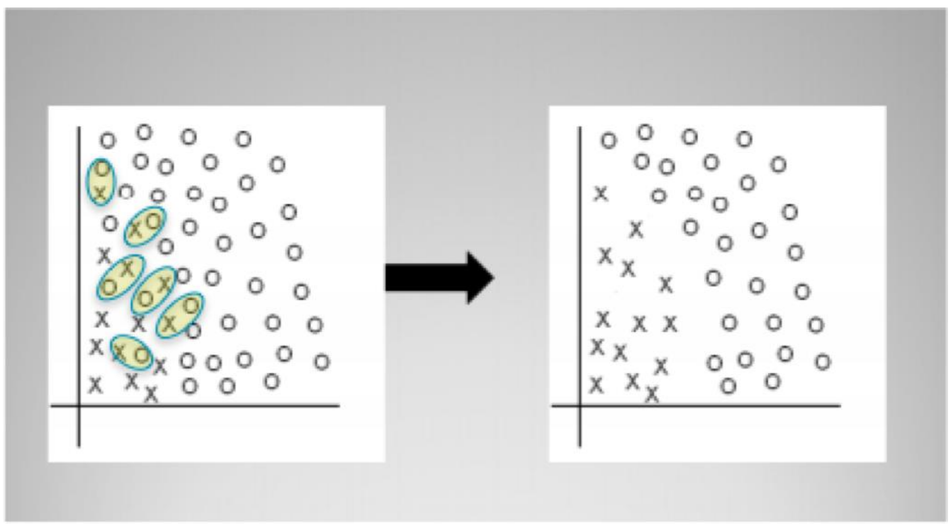


"Class Imbalance, Redux". Wallace, Small, Brodley and Trikalinos. IEEE Conf on Data Mining. 2011

Nearest neighbor techniques (Tomek)

Neighbourhood-based undersampling rather than random

- **Pair examples** of **opposite classes** that are each other's **nearest neighbors**…

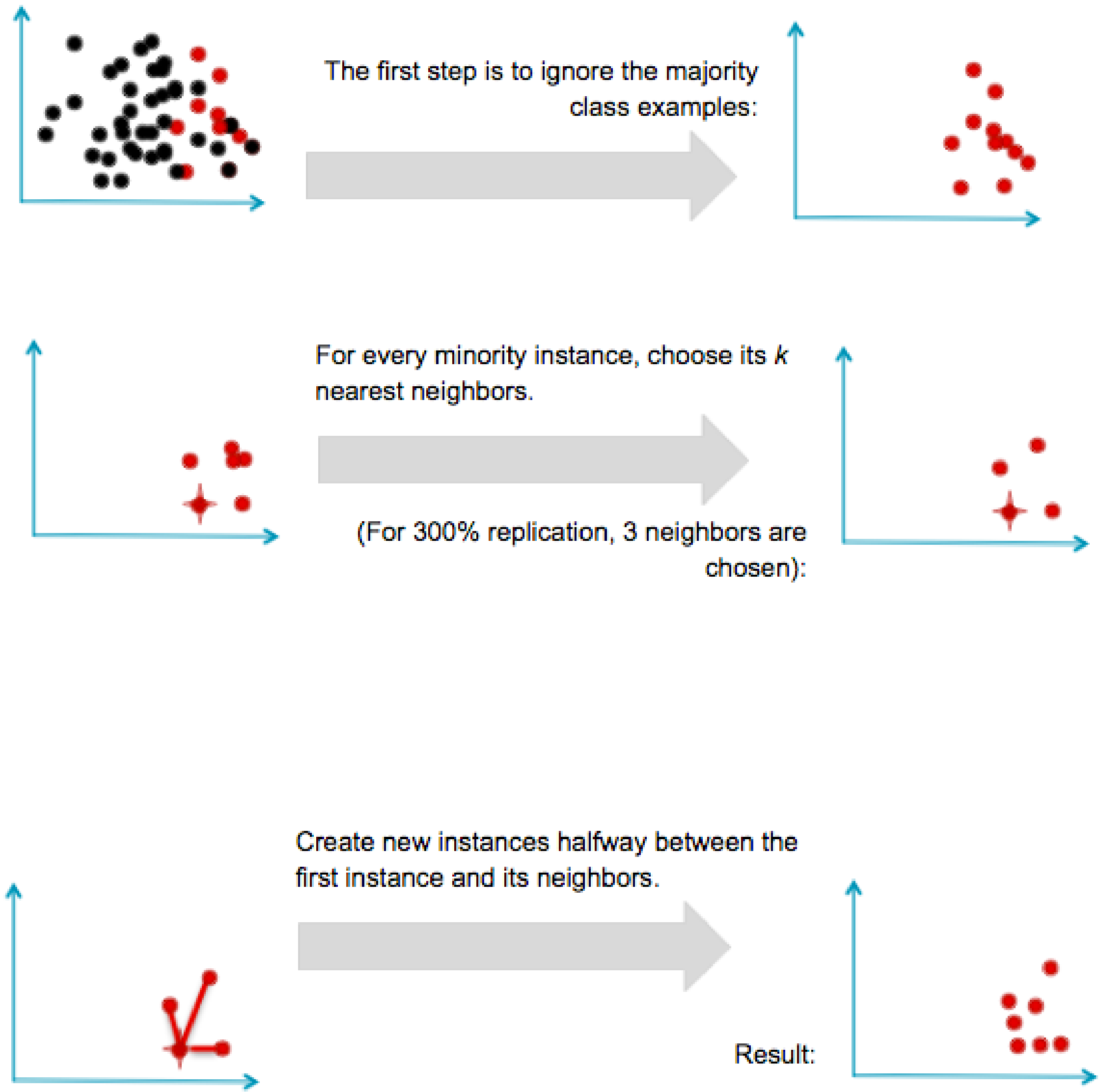


"An Experiment with the Edited Nearest-Neighbor Rule", Tomek. IEEE Trans. on Systems, Man, and Cybernetics. 1976

- …then **remove the majority instance of the pair**

Creating synthetic examples

- SMOTE: **create new minority examples by interpolating between existing ones**

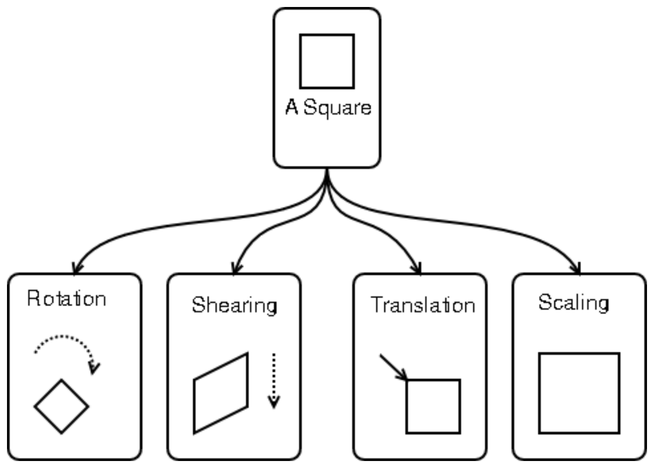


"SMOTE: Synthetic Minority Over-sampling Technique". Chawla, Bowyer, Hall, Kegelmeyer. Journal of Artificial Intelligence Research. 2002

Lots of variants…

Data augmentation

- Often, can create **new examples of the minority class** by **applying transformations to existing ones**



- Apply transformations that are **preserving the class** & **can be encountered in practice** **(use domain knowledge)**
- Some **specialized algorithms** are **already built to ignore certain types of transformations**, so this won't help

Take home messages

- Know **what you want** your classifier to do
- Avoid **eval. measures****loss functions** with **trivial optimizers**
- Inspect **confusion matrix** to spot classifier's **weaknesses**
- One model, many classifiers (**threshold manipulation**)
- When using **probability estimates**, **calibrate** them
- When **undersampling**, couple it with **bagging**
- When generating **synthetic data**, do so **reasonably** (**dom. knowledge**)
- You have **many tools at your disposal**, use them all

Further reading

- Tom Fawcet's blog post on '**Learning from Imbalanced Classes**': https://svds.com/learning-imbalanced-classes/

  (Some material from this was used in my talk)
- My i-python tutorial on **cost-sensitive boosting algorithms and calibration**: https://github.com/nnikolaou/Cost-sensitive-Boosting-Tutorial
- He, Haibo, and Edwardo A. Garcia. '**Learning from imbalanced data.**' *IEEE Transactions on knowledge and data engineering* (2009)
- Rich Caruana and Alexandru Niculescu-Mizil. **'An empirical comparison of supervised learning algorithms.'** ICML (2006)
- Bianca Zadrozny and Charles Elkan. **'Transforming classifier scores into accurate multiclass probability estimates.'** *KDD (*2002)

Further reading

- Lavrač N., Flach P., Zupan B. **‘Rule Evaluation Measures: A Unifying View.’** Inductive Logic Programming. (1999).
- Peter A. Flach. **‘The geometry of ROC space: understanding machine learning metrics through ROC isometrics.’** ICML 2003
- Paula Branco, Luís Torgo, and Rita P. Ribeiro. **‘A Survey of Predictive Modeling on Imbalanced Domains.’** *ACM Comput. Surv.* (2016)
- Saito, Takaya, and Marc Rehmsmeier. **‘The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets.’** *PloS one* (2015)

Thank you!

Questions?

Additional Slides
(not used in talk)

What **not** to do

- **Accuracy / misclassification error**

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad Error = 1 - Accuracy$$

  - Treats **all types of errors equally**
  - Can get a **nearly perfect score by predicting every example as Neg**

- **Minimize rare class misclassifications (FNs)**
  - Assigns **zero importance to frequent class errors (FPs)**
  - Can get a **perfect score by predicting every example as Pos**

What **not** to do

- **Maximize just Precision or just Recall**

$$Precision = \frac{TP}{TP+FP}$$ (**1 if a single Pos prediction that is indeed Pos**)

$$Recall = \frac{TP}{TP+FN}$$ (**1 if all examples are predicted Pos**)

- **Use uncalibrated probability estimates**
  - Don't make decisions using **unreliable estimates** $\hat{p}(y = Pos|\boldsymbol{x})$

Calibrating probability estimates

- Use **scoring rules** **(Brier score, log-loss)** to check (pre & post calibration)

  "Strictly Proper Scoring Rules, Prediction, and Estimation". Gneiting, Raftery Journal of the American Statistical Association. 2007

- **Isotonic regression**, **plat scaling** (should **correct for class imbalance**)

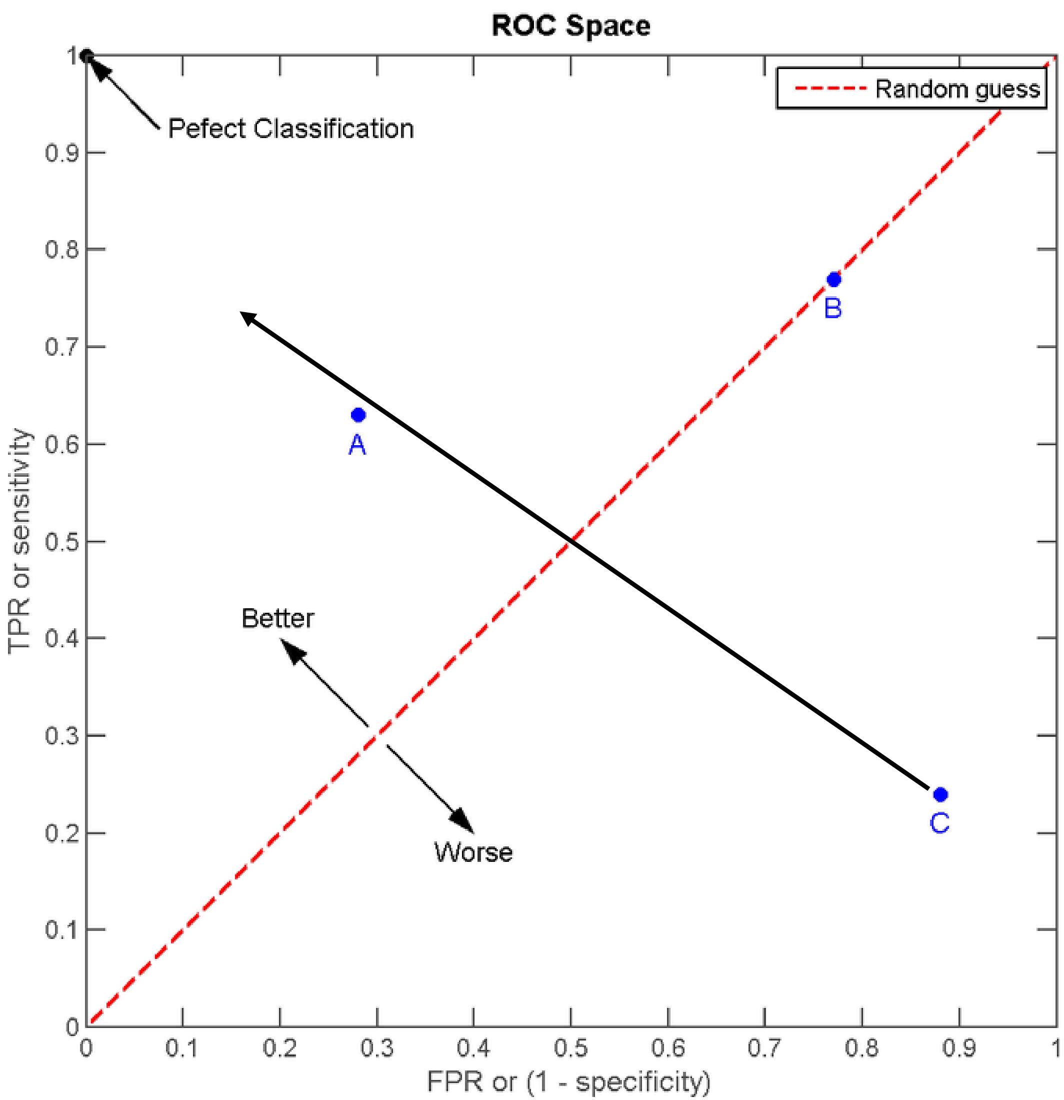
  "Predicting good probabilities with supervised learning". Niculescu-Mizil, Caruana. ICML. 2005

  ''Probabilities for SV machines''. Platt. Advances in Large Margin Classifiers. 2000

- Might need to use different loss function during calibration when your goal differs from risk minimization

  "Classifier Calibration". Flach. Encyclopedia of Machine Learning and Data Mining. 2016

ROC curves & AUC



Modifying the algorithm

- **Before training**: Reweight examples
  (not really modifying alg. but equiv. in expectation…)

  Can be equiv. to **oversampling minority w/o synthetic data**

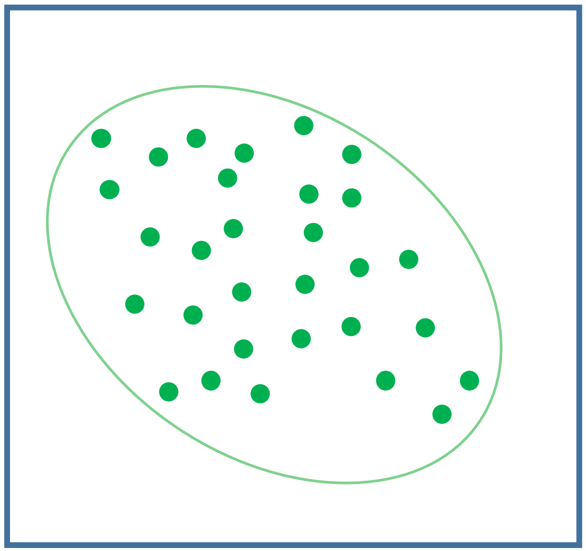
- **During training**: Change the loss function

  Use **appropriate measure** (see Part I)

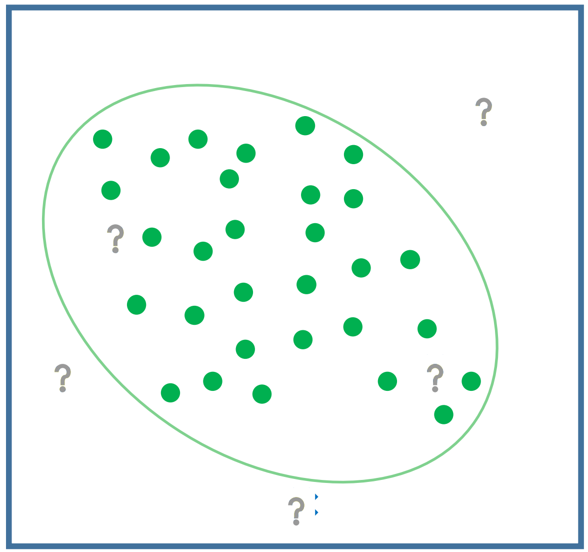
- **After training**: Shift the decision threshold

  Discussed in Part I; can set threshold with **cross-validation**
  or -if imbalance/costs known- using **decision theory**

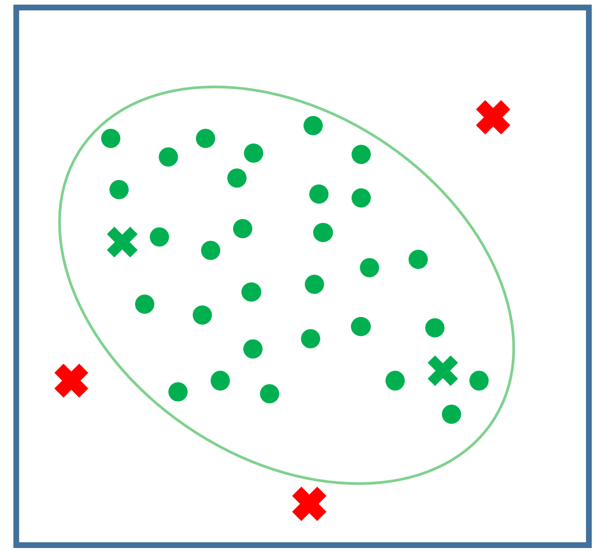
Anomaly detection



Only model majority class



Given new datapoints ?



Assign them to minority class only if '**significantly different**' than majority class

"Anomaly detection : a survey". Chandola, Banerjee, Kumar. ACM Computing Surveys. 2009

"Novelty detection : a review". Markou, Singh. Signal Processing. 2003