Known Unknowns in Discharge Summary Mining

Bob Carpenter, Breck Baldwin

Alias-i, Inc.

Carlos Cano, Leon Peshkin

Harvard University

i2b2 Obesity Challenge

- Obesity information and fifteen co-morbidities have been marked at a document level as: present (Y), absent (N), questionable (Q), or unmentioned (U)
- Two tracks:
  - textual judgments, i.e., what the text explicitly states
  - intuitive judgments, i.e., what the text implies [no U category]
- Evaluate systems on their ability to recognize whether a patient is obese and what co-morbidities they exhibit.

https://www.i2b2.org/NLP/

Known Unknowns, Negatives and Reporting

- Questionable (Q) annotations indicated textual or implied evidence that disease status was unknown (that is, known unknowns).
- Negative (N) annotations in textual judgment means an explicit recorded negative diagnosis
-

Type	Y	N	Q	U	Total
Textual	3208	87	39	8296	11,630
Intuitive	3267	7362	26	n/a	10,655

- The records aren't being annotated for uncertainty or negative diagnoses

Corpus Adjudication and Agreement

- Anonymized discharge summaries from Partners HealthCare for patients evaluated for obesity or diabetes
- Annotated by Massachusetts General Hospital Weight Center
- Two obesity experts independently coded everything
- One doctor broke ties on textual judgments
- Intuitive data censored to agreed cases
- Kappa Scores (P-E)/(1-E)
  - Textual Kappas: .71–.92
  - Intuitive Kappas: .44–.92
- Training/Test balanced for prevalence of obesity (not co-morbidities)

Evaluation: Macro-Averaged F-Measure

- Precision, recall and F-measure for all tasks
- Ranked by macro-averaged F-measure
  - F-measure per morbidity n and type $t \in \{Y, N, Q, U\}$:
    $F_{n,t} = 2P_{n,t}R_{n,t}/(P_{n,t} + R_{n,t})$
  - Macro-average per morbidity
    $F_n = (F_{n,Y} + F_{n,N} + F_{n,Q} + F_{n,U})/4$
  - Final average for evaluation
    $F = (F_1 + \cdots F_{16})/16$
- Low count Q (textual/intuitive) and N (textual) cases heavily weighted

Classification Problem

- One classifier per disease
  - Textual: Run 1: Y/U, Runs 2–3: Y/U/N
  - Intuitive: Runs 1–3: Y/N
- Ignored Q cases by misunderstanding task

Multinomial Logistic Regression

- Multiple category linear basis classifier (coeffs $\beta \in \mathbb{R}^n$, data $x \in \mathbb{R}^n$)

$$p(c|x) \propto \text{logit}^{-1}(\beta_c x^\top) = \frac{1}{1 + \exp(-\beta_c x^\top)}$$

- Laplace (double-exponential) priors (aka lasso, L_1 regularization)

$$\log p(\beta) \propto \sum_{1 \leq k \leq n} \frac{|\beta_k|}{\sigma} = \frac{|\beta|_1}{\sigma}$$

  favors zeros but otherwise broader than Gaussian priors

- Maximum a posteriori (MAP) estimates
- Optimizes micro-averaged log loss:

$$\text{Err}(\beta) = \log p(\beta) + \sum_i \log p(c_i|x_i)$$

- Mismatch with 0/1-loss macro-averaged F-measure

LingPipe's Logistic Regression

- Algorithm: Sparse, Regularized Stochastic Gradient Descent (SGD)
- Online processing; multiple epochs for small data (10K epochs here)
- Memory: only the coefficients and item being estimated
- Data vectors may be sparse (here, they're very sparse)
- General feature extraction interface
- Built-in implementations of text features (e.g. tokenizers, stemmers, normalizers)
- Pluggable priors for regularization (Laplace, Cauchy, Gaussian, or user-defined)
- Fits into LingPipe's classifier evaluation and runtime interfaces
- Data very close to being linearly separable (near 0 log likelihood)

Run 1: N-gram Features

- Character 5-grams
  - Input: Anemia and GI bleed.
  - N-gram Vector: "Anemi":1 "nemia":1, "emia ":1, "mia a":1, "ia an":1, "a and":1, " and ":1, "and G":1, "nd GI":1, "d GI ":1, "GI b":1, "GI bl":1, "I ble":1, " blee":1, "bleed":1, and "leed.":1.
- Limited cross-word effects
- Longer n-grams or mixed n-grams didn't help
- Pruned counts less than 20, resulting in about 20,000 features/task
- No character normalization (case, punctuation, etc.)
- Laplace prior variance of 0.1 forces many features to zero

Run 2: Negation and Drug Names

- Negative particles distributed over their sentences as additional features

```
IN: no alcohol, tobacco or drug use
ADD: NO_alcohol, NO_tobacco, NO_or, NO_drug, NO_use

IN: No family history of kidney disease or CAD.
ADD: NO_family NO_history NO_of NO_kidney
     NO_disease NO_or NO_CAD
```

- LingPipe sentence detection
- ...

Run 2 (cont.)

- Generic drug-treatment feature for drugs used to treat conditions as additional features (used external knowledge sources to collect synonyms)

  ```
  IN: started on clonidine
  OUT: Hypertension_DRUG

  IN: Clonodine 0.6 mg topical.
  ADD: [nothing, it's a misspelling]
  ```

- Pruned to features with counts of 20 or higher
- Laplace prior further drives parameters to zero
- Python code to munge input data

Run 3: Feature Selection

- Just like Run 2, only selecting limited number of features
- Information Gain is reduction in entropy (H) of category decisions (C) given the feature f and data X:

$$\mathrm{IG}(f) = p(f)\mathrm{H}(C|f, X) + (1 - p(f))\mathrm{H}(C|\neg f, X)$$

$$\mathrm{H}(C|f, X) = \sum_{c \in C|f(c)} p(c|X) \log_2 p(c|X)$$

- Like an ANOVA (without mixed effects) in log likelihood space
- Select features with highest IG (greedy and independent)
- Number of features wasn't sensitive between 5 and 25
- Reliable general purpose feature selection for text classification
- Python code to munge input data

Results

Run	Task	P-Mac	R-Mac	F-Mac	Accuracy= P,R,F-Mic
1)	Textual	0.956	0.446	0.451	0.918
2)	Textual	0.567	0.484	0.506	0.923
3)	Textual	0.763	0.457	0.464	0.929
1)	Intuitive	0.928	0.585	0.589	0.902
2)	Intuitive	0.935	0.585	0.592	0.906
3)	Intuitive	0.950	0.600	0.607	0.924

Conclusions

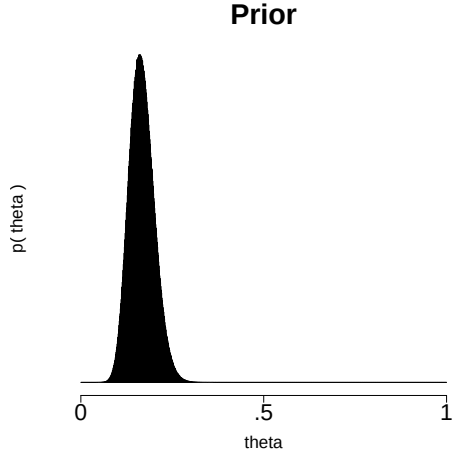
- Most cases very easy (positive obesity very highly correlated with word "obese" in text)
- Need more data on low-count categories for statistical approach
- You're only as good as your features
  - Need better approach to negation (ideally, parse for scope)
  - Need better normalization due to case and misspellings
  - Use structure of document for hierarchical modeling or feature factorization (e.g. principal diagnosis, secondary diagnoses, history of present illness, pre-admission meds, past history, family history, social history, discharge, hospital course, allergies, admission physical exam, studies, history, etc.)
- Exploit correlation among output categories

Postscript: Bayesian Diagnosis

- $\theta \in [0, 1]$: probability of disease
- $p(\theta)$: prior prevalence in population
- X: observed data (e.g. text, lab reports)
- $p(X|\theta)$: likelihood (generative)
- $p(\theta|X) \propto p(X|\theta)p(\theta)$: posterior (by Bayes' law)

Prior Distribution

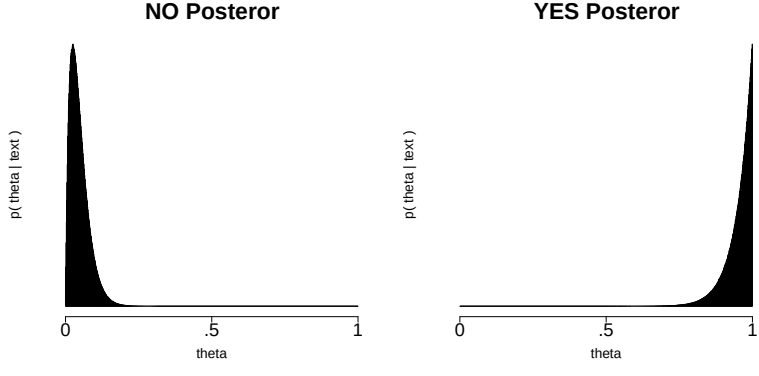
$p(\theta)$ is an estimate of the population distribution:



- Estimate reflects uncertainty in population distribution

Conclusive Posteriors: Y and N

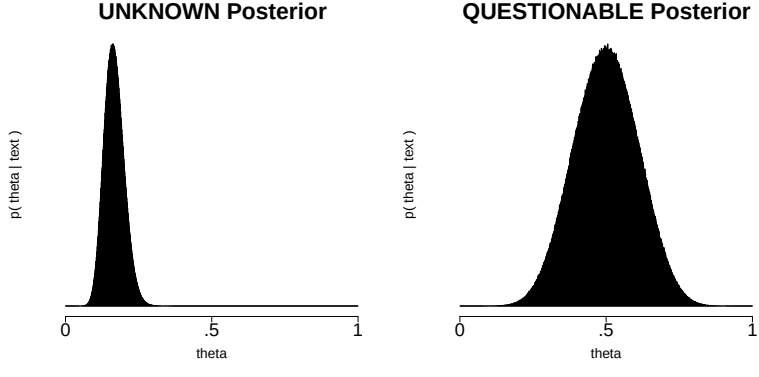
Posterior is definite in diagnosis



- Estimates reflect posterior uncertainty given data
- Easy to quantize into discrete categories U and Q

Inconclusive Posteriors: Q and U

Posterior looks like prior for unknowns, more central for borderline/questionable



- More difficult to quantize into discrete categories Y and N
- Still easy to use as a Bayesian component for inference

Thanks To

- Ozlem Uzuner for organizing everything
- Harvard's i2b2 for hosting
- NIH for funding

The project described was supported by Grant Number R44 RR020259 from the National Center For Research Resources. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Center for Research Resources or the National Institutes of Health.