Lecture 8
Multiclass/Log-linear models, Evaluation, and Human Labels

Intro to NLP, CS585, Fall 2014

http://people.cs.umass.edu/~brenocon/inlp2014/

Brendan O'Connor (http://brenocon.com)

Today

- Multiclass logistic regression
- Evaluation. Humans!
- Next PS: out next week

Multiclass?

- *Task:* Classify into one of **|*Y*|** multiple exclusive categories
  - e.g. ***Y*** = {sports, travel, politics}
  - Language models (word prediction)? ***Y*** = vocabulary
- One option: transform multiple binary classifiers into single multiclass classifier
  - |***Y***| one-versus-rest classifiers.
    Hard prediction rule: choose most confident.
- But what about probabilistic prediction?
  - Does the above procedure sum-to-1 ?

Binary vs Multiclass logreg

- Binary logreg: let *x* be a feature vector, and *y* either 0 or 1

  β is a weight vector across the *x* features.

$$p(y = 1|x, \beta) = \frac{\exp(\beta^\mathsf{T} x)}{1 + \exp(\beta^\mathsf{T} x)}$$

- Multiclass logreg: *y* is a categorical variable, attains one of several values in *Y*

  Each $\beta_{y'}$ is a weight vector across all *x* features.

$$p(y|x, \beta) = \frac{\exp(\beta_y^\mathsf{T} x)}{\sum_{y' \in \mathcal{Y}} \exp(\beta_{y'}^\mathsf{T} x)}$$

Log-linear models

Here's the NLP-style notation

$f(x, y)$ *feature function* of input ***x*** and output ***y***
Produces very long feature vector.

$f_{105}(x,y) = \begin{cases} 1 \text{ if "awesome" in } \boldsymbol{x} \text{ and } \boldsymbol{y}\text{=POSITIVE} \\ 0 \text{ otherwise} \end{cases}$

$f_{106}(x,y) = \begin{cases} 1 \text{ if "awesome" in } \boldsymbol{x} \text{ and } \boldsymbol{y}\text{=NEGATIVE} \\ 0 \text{ otherwise} \end{cases}$

$f_{533}(x,y) = \begin{cases} \textit{count}(\text{"awesome" in } \boldsymbol{x}) \text{ if } \boldsymbol{y}\text{=POSITIVE} \\ 0 \text{ if } \boldsymbol{y} \text{ != POSITIVE} \end{cases}$

In this view, it's the feature engineer's job when implementing f(x,y) to include features for all combinations of aspects of inputs x and outputs y.

θ One long single weight vector.

$$p(y|x, \theta) = \frac{\exp(\theta^\mathsf{T} f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T} f(x, y'))}$$

Unnormalized, positive "exp-goodness score"

Normalizer: sum the exp-goodnesses over all possible outcomes.

Softmax function

Define: "goodness score" for potential output y.

$$s_y = \theta^\mathsf{T} f(x, y)$$

Log-linear distribution then is

$$p(y|x,\theta) = \frac{\exp(s_y)}{\sum_{y' \in \mathcal{Y}} \exp(s_{y'})}$$

Softmax function: turns goodness scores into probabilities that sum to 1. Exponentiate then normalize.

$$\mathrm{softmax}(\{s_1 ... s_{|\mathcal{Y}|}\}) \rightarrow \left(\frac{\exp(s_1)}{\sum_{y' \in \mathcal{Y}} \exp(s_{y'})}, \frac{\exp(s_2)}{\sum_{y' \in \mathcal{Y}} \exp(s_{y'})}, \ldots, \frac{\exp(s_{|\mathcal{Y}|})}{\sum_{y' \in \mathcal{Y}} \exp(s_{y'})} \right)$$

```
def softmax(scores):
    exponentiated = [exp(s) for s in scores]
    Z = sum(exponentiated)
    return [escore/Z for escore in exponentiated]
```

In-class demo
[html] [ipynb]

“Log-linear” ?

$$p(y) = \frac{\exp(\theta^\mathsf{T} \mathbf{f}(y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T} \mathbf{f}(y'))}$$

$$\log p(y) = \theta^\mathsf{T} \mathbf{f}(y) - \log \sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T} \mathbf{f}(y))$$

The
Log prob is...

$$p(y) \propto \exp(\theta^\mathsf{T} \mathbf{f}(y))$$

“Proportional to” notation, since denominator is invariant to **y**

$$\log p(y) \propto \theta^\mathsf{T} \mathbf{f}(y)$$

Abusive “log proportional to” notation... somewhat common. Sometimes convenient.

"Log-linear" ?

$$p(y) = \frac{\exp(\theta^\mathsf{T}\mathbf{f}(y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T}\mathbf{f}(y'))}$$

$$\log p(y) = \theta^\mathsf{T}\mathbf{f}(y) - \log \sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T}\mathbf{f}(y))$$

The
Log prob is...

Linear in the weights and features...

$$p(y) \propto \exp(\theta^\mathsf{T}\mathbf{f}(y))$$

"Proportional to" notation, since denominator is invariant to **y**

$$\log p(y) \propto \theta^\mathsf{T}\mathbf{f}(y)$$

Abusive "log proportional to" notation... somewhat common. Sometimes convenient.

"Log-linear" ?

$$p(y) = \frac{\exp(\theta^\mathsf{T}\mathbf{f}(y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T}\mathbf{f}(y'))}$$

$$\log p(y) = \theta^\mathsf{T}\mathbf{f}(y) - \log \sum_{y' \in \mathcal{Y}} \exp(\theta^\mathsf{T}\mathbf{f}(y))$$

The
Log prob is...

Linear in the weights and features...

Um, but except for this.

<u>Log-sum-exp</u> is an important function for these models

$$p(y) \propto \exp(\theta^\mathsf{T}\mathbf{f}(y))$$

"Proportional to" notation, since denominator is invariant to **y**

$$\log p(y) \propto \theta^\mathsf{T}\mathbf{f}(y)$$

Abusive "log proportional to" notation... somewhat common. Sometimes convenient.

Log-linear gradients

- Similar as before: difference between the gold label, versus the model's predicted probability for that label.

Log-linear models

- Such a great idea it has been reinvented and renamed in many different fields

Multinomial logistic regression is also known as **polytomous**, **polychotomous**, or **multi-class logistic regression**, or just **multilogit regression**.

Maximum Entropy Classifier
Logistic regression estimation obeys the maximum entropy principle, and thus logistic regression is sometimes called "**maximum entropy modeling**", and the resulting classifier the "**maximum entropy classifier**".

Common in 1990's-era NLP literature. Still sometimes used.

Neural Network: Classification with a Single Neuron
Binary logistic regression is equivalent to a one-layer, single-output neural network with a logistic activation function trained under log loss. This is sometimes called classification with a single neuron.

Currently popular again

Generalized Linear Model and **Softmax Regression**
Logistic regression is a generalized linear model with the logit link function. The logistic link function is sometimes called softmax and given its use of exponentiation to convert linear predictors to probabilities, it is sometimes called an **exponential model**.

http://alias-i.com/lingpipe/demos/tutorial/logistic-regression/read-me.html

Evaluation

- Given we have gold-standard labels, how good is a classifier? Evaluate on a test set (or dev set).
- Evaluating probabilistic predictions
  - Log-likelihood: we were doing this for LM's.
- Evaluating hard predictions
  - Accuracy: fraction of predictions that are correct.
    - What about class imbalance? Note also "most frequent class" baseline.
  - Many different statistics available from the ***confusion matrix***
    - Precision, recall, F-score
    - Expected utility...

Confusion matrix

	pred 1	pred 0
gold 1	True Pos	**False Neg**
gold 0	**False Pos**	True Neg

	Predicted Spam	Predicted Non-Spam
Actual Spam	5000	100
Actual Non-Spam	7	400

http://brenocon.com/confusion_matrix_diagrams.pdf

Confusion matrix

	pred 1	pred 0
gold 1	True Pos	**False Neg**
gold 0	**False Pos**	True Neg

	Predicted Spam	Predicted Non-Spam
Actual Spam	5000	100
Actual Non-Spam	7	400

	1	0
1		**FN**

p(correct | gold=1)
Sensitivity a.k.a. **Recall**
a.k.a. TruePosRate
= TP : **FN**
= 1 - FalseNegRate
= 1 - (**FN**:TP)

= 5000 / (5100)

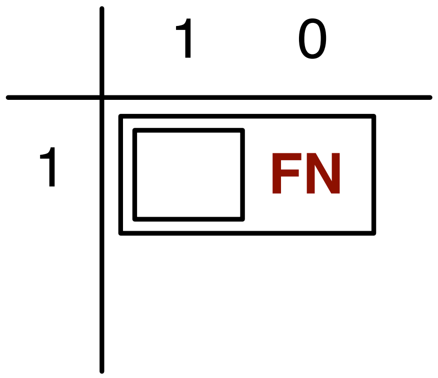
http://brenocon.com/confusion_matrix_diagrams.pdf

Confusion matrix

	Predicted Spam	Predicted Non-Spam
Actual Spam	5000	100
Actual Non-Spam	7	400

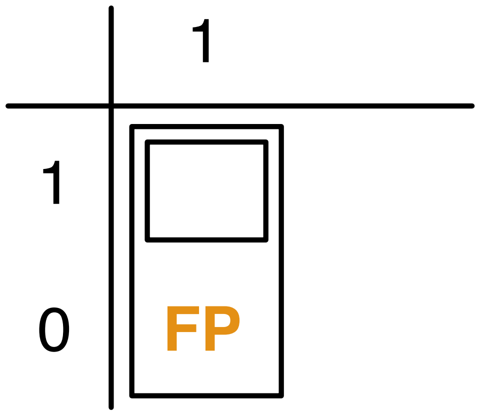
pred

gold	1	0
1	True Pos	**False Neg**
0	**False Pos**	True Neg



p(correct | gold=1)
Sensitivity a.k.a. **Recall**
a.k.a. TruePosRate
= TP : **FN**
= 1 - FalseNegRate
= 1 - (**FN**:TP)

= 5000 / (5100)



p(correct | pred=1)
Precision
a.k.a. Pos Predictive Value
= TP : **FP**
= 1 - FalseDiscoveryRate
= 1 - (**FP**:TP)

= 5000 / 5007

http://brenocon.com/confusion_matrix_diagrams.pdf

Confusion matrix

	pred 1	pred 0
gold 1	True Pos	**False Neg**
gold 0	**False Pos**	True Neg

	Predicted Spam	Predicted Non-Spam
Actual Spam	5000	100
Actual Non-Spam	7	400

	1	0
1		**FN**

p(correct | gold=1)
Sensitivity a.k.a. **Recall**
a.k.a. TruePosRate
= TP : **FN**
= 1 - FalseNegRate
= 1 - (**FN**:TP)

= 5000 / (5100)

	1
1	
0	**FP**

p(correct | pred=1)
Precision
a.k.a. Pos Predictive Value
= TP : **FP**
= 1 - FalseDiscoveryRate
= 1 - (**FP**:TP)

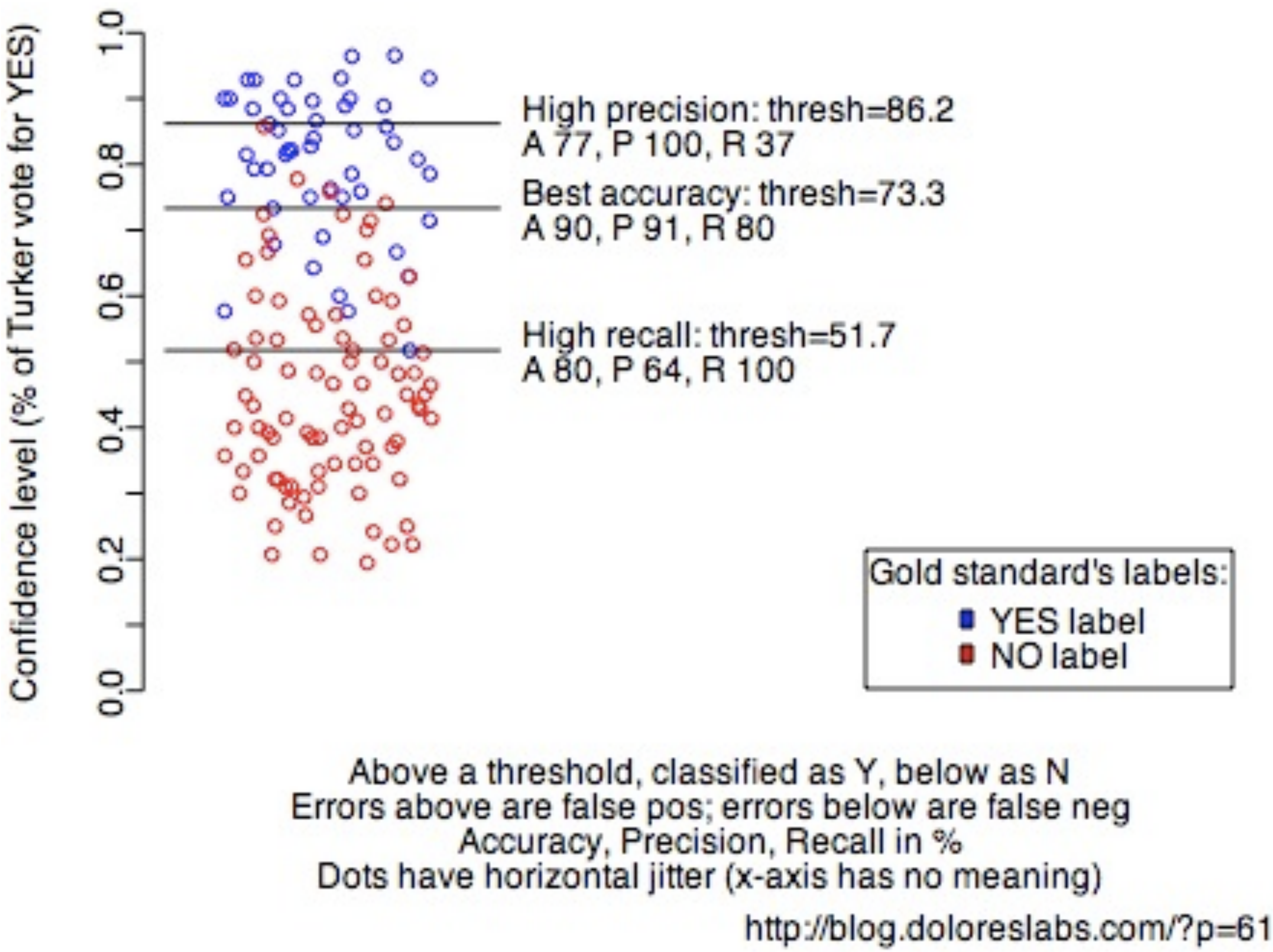
= 5000 / 5007

- Precision and Recall are metrics for binary classification.
- You can make one arbitrarily high by changing your decision threshold.
- F-score: harmonic mean of P and R.

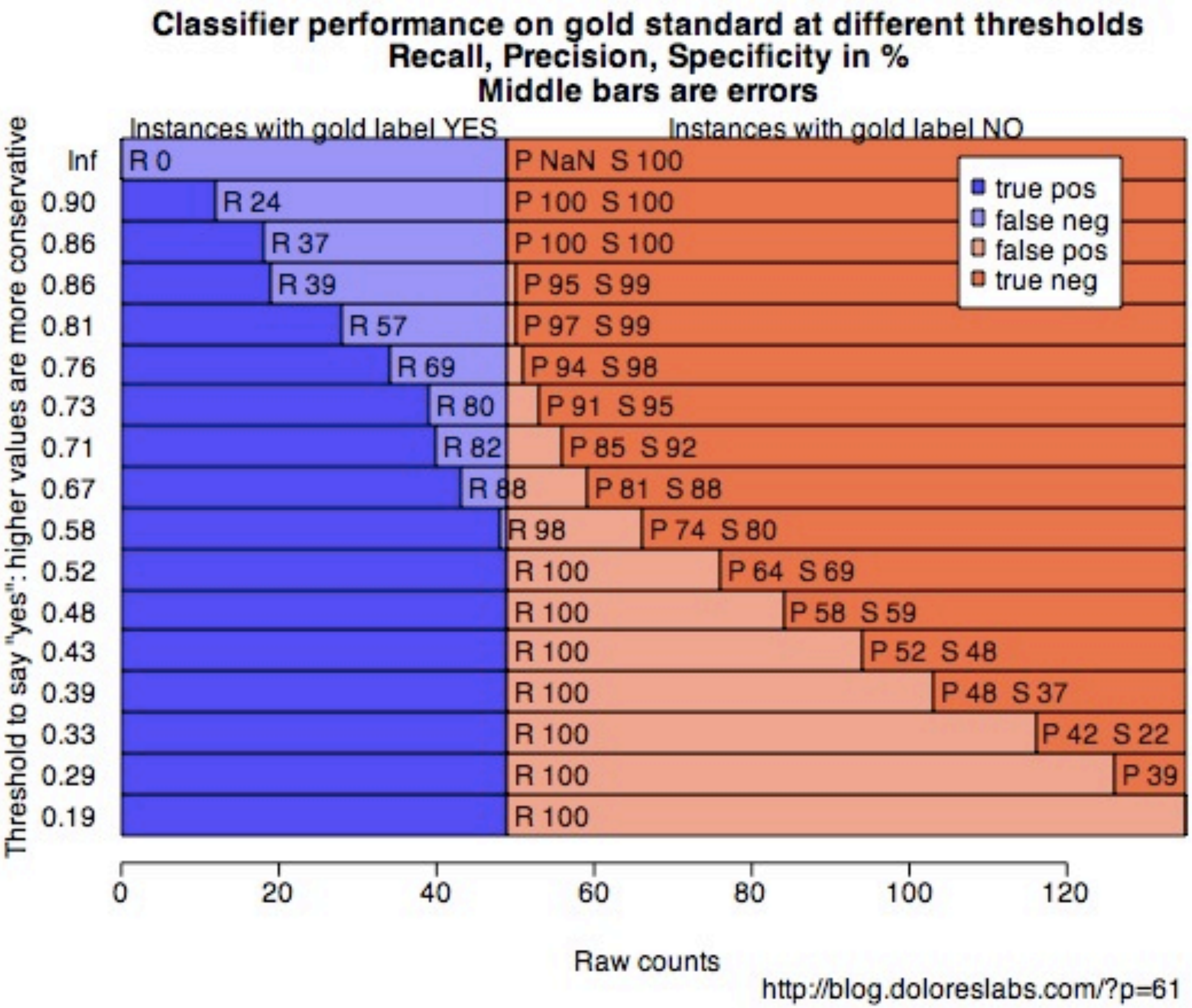
http://brenocon.com/confusion_matrix_diagrams.pdf

Decision threshold

Decide "1" if $p(y=1|x) > t$ .... could vary threshold $\boldsymbol{t}$



Decision threshold



Many other metrics

- *Expected utility*: Perhaps there's -5 points of user happiness for reading a spam message, but -1000 points for false positive spam.
- Metrics that are invariant to a decision threshold
  - Log-likelihood
  - "ROC AUC": rank by confidence value. Choose gold-positive and gold-negative examples. Are they ranked correctly?
    "area under the receiving-operator-characteristic curve"
  - Many other related things with different names from different disciplines (medical, engineering, statistics...)

http://brenocon.com/confusion_matrix_diagrams.pdf

Human annotations

- We usually think data from humans is "gold standard"
- But our tasks are subjective! What does this mean?

- Compare answers to your neighbor. How many did you agree on, out of 10?

	num HAPPY	num SAD	fraction HAPPY	Truth (?)
@AppleEl @melissaclse Oh ok! GO! - Lol you guys, I wanna join you... damn homework _EMOTICON_	0	45	0.000	>:(
My phone is so bad... _EMOTICON_	0	46	0.000	:(
Swim is on pause. It's raining. _EMOTICON_	2	44	0.043	:(
wat does that text mean.? _EMOTICON_	4	40	0.091	:)
@Moonchild66 ouch, i empathise, i get that a bit from sleeping awkwardly! _EMOTICON_	7	39	0.152	:(
surreal knowing middle school is officially over _EMOTICON_	19	26	0.422	:(
If dnt nobody else love me i love me _EMOTICON_	41	5	0.891	:)
Fireflies - Owl City #nowplaying _EMOTICON_	45	1	0.978	=)
Oh, is anyone watching Dancing with the Stars? That old lady is made of win _EMOTICON_	45	1	0.978	^_^
U see the name _EMOTICON_ http://t.co/Bns4wQyF	45	1	0.978	:)

SAD	@AppleEl @melissaclse Oh ok! GO! - Lol you guys, I wanna join you... damn homework _EMOTICON_	@AppleEl @melissaclse Oh ok! GO! - Lol you guys, I wanna join you... damn homework >:(
SAD	surreal knowing middle school is officially over _EMOTICON_	surreal knowing middle school is officially over :[
SAD	@Moonchild66 ouch, i empathise, i get that a bit from sleeping awkwardly! _EMOTICON_	@Moonchild66 ouch, i empathise, i get that a bit from sleeping awkwardly! :(
HAPPY	If dnt nobody else love me i love me _EMOTICON_	If dnt nobody else love me i love me :)
SAD	My phone is so bad... _EMOTICON_	My phone is so bad... :(
HAPPY	Oh, is anyone watching Dancing with the Stars? That old lady is made of win _EMOTICON_	Oh, is anyone watching Dancing with the Stars? That old lady is made of win ^_^
SAD	Swim is on pause. It's raining. _EMOTICON_	Swim is on pause. It's raining. :-(
HAPPY	U see the name _EMOTICON_ http://t.co/Bns4wQyF	U see the name :) http://t.co/Bns4wQyF
HAPPY	Fireflies - Owl City #nowplaying _EMOTICON_	Fireflies - Owl City #nowplaying =)
HAPPY	wat does that text mean.? _EMOTICON_	wat does that text mean.? :)

Human agreement

- Should we expect machines to do subjective tasks?
  - Is the task "real", or is it a fake made-up thing? What does "real" mean anyways? Is sentiment a "real" thing?
- Inter-annotator agreement rate (IAA) is the standard way to measure "realness" and quality of human annotations.
  - Have two annotators annotate the same item.
  - Fraction of the time they agree.
  - *Alternate view:* accuracy rate that one has when trying to model the other.
  - Cohen's *kappa*: a variation on IAA that controls for base rates (compare against null case of everyone answering the most common answer).
- Human factors affect agreement rates!!
  - Are the annotators trained similarly?
  - Are the guidelines clear?
  - Is your labeling theory of sentiment/semantics/etc "real"?
- Common wisdom: IAA is upper bound on machine accuracy. Really? Discuss.