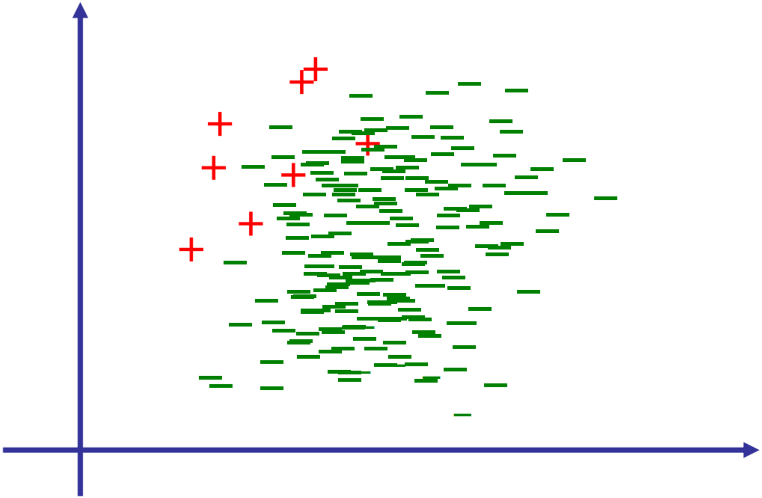
Intro ML: Tutorial on Class Imbalance

Vincent Fortuin, [1] Gideon Dresdner[1]

[1]ETH Zurich

Motivation



Examples

- Diagnosis of rare diseases
- Spam detection
- Fault detection in manufacturing
- Fraud discovery
- And many more...

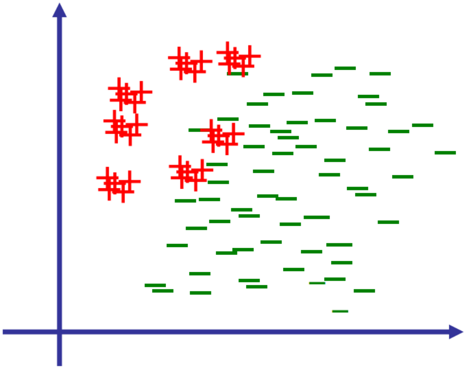
Class imbalance in practice

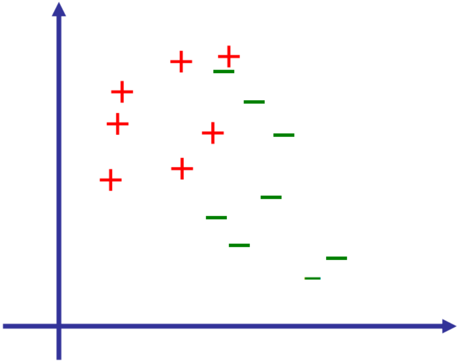
- Let's assume w.l.o.g. that we have a binary classification problem where the positive class is rare

- As we saw on the previous slide, many interesting problems in the real world have this property

- There are different approaches to deal with it:
  - Upsampling
  - Downsampling
  - Cost-sensitive loss functions

Upsampling and downsampling





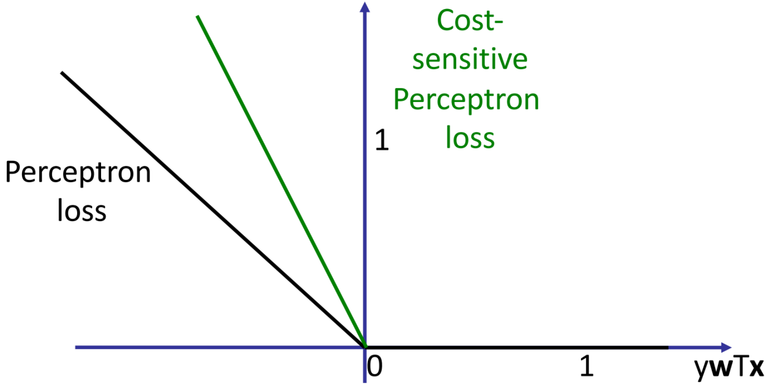
Upsampling and downsampling: A tradeoff

- Neither of these two methods is perfect

- Upsampling …
  - Uses some arbitrary augmentation technique
  - Might overfit to the data examples in the minority class
  - But it uses all the available data

- Downsampling …
  - Throws away data from the majority class
  - But it is faster

Cost-sensitive loss functions



$$\ell(\mathbf{w}; \mathbf{x}, y) = c_y \max(0, -y\mathbf{w}^T\mathbf{x})$$

Performance measures

True label

Predicted label

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Performance measures

- Accuracy $= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- True positive rate (TPR) / Recall $= \frac{\text{TP}}{\text{TP}+\text{FN}}$
- False positive rate (FPR) $= \frac{\text{FP}}{\text{FP}+\text{TN}}$
- Precision $= \frac{\text{TP}}{\text{TP}+\text{FP}}$
- F-Score $= \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}} = 2\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$

Performance measures: overview

		True condition		
	Total population	Condition positive	Condition negative	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	

Source: https://en.wikipedia.org/wiki/Precision_and_recall#Definition_(classification_context)

Multi-class performance measures

- In a multi-class setting, we can still compute the discussed measures for each class individually

- For any class, we can do that by considering that class as being the positive label and all other classes as being negative

- We can then either report the measures for each class separately (e.g., sklearn.metrics.classification_report) or average them

Multi-class performance measures: Averaging

- Micro-averaging: Take the average of the TPs, FPs, TNs, and FNs across all classes and use those to compute the different performance measures

- Macro-averaging: Compute the different measures on every class separately and then average across all classes

- Weighted averaging: Like macro-averaging, but every measure gets weighted by the true number of samples in that class (makes a difference for imbalanced data)

Multi-class performance measures: Averaging

- Micro-averaging:
  $TP_{\text{micro}} = \frac{\sum_{c=1}^{C} TP_c}{C}; \quad \text{prec}_{\text{micro}} = \frac{TP_{\text{micro}}}{TP_{\text{micro}} + FP_{\text{micro}}}$

- Macro-averaging: $\text{prec}_c = \frac{TP_c}{TP_c + FP_c}; \quad \text{prec}_{\text{macro}} = \frac{\sum_{c=1}^{C} \text{prec}_c}{C}$

- Weighted averaging:
  $n_c = |\{i : y_i = c\}|; \quad \text{prec}_{\text{weighted}} = \frac{\sum_{c=1}^{C} n_c \, \text{prec}_c}{N}$

Caveats: Micro-averaging

- Every prediction error of the model is a FP for one class and a FN for another one
- Thus, the micro-averaged FP will be equal to the micro-averaged FN
- In effect, this means that

$$\textit{micro-precision} = \textit{micro-recall} = \textit{micro-F-Score}$$

Motivation for ROC- and PR-curves

- Say you have trained an SVM.

Motivation for ROC- and PR-curves

- Say you have trained an SVM.How do you use it to make predictions?

Motivation for ROC- and PR-curves

- Say you have trained an SVM.How do you use it to make predictions?

- Right – you use $\text{sign}(w^T x) = \begin{cases} 1 & w^T x > 0 \\ 0 & \text{otherwise} \end{cases}$

Motivation for ROC- and PR-curves

- Say you have trained an SVM.How do you use it to make predictions?

- Right — you use $\text{sign}(w^T x) = \begin{cases} 1 & w^T x > \cancel{0}\, C \\ 0 & \text{otherwise} \end{cases}$

- What if you vary $w^T x > C$ for many C's, not just 0?

Motivation for ROC- and PR-curves

- Say you have trained an SVM.How do you use it to make predictions?

- Right — you use $\mathrm{sign}(w^T x) = \begin{cases} 1 & w^T x > \cancel{0}\,C \\ 0 & \text{otherwise} \end{cases}$

- What if you vary $w^T x > C$ for many C's, not just 0?

- Think: if C is large

Motivation for ROC- and PR-curves

- Say you have trained an SVM.How do you use it to make predictions?

- Right – you use $\text{sign}(w^T x) = \begin{cases} 1 & w^T x > \cancel{0}\, C \\ 0 & \text{otherwise} \end{cases}$

- What if you vary $w^T x > C$ for many C's, not just 0?

- Think: if C is large…it's "hard" to get classified as positive but you are sure of decision

Motivation for ROC- and PR-curves

- Say you have trained an SVM.How do you use it to make predictions?

- Right — you use $\mathrm{sign}(w^T x) = \begin{cases} 1 & w^T x > \cancel{0}\, C \\ 0 & \text{otherwise} \end{cases}$

- What if you vary $w^T x > C$ for many C's, not just 0?

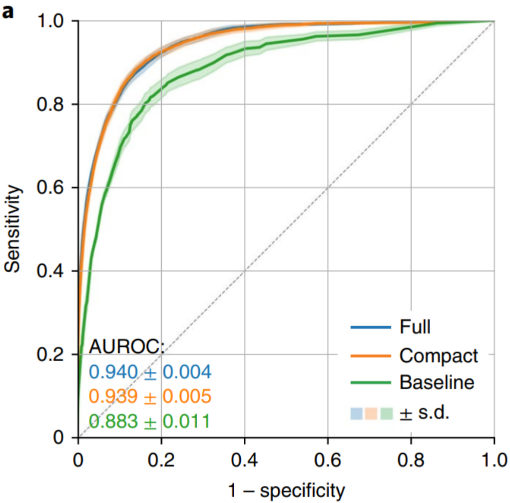
- Think: if C is large…it's "hard" to get classified as positive but you are sure of decision $\implies$ trade-off

ROC- and PR-curves

- Need to compare different quantities for each value of C. There are two classical comparisons:
    1. TPR vs. FPR, called Receiver Operator Characteristic (ROC).
    2. precision vs. recall , called PR curve

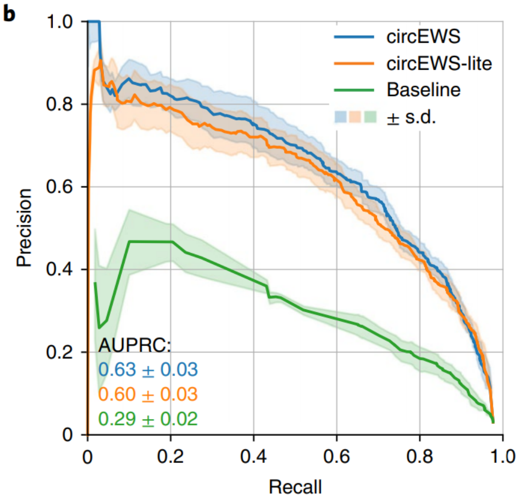
- Natural summary of a curve — area under (AU) curve (AUC) — axes are set so that larger area is better.

ROC-curve



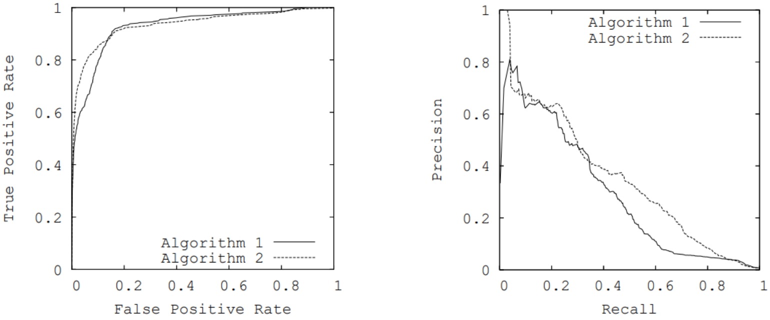
Source: https://www.nature.com/articles/s41591-020-0789-4

PR-curve



Source: https://www.nature.com/articles/s41591-020-0789-4

ROC vs. PR (Davis and Goadrich, 2006)



(a) Comparison in ROC space

(b) Comparison in PR space

Figure 1. The difference between comparing algorithms in ROC vs PR space

ROC vs. PR (Davis and Goadrich, 2006)

- Which should we optimize for AUPRC or AUROC?
- Result: strictly better (at all points) ROC-curve iff. strictly better PR-curve
- But, unrealistic assumption. If one alg. is *not strictly* better, then no guarantee and need to make a modelling decision.

Takeaways: ROC vs. PR

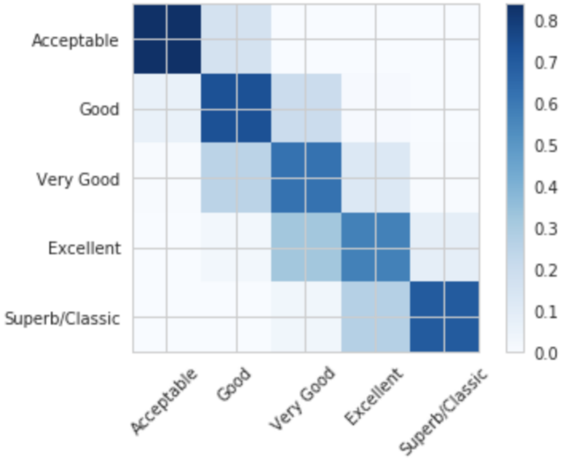
- The random classifier has an AUROC of 0.5 and an AUPRC of the rate of positive examples

- Optimizing for AUROC generally yields different algorithms than optimizing for AUPRC

- Imbalanced data might skew the ROC-curves and make them look more similar than the PR-curves

- For imbalanced data, the PR-curve might be more informative than the ROC-curve

Multi-class confusion matrix



Source: Source (written) . Source code

Exercise: Computing confusion matrix

Compute the confusion matrix for the following output of (prediction, actual) pairs for a binary classifier:

$$\{(0,0),(1,1),(0,1),(1,0),(1,1),(1,0)\}$$

Exercise: Computing confusion matrix

Compute the confusion matrix for the following output of (prediction, actual) pairs for a binary classifier:

$$\{(0,0),(1,1),(0,1),(1,0),(1,1),(1,0)\}$$

Summarize as:

$$\begin{aligned}\#(0,0) =& \quad 1\\ \#(1,1) =& \quad 2\\ \#(0,1) =& \quad 1\\ \#(1,0) =& \quad 2\end{aligned}$$

Exercise: Computing confusion matrix

Compute the confusion matrix for the following output of (prediction, actual) pairs for a binary classifier:

$$\{(0,0),(1,1),(0,1),(1,0),(1,1),(1,0)\}$$

Summarized as a table:

		Actual 0	1
	0	1	1
Predicted	1	2	2

Remark: confusion matrix for 2 classes

Confusion matrix for binary classification is the same as the table on slide 8.

Exercise: Computing metrics from confusion matrix

- Given a confusion matrix for 3-class classification model,

		Actual		
		Sunset	Sunrise	Midday
Predicted	Sunset	9	4	1
	Sunrise	1	3	1
	Midday	0	0	1

Exercise: Computing metrics from confusion matrix

1. Is this dataset balanced? If not, what is the most common class?
2. Speculate: what is the source of confusion for the classifier?
3. What is the accuracy?
4. What is the precision and recall for classes “sunrise” and “sunset”?

Exercise: Computing metrics from confusion matrix

1. Is this dataset balanced? If not, what is the most common class? Sunsets, count = 10
2. Speculate: what is the source of confusion for the classifier? Sunsets and sunrises look similar. Also, the labels are imbalanced.
3. What is the accuracy? $(9+3+1)/20 = 13/20 = .65$
4. What is the precision and recall for classes “sunrise” and “sunset”?
   sunset: $P = 9/(9+4+1) \quad R = 9/(9+1+0)$
   sunrise: $P = 3/(3+1+1) \quad R = 3/(4+3+0)$

References

1. Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).

END OF PRESENTATION
BEGININNG OF Q&A