Week 2 Video 1

Detector Confidence

Classification

- There is something you want to predict ("the label")
- The thing you want to predict is categorical

It can be useful to know yes or no

It can be useful to know yes or no

- The detector says you don’t have Ptarmigan’s Disease!

It can be useful to know yes or no

- But it’s even more useful to know how certain the prediction is

It can be useful to know yes or no

- But it’s even more useful to know how certain the prediction is
  - The detector says there is a 50.1% chance that you don’t have Ptarmigan’s disease!

Uses of detector confidence

Uses of detector confidence

- Gradated intervention
  - Give a strong intervention if confidence over 60%
  - Give no intervention if confidence under 60%
  - Give “fail-soft” intervention if confidence 40-60%

Uses of detector confidence

- Decisions about strength of intervention can be made based on cost-benefit analysis

- What is the cost of an incorrectly applied intervention?

- What is the benefit of a correctly applied intervention?

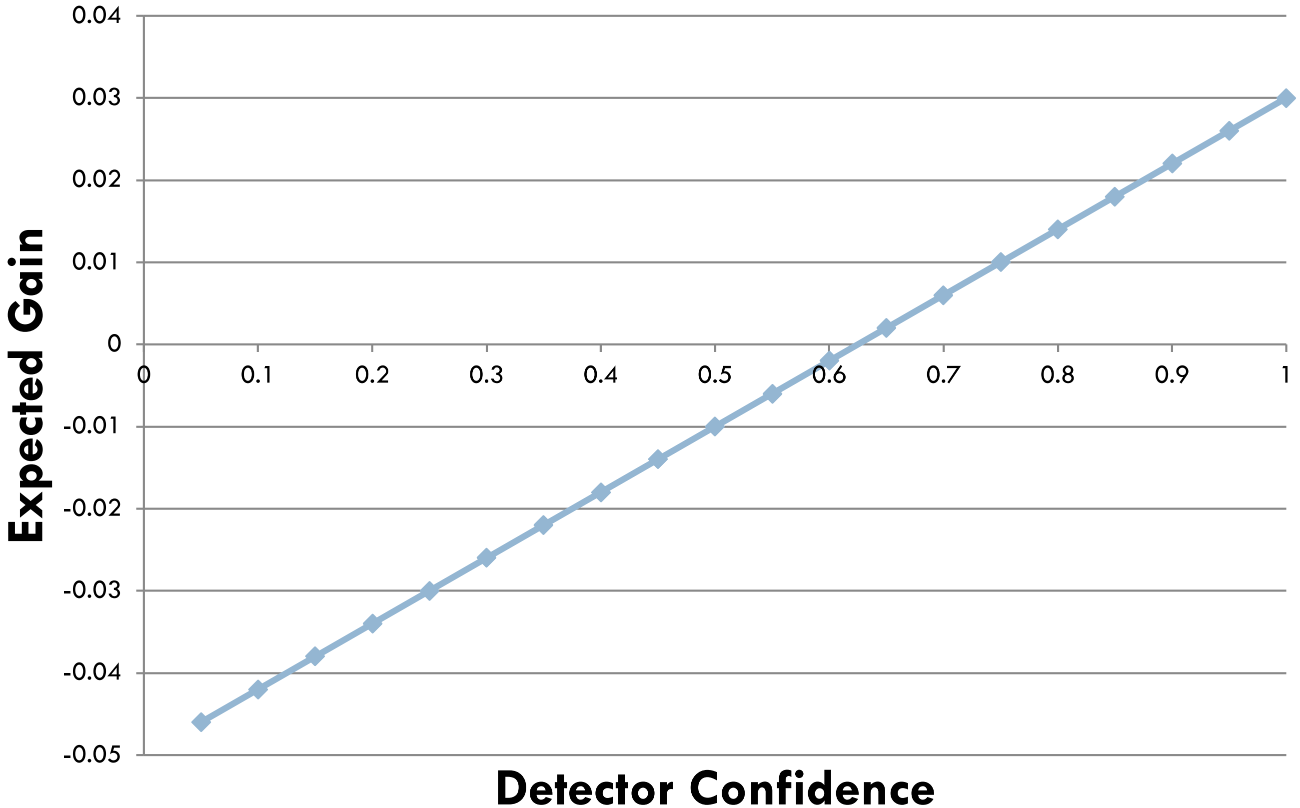
Example

- An incorrectly applied intervention will cost the student 1 minute
- Each minute the student typically will learn 0.05% of course content

- A correctly applied intervention will result in the student learning 0.03% more course content than they would have learned otherwise

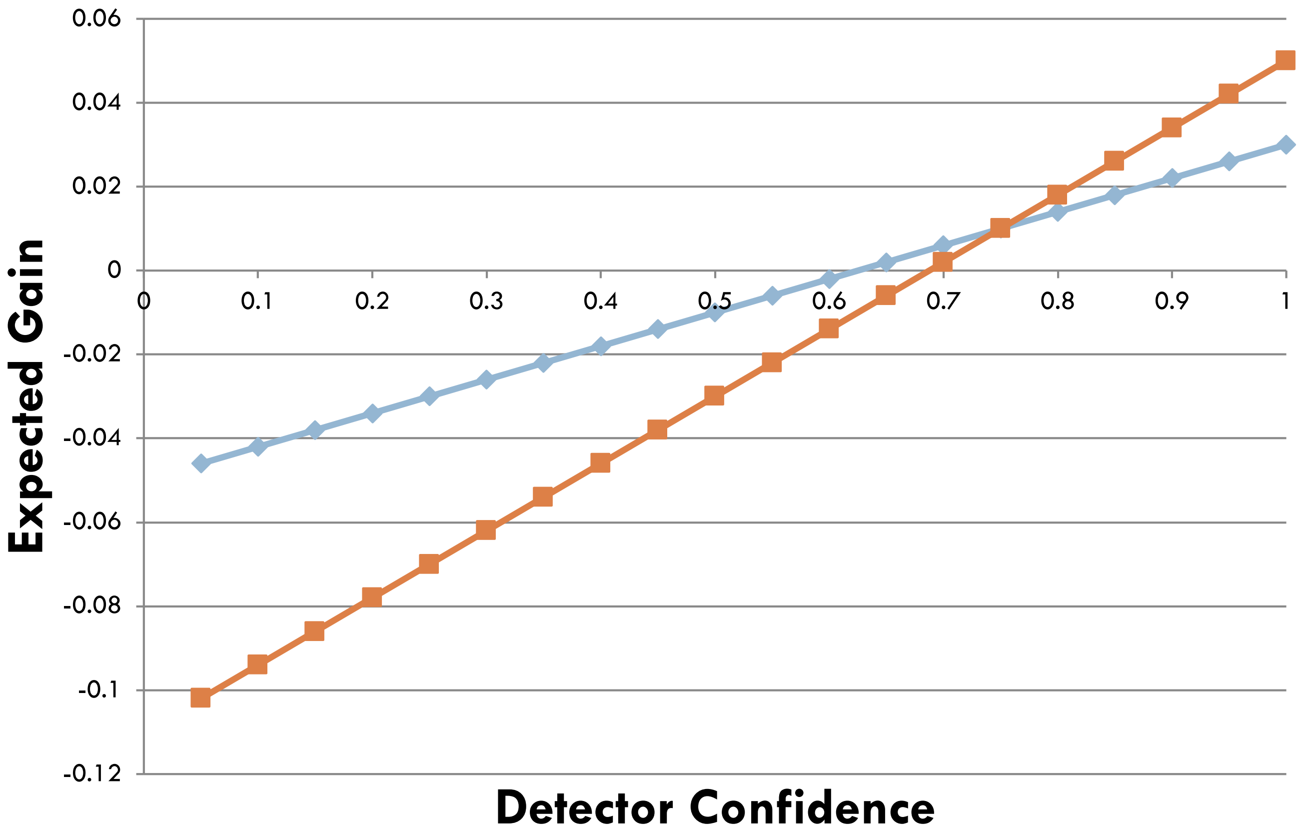
Expected Value of Intervention

- 0.03*Confidence – 0.05 * (1-Confidence)

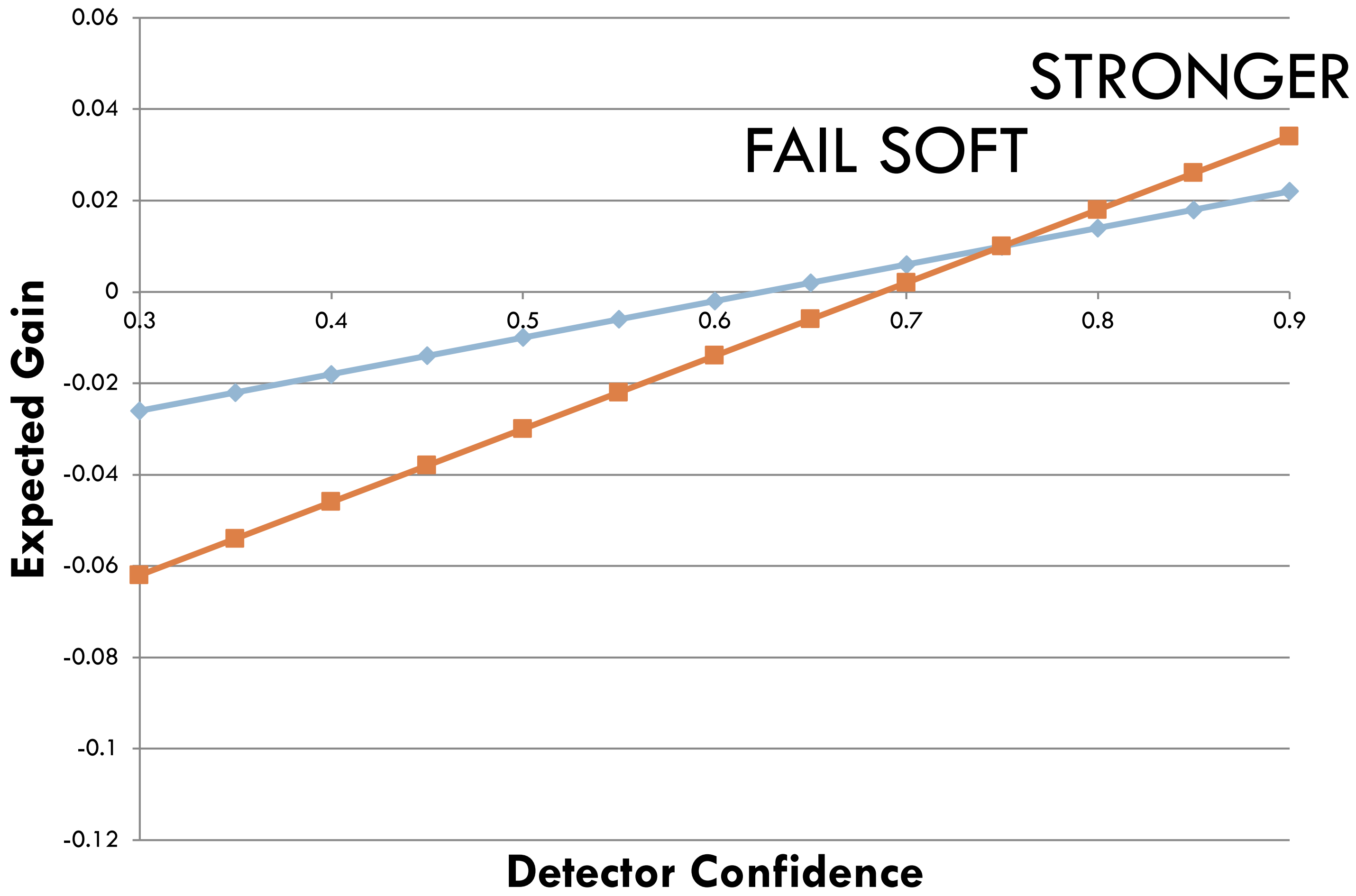


Adding a second intervention

- 0.05*Confidence – 0.08 * (1-Confidence)



Intervention cut-points



Uses of detector confidence

Uses of detector confidence

- Discovery with models analyses
  - When you use this model in further analyses
  - We’ll discuss this later in the course
  - Big idea: keep all of your information around

Not always available

- Not all classifiers provide confidence estimates

Not always available

- Not all classifiers provide confidence estimates

- Some, like step regression, provide pseudo-confidences
  - do not scale nicely from 0 to 1
  - but still show relative strength that can be used in comparing two predictions to each other

Some algorithms give it to you in straightforward fashion

- “Confidence = 72%”

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

C = Y / (Y+N)

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

C = 2 / (2+1)

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

C = 66.6667%

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

C = 100%

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

C = 2.22%

With others, you need to parse it out of software output

Tree

```
a > 1.174: Y {N=0, Y=47}
a ≤ 1.174
|   d > 1.491: Y {N=0, Y=2}
|   d ≤ 1.491
|   |   d > 1.431: Y {N=1, Y=2}
|   |   d ≤ 1.431
|   |   |   day > 8.500: Y {N=1, Y=1}
|   |   |   day ≤ 8.500: N {N=44, Y=1}
```

C = 2.22% (or NO with 97.88%)

Confidence can be "lumpy"

- Previous tree only had values
  - 100%, 66.67%, 50%, 2.22%
- This isn't a problem per-se
  - But some implementations of standard metrics (like A') can behave oddly in this case
  - We'll discuss this later this week
- Common in tree and rule based classifiers

Confidence

- Almost always a good idea to use it when it’s available
- Not all metrics use it, we’ll discuss this later this week

Risk Ratio

- A good way of analyzing the impact of specific predictors on your prediction
- Not available through all tools

Risk Ratio

- Used with binary predictors
- Take predictor P

$$RR = \frac{Probability\ when\ P = 1}{Probability\ when\ P = 0}$$

Risk Ratio: Example

- Students who get into 3 or more fights in school have a 20% chance of dropping out

- Students who do not get into 3 or more fights in school have a 5% chance of dropping out

$$RR = \frac{Probability\ when\ 3Fights{=}1}{Probability\ when\ 3Fights{=}0} = \frac{0.2}{0.05} = 4$$

- The Risk Ratio for 3+ fights is 4
- You are 4 times more likely to drop out if you get into 3 or more fights in school

Risk Ratio: Notes

- You can turn numerical predictors into binary predictors with a threshold
  - Like our last example!

- Clear way to communicate the effects of a variable on your predicted outcome

Thanks!