Statistical Language Models
2019
Week 4 part 2

Dr. Dave Campbell
davecampbell@math.carleton.ca

Approximate Course Outline

- Week 1: ShinyApps and Dashboarding
- Week 2: TidyText & obtaining data, dealing with time events
- Week 3: Regular Expressions; Word co-occurrence explorations
- Week 4: Sentiment Analysis; Stochastic process models
- Week 5: Exponential models for time between events.
- Week 6: Bayesian Basics; Author attribution models; hierarchical models
- Week 7: MCMC Diagnostics
- Week 8: Embeddings and Word2Vec; Cryptography
- Week 9: Clustering; Latent Dirichlet Allocation and topic models.
- Week 10: Variational Inference
- Week 11: Getting Fancier with Language Models
- Week 12: Student projects and presentations

- # decode and log probability functions comes from a similar code from similar problem:
- http://alandgraf.blogspot.ca/2013/01/text-decryption-using-mcmc.html?m=1
- # also provides a decoding function (decode) based on the substitution
- # rule and a likelihood function sum(log.prob) based on the stochastic
- # process. I've modified their log.prob function to deal with the
- # dyad structure.

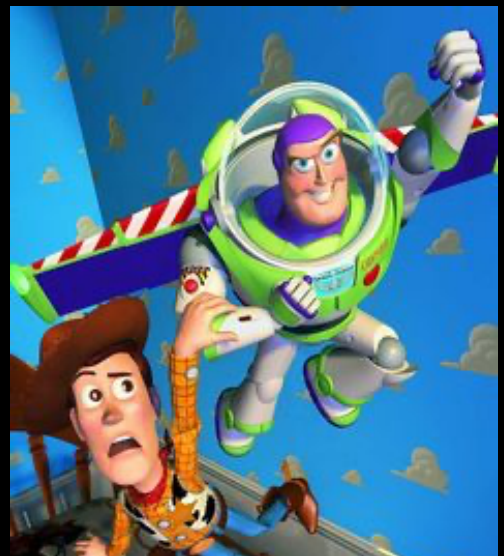
https://ssc.ca/en/meetings/annual/2020-annual-meeting/case-studies

- SSC case study competitions!
- Impress your friends and judges!
- Solve a real data challenge!
- Travel to Carleton University to show off your work as a poster
- Add a line to your resume and something to talk about in job interviews!
- Win prizes!

Stochastic processes

- In large texts think about the limiting distribution of state i:
  $\pi_i = \lim_{n \to \infty} P(X(t+n) = i)$

- Calculate via the Buzz Lightyear strategy



- Solve the system of equations

$$\pi_j = \sum_i \pi_i P_{ij}$$

$$\sum_j \pi_i = 1$$

In statistics we:
1 make a model like $X \sim Bin(n, \theta)$ ($\Rightarrow$ the likelihood)

2 do an experiment/ collect data, i.e. collect X from n trials

3 update our model and/or test our beliefs. estimate θ and quantify it's uncertainty with an interval or test $Ho{:}\theta \leq .5$ etc...

4 repeat as necessary

5 predict the future/ save the world

The usual (frequentist) idea is that our 95% confidence interval will on contain the true θ 95% of the times if the study is repeated many many times.

Bayesian statistics estimates θ using all available knowledge.

1 make a model like $X \sim Bin(n, \theta)$ ($\Rightarrow$ the likelihood)and quantify what we know about θ with $P(\theta)$, the Prior distribution on θ.

2 do an experiment/ collect data, i.e. collect x from n trials

3 update our model and/or test our beliefs. estimate $P(\theta \mid x)$

4 repeat as necessary

5 predict the future/ save the world

This gives the distribution $P(\theta \mid x)$ outlining our belief in the value of θ and provides Bayesian credible intervals

WANT TO TALK TO US ? Foreign editor: Raymond Beauchemin 514 987-2457 rbeauchemin@thegazette.canwest.com

Merck vaccine blocks cervical cancer

Final-stage study proves 100% effective

Breakthrough could lift sagging company, which has been hit hard by Vioxx withdrawal

LINDA A. JOHNSON
ASSOCIATED PRESS

TRENTON, N.J. – The first large study of an experimental cervical cancer vaccine found it was 100-per-cent effective, in the short ferm, at blocking the most common cause of the disease, the vaccine's maker said yesterday.

Merck's genetically engineered vaccine prevents cervical cancer by blocking infection from the human papilloma virus strains that cause 70 per cent of cervical cancers.

Other types of HPV, which is sexually transmitted, also can cause cervical cancer and painful genital warts. About 20 million Americans have some form of HPV.

The final-stage study of the vaccine included 10,559 sexually active women ages 16 to 26 in the United States and 12 other countries who were not infected with the HPV strains 16 or 18. Half got three vaccine doses over six months; half got dummy shots.

Among those still virus-free after the six months, none who received the vaccine developed either cervical cancer or precancerous lesions likely to turn cancerous over an average two years of followup, compared with 21 who got dummy shots.

"To have 100-per-cent efficacy is something that you have very rarely," Dr. Eliav Barr, Merck's head of clinical development for the vaccine called Gardasil, told Associated Press. "We're breaking out the champagne."

Merck shares rose 56 cents, or 2.1 per cent, to $27.45 U.S.

The study, which was funded by Merck, was to be presented today at a meeting of the Infectious Diseases Society of America.

A second analysis, including hundreds more women in the study, showed that after only one dose the vaccine was 97-per-cent effective. That analysis found only one of the 5,736 women who got the vaccine developed cervical cancer or precancerous lesions, compared with 36 among the 5,766 who got dummy shots.

Barr said the 97-per-cent rate was more "real world," given that patients sometimes miss or delay follow-up shots or tests.

"I see this as a phenomenal breakthrough," said Dr. Gloria Bachmann, director of The Women's Health Institute at Robert Wood Johnson Medical School in New Brunswick.

Bachmann said diagnosis of infection leaves women anxious over the heightened risk of cervical cancer and raises questions among couples about infidelity and prior sexual activity.

"You have to get students in grammar school, middle school, high school (vaccinated) before they become sexually active," she said.

Cervical cancer is the second-most common cancer in women and their No. 2 cause of cancer deaths, resulting in about 3,000 deaths in the United States and nearly 300,000 around the world each year. At least half of sexually active men and women become infected with genital HPV at some point.

The immune system clears most such infections in a year or two, but several types of HPV can persist, cause cervical cancer or trigger other cancers in the genital area. There is no cure for HPV, but the cancers can be treated and an improved Pap test is catching more cervical cancer before it has spread.

Whitehouse Station-based Merck, hammered by slumping revenues and profits and facing about 5,000 lawsuits over its withdrawn painkiller Vioxx, is seeking to beat rival drug maker GlaxoSmithKline PLC to market with the first cervical cancer vaccine.

Merck plans by year's end to seek Food and Drug Administration approval to sell its vaccine for use by girls and young women.

"If all goes well, sometime in 2006 it should be on the market," Barr said.

MERCK Vaccine

- Merck does a study to examine if it's new vaccine works. The Beta distribution:

- $f(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}, \quad 0 \leq \theta \leq 1, \;\; \alpha, \beta > 0$

- $E(\theta) = \frac{\alpha}{\alpha+\beta} \quad Var(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

```
T=1000000
q=rep(0,1,T+1)
y=1
N=5736
stepvar=.004
accepts=0
rate=rep(0,1,10)
k=1
q[1]=1/5736
for(lp in 1:(T-1)){
    x=runif(1,q[lp]-stepvar,q[lp]+stepvar)
    if(x>0 & x<1){
        alpha= dbinom(y,N,x)*(x^36)*(1-x)^(5730)*(2-2*x) /
(dbinom(y,N,q[lp])*(q[lp]^36)*(1-q[lp])^(5730)*(2-2*q[lp]))
        u=runif(1)
        if(u<alpha){
            accepts=accepts+1
            q[lp+1]=x
        }
        else{
            q[lp+1]=q[lp]
        }
    }
    else{
        q[lp+1]=q[lp]
    }
}
```

Metropolis Hastings

- Start with X(t-1) = j
- Propose a value Y(t) | X(t-1)=j from transition probability matrix Q as a candidate for X(t)
- compute $\alpha_{ij} = min\left(\frac{P(Y)P_{ji}}{P(X[t]P_{ij}}, 1\right)$ and sample u~Unif(0,1)
- If u< α_{ij} then accept the proposal and set Xt=Y and if not then set X(t)=X(t-1).
- Repeat (T times) until you obtain a sufficient sample from the distribution of X

- Use the sampled values of X to compute
- $$E[h(X)] = \sum_{j=1} h(x_j) P(X = x_j)$$
$$= \sum_{j=1}^{T} \frac{h(x_j)}{T}$$
- We often use the sampled values to get an approximation for the mean, median, modes, variance, interval estimates, quantiles...
- Bayesian statistics uses MCMC to give an approximation to the full posterior distribution.

```
for(lp in 1:(T-1)){
   x=runif(1,q[lp]-stepvar,q[lp]+stepvar)
   if(x>0 & x<1){
      alpha= dbinom(y,N,x)*(x^36)*(1-x)^(5730)*(2-2*x) / (dbinom(y,N,q[lp])*(q[lp]^36)*(1-q[lp])^(5730)*(2-2*q[lp]))
      u=runif(1)
      if(u<alpha){
         accepts=accepts+1
         q[lp+1]=x
      }
      else{
         q[lp+1]=q[lp]
      }
   }
   else{
      q[lp+1]=q[lp]
   }
   if(lp%%(T/40)==0&&lp<T*.25){  # check on the acceptance rate for the first quarter of the sampled values
      rate[k]=accepts/(T/40)
      if(rate[k]>.49 | rate[k]<.39){
         stepvar=stepvar*rate[k]/.44
      }
      k=k+1
      accepts=0
   }
}
```

- hist(q)
- qbeta(.025,2,5737) [1] 4.221423e-05
  qbeta(.975,2,5737) [1] 0.0009706212
- x = seq(0,.01,length=100000)
  lines(x, dbeta(x,2,5737) , type='l') > abline(v=qbeta(c(.025,.975),2,5737))
-

- Letter Swapping Code, solve the puzzle: http://cryptogramcorner.org
- Model:
- Prior : P(cypher ϕ) = Uniform
- Cypher is swap from A—>A, A—>B, A—>C…
- Likelihood : P(decoded phrase | cypher ϕ, transition probabilities θ)
- Posterior : P(cypher ϕ | decoded phrase, transition probabilities θ)

Cryptography

- Letter Swapping Code, solve the puzzle: http://cryptogramcorner.org
- Model:
- Prior : P(cypher ϕ) = Uniform
- Cypher is swap from A—>A, A—>B, A—>C…

- We start with $X(t=0)=x$. Is this reasonable? What if it isn't?
- How do we know if T is big enough?
- Are some proposal distributions better than others?

- Stochastic process $\{X(t), t \in T\}$, consider discrete case.

  If $X(t) = a$, the stochastic process is at state a at time t

- If, whenever $X(t) = a$, for all t there is a fixed probability P_{ab} that $X(t + 1) = b$, then the stochastic process is a Markov chain

- a Markov chain is a stochastic process where for all states $a_{t-1}, \ldots, a_0$, and for all $t \geq 0$

- $P[X(t+1) = b \mid X(t) = a, X(t-1) = a_{t-1}, \ldots X(0) = a_0) = P_{ab}$

Letters

- At letter position t, state X(t) = "N"
- At letter position t+1, we x(t+1) = ?
- P_{NO} = ?
- Find this empirically by sifting through books

Books.txt = "A Tale of Two Cities", "Moby Dick", "The Adventure of Sherlock Holmes", and "Pride and Prejudice"

- reference=readLines("Books.txt",encoding="UTF-8")
- reference=toupper(reference)

Define states; (letter1,letter2) —> (letter2,letter3)

- LetterDyadSet = NULL # All possible states (pairs of letters)
- characters2use = c(toupper(letters),"")
- for(lp in 1:length(characters2use)){
- LetterDyadSet=c(LetterDyadSet,paste(characters2use[lp],characters2use))
- }
- possible.moves=matrix(0,nrow=27*27,ncol=27*27,dimnames=list(LetterDyadSet,LetterDyadSet))

```
# build a transition matrix
Twobackletter = ""
lastletter = ""
for (Twobackletter in characters2use) {
    for (lastletter in characters2use) {
        BackDyad=paste(Twobackletter,lastletter)
        for (curletter in characters2use) {
            currDyad  = paste(lastletter,curletter)
            possible.moves[dimnames(possible.moves)[[1]]==BackDyad,
                    dimnames(possible.moves)[[2]]==currDyad]= 1
        }
    }
}
trans.mat = possible.moves
```

Scan through texts to fill in occurrence data

```
Twobackletter = ""
lastletter = ""
for (ln in 1:length(reference)) {
  if (ln %% 2500 ==0) {cat("Line of text",ln,"\n") }      # let me know when I go through multiples of 2500 lines
  for (pos in 1:nchar(reference[ln])) {                   # position within a line for current letter
    curletter = substring(reference[ln],pos,pos)          # current letter
    currDyad  = paste(lastletter,curletter)               # current dyad state
    BackDyad  = paste(Twobackletter,lastletter)           # previous dyad
  if (currDyad %in% LetterDyadSet) {      #add one to observed transition
      trans.mat[dimnames(trans.mat)[[1]]==BackDyad,
                dimnames(trans.mat)[[2]]==currDyad]=
         trans.mat[dimnames(trans.mat)[[1]]==BackDyad,
                   dimnames(trans.mat)[[2]]==currDyad]+1
      Twobackletter=lastletter                            #adjust names
      lastletter=curletter                                #adjust names
    } else {  # if the current letter is a punctuation mark or space
      curletter = ""              #convert to my generic space letter
      currDyad  = paste(lastletter,curletter)
      trans.mat[dimnames(trans.mat)[[1]]==BackDyad,
                dimnames(trans.mat)[[2]]==currDyad]=
        trans.mat[dimnames(trans.mat)[[1]]==BackDyad,
                  dimnames(trans.mat)[[2]]==currDyad]+1
      Twobackletter=lastletter
      lastletter=curletter
      }
    }
}
# Now to normalize the whole thing to make sure that the row sums are equal to 1
trans.prob.mat = sweep(trans.mat,1,rowSums(trans.mat),FUN="/")
```

How I figured out if it worked

- for (ln in 1:length(reference)) {
- if (ln %% 2500 ==0) {cat("Line of text",ln,"\n") } # let me know when I go through multiples of 2500 lines
- for (pos in 1:nchar(reference[ln])) { # position within a line for current letter
- curletter = substring(reference[ln],pos,pos) # current letter
- currDyad = paste(lastletter,curletter) # current dyad state
- BackDyad = paste(Twobackletter,lastletter) # previous dyad
- #diagnostic help I used this when I was figuring out if it was working but now it is commented out
- #if (ln %% 500 ==0) {cat("Line of text",ln,"\n")
- # print(paste(BackDyad,currDyad))
- #}

Plots of transition probabilities

- par(mfrow=c(2,2))
- temp = 1:27
- names(temp) = c(characters2use[1:26],"_")
- for(lp in 1:27){
- temp[lp] = trans.prob.mat[paste("T","H"),paste("H",characters2use[lp])]
- }
- matplot(1:27, temp, type="b", pch=1, xaxt="n",main="T H")
- axis(1, 1:27, labels=names(temp))
- #
- for(lp in 1:27){
- temp[lp] = trans.prob.mat[paste("A","N"),paste("N",characters2use[lp])]
- }
- matplot(1:27, temp, type="b", pch=1, xaxt="n",main="A N")
- axis(1, 1:27, labels=names(temp))

- for(lp in 1:27){
- temp[lp] = trans.prob.mat[paste("R","E"),paste("E",characters2use[lp])]
- }
- matplot(1:27, temp, type="b", pch=1, xaxt="n",main="R E")
- axis(1, 1:27, labels=names(temp))
- ###

- for(lp in 1:27){
- temp[lp] = trans.prob.mat[paste("",""),paste("",characters2use[lp])]
- }
- matplot(1:27, temp, type="b", pch=1, xaxt="n",main="two blanks")
- axis(1, 1:27, labels=names(temp))

```
decode = function(mapping,coded) {
  coded=toupper(coded)
  decoded=coded
  for (i in 1:nchar(coded)) {
    if (substring(coded,i,i) %in% toupper(letters)) {
      substring(decoded,i,i)=toupper(letters[mapping==substring(coded,i,i)])
    }
  }
  decoded
}


log.prob = function(mapping,decoded) {
  logprob=0
  Twobackletter = ""
  lastletter = ""

  for (i in 1:nchar(decoded)) {
   curletter  = substring(decoded,i,i)
   if(curletter == " "){curletter=""}
    currDyad   = paste(lastletter,curletter)
    BackDyad   = paste(Twobackletter,lastletter)
   if (currDyad %in% LetterDyadSet) {
    logprob=c(logprob,log(trans.prob.mat[rownames(trans.mat)==BackDyad,
                                         colnames(trans.mat)==currDyad]))
    Twobackletter=lastletter
    lastletter=curletter
   } else {
    #print(paste(BackDyad,currDyad))
    curletter =""
    currDyad  = paste(lastletter,curletter)
```