Profit Curves and Imbalanced Classes

Today’s objectives and plan

Afternoon:

- Fraud detection
- Cost-sensitive methods (profit curves)
- Sampling methods (undersampling, oversampling, SMOTE)
- Pair programming:
  - Profit curve by hand
  - Implement profit curve
  - Implement sampling methods

Motivating example: fraud detection

(in reality, fraudulent over legitimate transaction rate is ~ 1 / 100,0000)

		Actual	
		Fraudulent **p**	**Legitimate** **n**
Predicted	**Y**	Hit (TP) 40	Miss, Type II Error (FP) 50
	N	False Alarm, Type I Error (FN) 10	Correct rejection (TN) 900
		P = 50 << N	N = 950 >> P

Motivating example: fraud detection (cont.)

- Fraud classification datasets is very “imbalanced”
  - Most transactions are legitimate; very few are fraudulent
- Classifiers are typically more sensitive to detecting the majority class and less sensitive to the minority class
  - Output will be biased and will over-predict the majority class (specificity close to 1; sensitivity near 0)
    - → maximizing accuracy is the wrong objective
- Business costs of false positives and false negatives are very different
  - High cost in missing to detect a fraudulent transaction can be very costly vs. much lower cost of flagging a legitimate transaction as fraudulent
    - → again, maximizing accuracy is the wrong objective

Motivating example: fraud detection (cont.)

		Actual	
		Fraudulent **p**	**Legitimate** **n**
Predicted	**Y**	$Sensitivity = TPR = \frac{TP}{P} = \frac{40}{50} = .8 = 80\%$	$FPR = \frac{FP}{N} = \frac{50}{950} = .05 = 5\%$
	N		$Specificity = TNR = \frac{TN}{N} = \frac{900}{950} = .95 = 95\%$

Solutions

- Cost-sensitive learning:
    - Profit curves/thresholding
    - Modified objective functions

- Sampling:
    - Oversampling
    - Undersampling
    - SMOTE (Synthetic Minority Oversampling TEchnique)

Cost-Sensitive Learning
Profit Curves/Thresholding

(1) First, start with the confusion matrix

		Actual	
		P	**n**
Predicted	**Y**	*TP* (True Positive)	*FP* (False Positive)
	N	*FN* (False Negative)	*TN* (True Negative)
		$P = TP + FN$	$N = FP + TN$

and normalize to rate to get expected rates (probability matrix)

		Actual	
		p	**n**
Predicted	**Y**	$\mathrm{P}(Y,p) = \frac{TP}{P+N}$	$\mathrm{P}(Y,n) = \frac{FP}{P+N}$
	N	$\mathrm{P}(N,p) = \frac{FN}{P+N}$	$\mathrm{P}(N,n) = \frac{TN}{P+N}$

Expected rates for our motivating example

		Actual	
		p	**n**
Predicted	**Y**	$\mathrm{P}(Y,p) = \frac{40}{50+950} = .04 = 4\%$	$\mathrm{P}(Y,n) = \frac{50}{50+950} = .05 = 5\%$
	N	$\mathrm{P}(N,p) = \frac{10}{50+950} = .01 = 1\%$	$\mathrm{P}(N,n) = \frac{900}{50+950} = .9 = 90\%$

(note: We will use the conditional probability form in the next few slides)

		Actual	
		p	**n**
Predicted	**Y**	$\mathrm{P}(Y,p) = \mathrm{P}(Y\|p) \cdot \mathrm{P}(p)$	$\mathrm{P}(Y,n) = \mathrm{P}(Y\|n) \cdot \mathrm{P}(n)$
	N	$\mathrm{P}(N,p) = \mathrm{P}(N\|p) \cdot \mathrm{P}(p)$	$\mathrm{P}(N,n) = \mathrm{P}(N\|n) \cdot \mathrm{P}(n)$

(2) Then define the cost/benefit matrix (defined from the business situation)

		Actual	
		p	**n**
Predicted	**Y**	$\mathrm{b}(Y,p)$ (‘b’ as “benefit”)	$\mathrm{c}(Y,n)$ (‘c’ as “cost”)
	N	$\mathrm{c}(N,p)$	$\mathrm{b}(N,n)$

Cost/benefit matrix for our motivating example

		Actual	
		p	**n**
Predicted	**Y**	0	-$5
	N	-$5,000	0

(3) Combining the expected rates and cost/benefit matrices, we can derive expected profit:

$$E[profit] = \mathrm{P}(Y,p) \cdot \mathrm{b}(Y,p) + \mathrm{P}(Y,n) \cdot \mathrm{b}(Y,n)$$
$$+\mathrm{P}(N,p) \cdot \mathrm{b}(N,p) + \mathrm{P}(N,n) \cdot \mathrm{b}(N,n)$$

$$E[profit] = \mathrm{P}(Y|p) \cdot \mathrm{P}(p) \cdot \mathrm{b}(Y,p) + \mathrm{P}(Y|n) \cdot \mathrm{P}(n) \cdot \mathrm{b}(Y,n)$$
$$+\mathrm{P}(N|p) \cdot \mathrm{P}(p) \cdot \mathrm{b}(N,p) + \mathrm{P}(N|n) \cdot \mathrm{P}(n) \cdot \mathrm{b}(N,n)$$

$$E[profit] = [\mathrm{P}(Y|p) \cdot \mathrm{b}(Y,p) + \mathrm{P}(N|p) \cdot \mathrm{c}(N,p)] \cdot \mathrm{P}(p)$$
$$+[\mathrm{P}(Y|n) \cdot \mathrm{c}(Y,n) + \mathrm{P}(N|n) \cdot \mathrm{b}(N,n)] \cdot \mathrm{P}(n)$$

(This way of writing this last one nice because it allows us to change the ‘priors’ by hand and see what happens. The other terms don’t depend on the priors, so we can change the priors any way we’d like; but most likely we’ll change them to match real-world so that we have an idea of the profit in the real-world)

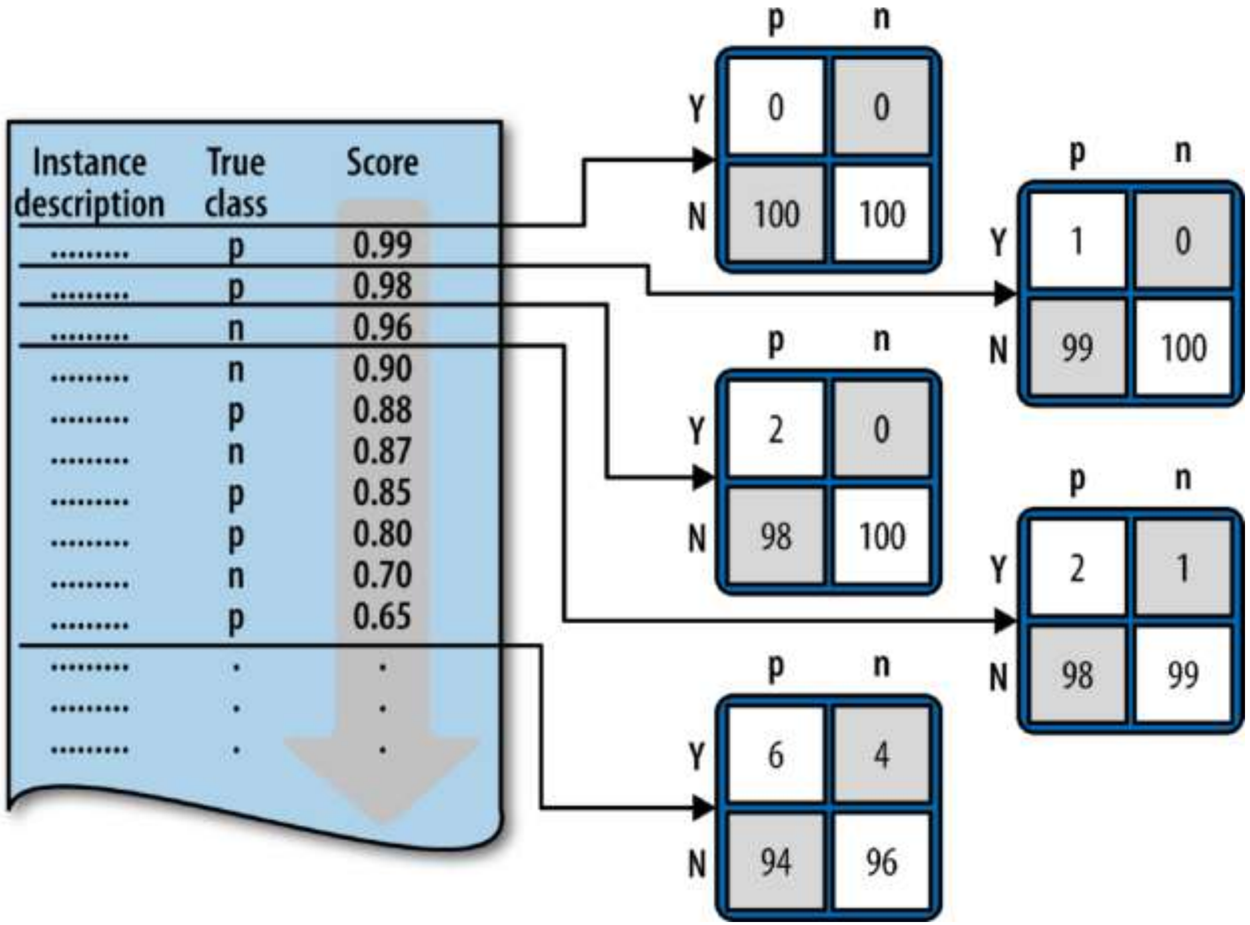
Expected profit for our motivating example

$$E[profit] =$$

$$\$0 \times 4\% + (-\$5) \times 5\%$$
$$+(-\$5{,}000) \times 1\% + \$0 \times 90\%$$

$$= -\$50.25$$

Building profit curves



Instance description	True class	Score
....	p	0.99
....	p	0.98
....	n	0.96
....	n	0.90
....	p	0.88
....	n	0.87
....	p	0.85
....	p	0.80
....	n	0.70
....	p	0.65
....	.	.
....	.	.
....	.	.

	p	n
Y	0	0
N	100	100

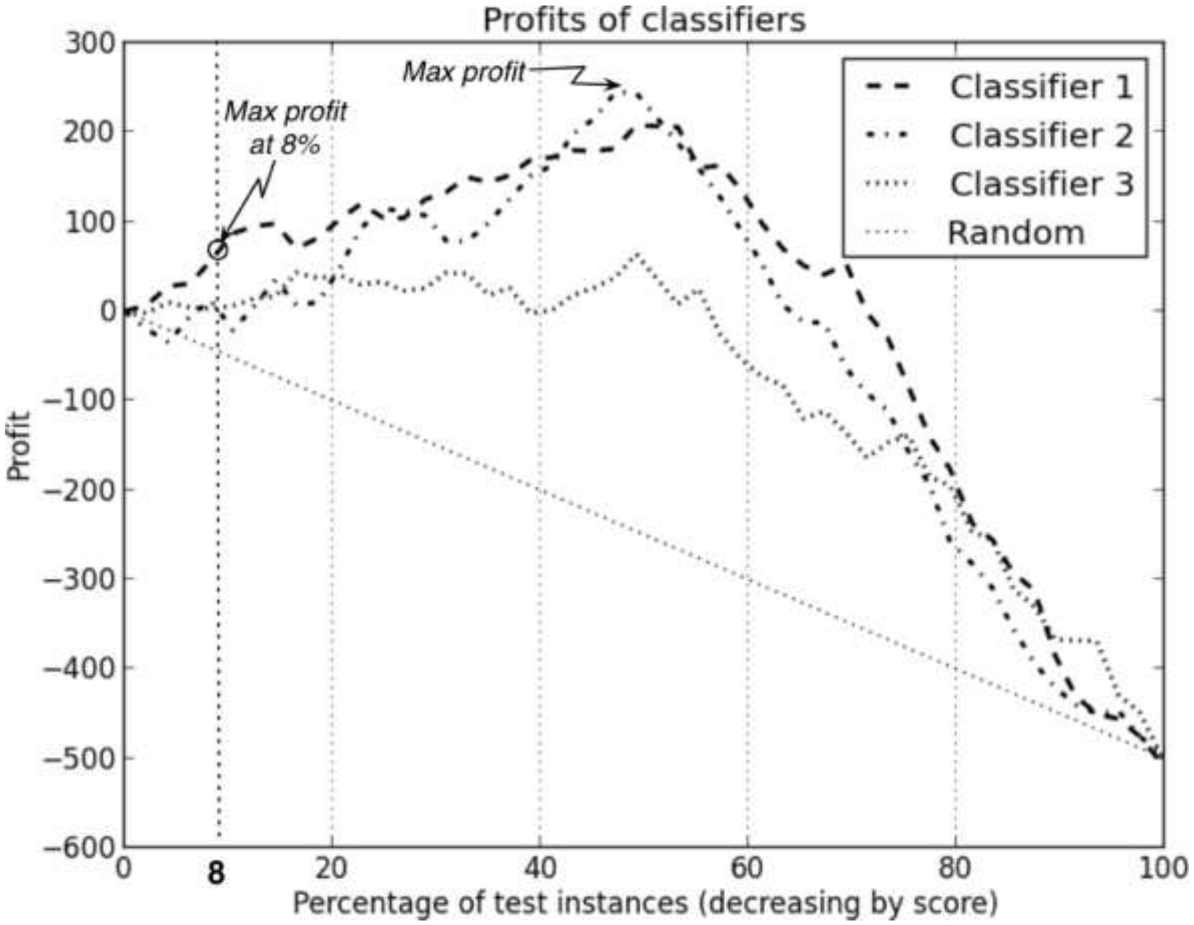
	p	n
Y	1	0
N	99	100

	p	n
Y	2	0
N	98	100

	p	n
Y	2	1
N	98	99

	p	n
Y	6	4
N	94	96

Building profit curves (cont.)



Cost-Sensitive Learning
Modified Objective Functions

Modified objective functions

- Models with explicit objective function can be modified to incorporate classification benefit/cost
  - E.g. logistic regression
- However, this will affect optimization
  - Because cost-sensitive logistic regression is not convex, the optimization algorithm might get stuck in a local optimum

Sampling

Undersampling

Undersampling randomly discards majority class observations to balance training sample

PRO: Reduces runtime on very large datasets

CON: Discards potentially important observations

Oversampling

Oversampling replicates observations (with replacement) from minority class to balance training sample

PRO: Doesn’t discard information

CON: Likely to overfit

Oversampling – Take 2

SMOTE – Synthetic Minority Oversampling Technique

Generates **synthetic** observations (using interpolation) from minority class

SMOTE: Algorithm

```
For each new synthetic observation:
      S1 ← randomly selected minority class observation
      S2 ← randomly selected neighbor of S1
      For every new feature of the synthetic observation:
            weight ← random value between 0 and 1
            new feature ← weight * S1’s feature
                              + (1 – weight) * S2’s feature
            # (i.e., interpolate each feature from S1 and S2)
```

(see also pseudocode in original SMOTE paper: https://www.jair.org/media/953/live-953-2037-jair.pdf)

What’s the right amount of over/under sampling?

- If you know the cost matrix:
  - Maximize profit curve over target proportion

- If you don’t know the cost matrix:
  - No clear answer
  - ROC plot’s AUC may be more useful

Cost-Sensitive Learning vs. Sampling

- Neither is strictly superior
- Some algorithms don’t have an obvious cost-sensitive adaptation, requiring sampling
- Oversampling tends to work better than undersampling on small datasets

(See also “Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?” http://storm.cis.fordham.edu/gweiss/papers/dmin07-weiss.pdf)

Afternoon pairing