

Probabilistic Scenario-Based Design: Generating insight from probability-based theoretical user models

Contents

0.1 Probabilistic Scenario-Based Design: Generating insight from probability-based theoretical user models	1
1 Abstract	2
2 Introduction	2
3 Previous Work	2
3.1 Intelligent User Interfaces:	2
3.2 Scenario Based Design	3
3.3 Bayesian Inference	4
3.4 Therefore,	4
4 Limitations	4
4.1 Bayesian Assimilation in Interface Design	4
4.2 Case Study Introduction	4
5 Method	4
6 Case Study	6
6.1 Use Cases	6
6.2 Wireframes	7
6.3 Initial Proposals for Distributions	7
6.4 Negotiation of Initial Distributions	8
6.5 Visually Assess the Data	8
6.6 Principle Components Analysis	10
6.7 Cluster Analysis	12
6.8 Bayesian Data Assimilation	14
7 Discussion	14
7.1 Statistical Testing of Usage Differences	14
7.2 Novel Visualization Potential	14
7.3 Bayesian Assimilation with Real Usage Patterns	14
8 Conclusion	16
9 References	16

0.1 Probabilistic Scenario-Based Design: Generating insight from probability-based theoretical user models

0.1.0.1 Scott Farley

0.1.0.2 April 2017

1 Abstract

Constructing hypothetical scenarios and user narratives is a common technique for communicating the envisioned user experience (UX) of a tool. Using this approach, application developers rapidly build stories of expected use cases that concretize the intended UX. This design strategy is effective, cheap, flexible, and simple to implement. However, most narrative scenarios are not formal, rather they lack a mathematical basis suitable for statistical testing and rigorous delineation of usership patterns. In this paper, I describe an effort for underpinning use case scenarios (UCSs) with a probability framework. Under this framework, UCSs are modified to include probability distributions describing the use of each interface component. This approach offers at least three advantages that complement traditional scenario-based design approaches: (1) it enables formal statistical testing of use patterns, with both hypothetical data and, once the tool is put into production, real usage data, (2) it creates the potential for novel visualizations of application use, and (3) it creates the possibility of incorporating Bayesian data assimilation into interface design decisions, offering insight for future designers. Here, I discuss the application of this method to an in-development software tool, Ice Age Mapper, a tool for exploring multivariate, spatiotemporal ecological change in recent geologic history. Several UCSs are explored and fuzzy statistical methods are shown to effectively delineate between them when configurations are drawn at random from the newly implemented probability distributions.

2 Introduction

Front-end code and design frameworks, widespread access to low-cost, high-speed interaction connection, and ubiquitous mobile computing have resulted in richly featured, highly interactive applications distributed through the internet as web apps. So-called Web 2.0 technologies have made these applications effective platforms for user-generated content, socially enabled to engage users. Recent growth in cloud services have caused many established companies to push services that traditionally appear in a desktop environment into a cloud-enabled software-as-a-service (SaaS). SaaS makes user profile information available synchronously across multiple devices.

Recent trends in cloud

Recommender systems – music, movies, home automation

intelligent tutoring systems – student model controls how a user is progressing towards an educational goal
in general, user modeling adapts interaction to improve user specific session based on user needs

AI-based interfaces can improve UI – expose the user to less choice

Better decision making Most AI systems are designed for plan recognition

AI needs formal logic

Predictive statistical models of user behavior exist for ecommerce

3 Previous Work

3.1 Intelligent User Interfaces:

Harnessing AI to improve user interfaces. Giving the user what they want when they want it

p.171: We can project a not-so-distant future where “intelligent science assistant” programs identify and summarize relevant research described across the worldwide multilingual spectrum of blogs, preprint archives, and discussion forums; find or generate new hypotheses that might confirm or conflict with ongoing work; and even rerun old analyses when a new computational method becomes available. Aided by such a system,

the scientist will focus on more of the creative aspects of research, with a larger fraction of the routine work left to the artificially intelligent assistant. – Highlighted May 2, 2017

p.172: Eureka, usable in many scientific fields, searches a vast space of hypotheses consistent with given data observed in an experiment, selects those most promising, and designs experiments to test them (13). – Highlighted May 2, 2017

Two categories Adaptive user interfaces Recognition of user based on series of events, what to do next? p.1: The promise of adaptive user interfaces is that they will provide a flexible mechanism for systems to adapt to the needs of different users for a variety of tasks. – Highlighted May 2, 2017 Recommendation systems Recognition of user based on purchased items, what to purchase next?

AI for design (?) p.1: In particular, we can apply the available computational power to determine how and when to proactively assist users. – Highlighted May 2, 2017

p.378: The IM's output is realized by diverse User Interface components, which are able to use different widgets to realize the so-called "Final User Interface" (FUI). – Highlighted May 2, 2017

p.379: Contrary to that, a novice user would receive a more detailed and longer sequence of dialog steps with additional extent of assistance. – Highlighted May 2, 2017

p.380: checks whether the user has the required knowledge. – Highlighted May 2, 2017

p.380: It determines an adequate combination of diverse user interface components, adapted to the identified context of use. – Highlighted May 2, 2017

p.380: Each addressed user interface component renders an individually assigned part of the output as final user interface (see Fig. 2). – Highlighted May 2, 2017

p.172: Sunfall incorporates usability principles and cognitive load considerations in the design of a visual analytics interface; – Highlighted May 2, 2017

Multilevel interfaces – complex systems could benefit from AI p.1: Mixed-initiative interaction strategy in which each agent (human or computer) communicates what it is best suited to do. In most appropriate situations. – Highlighted May 2, 2017

Systems must have formally represented information on which to make logical decisions

p.3: To detect user behavior patterns, the interface must be able to track all basic interface events. – Highlighted May 2, 2017

This system would allow informal mixing of interface events and developer/expert intuition

Predictive statistical models for user models tutorial, Andes plan making content-based collaborative: p.11: These enhanced models can then make more accurate predictions about the behaviour of individual users by matching these users to a particular group. – Highlighted May 2, 2017 raised web metrics of ecommerce concerned with commerce, not design

How proactive to be? p.2: The IP continuum expresses possible balances of proactivity between the user and the system: what combination of actions by those two could be responsible for accomplishing a particular task. – Highlighted May 2, 2017

p.3: The level of system proactivity directly impacts the benefit to the user when the system is correct and the cost to the user when the system makes a mistake. – Highlighted May 2, 2017

p.4: Adaptive interfaces also need data from users after they act. The more proactive an adaptive interface, the larger the amount and quality of feedback it needs from users in order to determine whether or not it has acted appropriately and whether it should modify its behavior. – Highlighted May 2, 2017

3.2 Scenario Based Design

Scenario based design is a quick and dirty way to communicate envisioned use and who is using a tool

SBD has been used in many contexts

SBD is widely used and provides concrete examples that are easy to communicate

SBD is qualitative – requirements are formal

PSBD could be a bridge between the two

3.3 Bayesian Inference

Bayesian networks are common in user modeling studies

Here, we don't propose a conditional probability bayesian network – that would be difficult to construct by hand

Thought for future research: Bayesian network for design

If we had PSBD, we could use Bayesian inference to see if our expectations are met by actual use We could improve our expectation for future interfaces, even if we don't build the same interface again (better priors)

Bayesian data assimilation is often used in weather forecasting, etc

Could extend this to a multivariate case, right now its a simple univariate case

3.4 Therefore,

PSBD would

4 Limitations

Predictive user model (plan making) AI algorithms assume that users maintain their preferences over time p.10: An implicit working assumption of many predictive models is that of “persistence of interest” (Lieberman, 1995), whereby users maintain their behaviour or interests over time. – Highlighted May 2, 2017

But, the group as a whole can change their preferences, and this system will reflect that paradigm shift p.10: An implicit working assumption of many predictive models is that of “persistence of interest” (Lieberman, 1995), whereby users maintain their behaviour or interests over time. – Highlighted May 2, 2017

4.1 Bayesian Assimilation in Interface Design

4.2 Case Study Introduction

Here, I introduce a hypothetical interface with two components and two target user groups as a case study with which to demonstrate my method.

5 Method

The method of underpinning a use-case scenario with probabilities of component use has two prerequisites. First, the application developer and stakeholders must work together to come up with a wireframe of the intended interface. A wireframe of the application, including all components, their function, and their position on the page is essential to this method. Wireframes can be either high-fi or low-fi, as long as all stakeholders are aware of the components and their functionality. The wireframes will be used to identify and name the components when assigning probabilities of use to the interface.

A second prerequisite is that the application developer have one or more clearly defined user personas or use case scenarios developed for the intended application. This use case should be crisply defined and include the affinities of the users expected to follow this use.

Once both the use cases and wireframes have been developed, the process of developing the formal user model can begin. During this phase, each application component or feature is considered and a probability distribution describing the probability that the intended users will utilize this component is proposed. Additionally, some real-valued distributions can be assigned as well, such as the width of a panel, if the proposed interface allows interactive resizing.

Whether or not a component is used by a hypothetical user following a use case scenario can be modeled as a Bernoulli distribution with probability of success α . Thus, the probability of use of a given component is considered a discrete random variable. In this way, variation between users can be simulated while enforcing patterns between expected groups.

Real valued components can be modeled using a continuous distribution. However, few values evaluated on an interface can be infinite, as most user interaction is constrained in some way. For example, panel widths cannot exceed 100% of the page width or be less than 0% of the width. Therefore, two potentially useful distributions are the gamma distribution and the binomial distribution. The gamma distribution is truncated at zero, with support from zero to infinity. The gamma distribution is effective for modeling positively real-valued phenomena, such as the zoom level of a map component. The binomial distribution is effective for modeling the dimensions of interface components. Given the need for mobile responsive applications, it is often effective to measure the dimensions of a component in percentage terms, rather than in pixels. This percentage can be modeled as a binomial distribution with 100 trials and a probability of success on each trial of α . Thus, as with the Bernoulli distribution described above, group-level similarities are imposed by group-to-group differences in α , while intragroup variation may remain, as it is likely to in real life.

Once a distribution for each component of interest has been proposed by the model developer, it can be negotiated with stakeholders. While the application developer has a clear understanding of expected use, it may not align with the target user group's expectations. Because the distributions are, ultimately, a way of communicating between developers and stakeholders, efforts should be made to refine the distributions with the input of members of the target use cases. Alternate distributions may be proposed, or, more likely, the distribution parameters may be tweaked to more accurately reflect the stakeholder's input. Stakeholders are likely to more effectively understand how that user group would actually utilize application components.

Communication and negotiation with the target users can be done in several ways. Informal interviews with key members of each stakeholder may be the best approach. Other social science data collection methods, including formal interviews or focus groups may also be effective. While an online survey would capture the opinion of more potential users, distribution negotiation should, in most cases, be limited to the users most qualified to comment on the potential use cases of the user group. Real world uses may be compared and used to refine the distributions using Bayesian data assimilation once the application has been released.

Once the distributions of use have been proposed and negotiated, a user model is then available for subsequent analysis. This user model is used to generate a probability generated configuration (PGC) by drawing from the distributions of each component. To create a PGC, a single draw from each component's distribution is taken. Each draw is taken independently of all others. Therefore, a PGC is a configuration of the application potential under the given use case scenario. Variability is inherent within users within a given use case scenario. Because each draw is from a random distribution, this variation is captured while maintaining the salient features of scenario membership. Analyzing the set of PGCs, particularly if PGCs are generated from multiple scenarios, can yield insight into the variability between these scenarios.

Below, a case study describing the proposal and negotiation of the distributions for a new web-based interactive application is presented. PGCs are generated and subsequently analyzed and visualized to understand differences between user groups. Finally, initial data from beta testing is introduced and used to refine probability estimates using Bayesian data assimilation.

6 Case Study

Ice Age Mapper (IAM) is a novel, highly interactive tool for communication of ecological change since the Last Glacial Maximum (LGM). Paleoecological databases offer the primary source of information about the spatial responses of species and communities to large-scale climatic changes. However, the spatiotemporal characteristics, including significant positional and taxonomic uncertainty, and high dimensionality of the fossil record create significant challenges for effective visualization of floral and faunal community change over major shifts in the earth's climate, and, more recently, in response to anthropogenic change. IAM leverages recent advances in interactive cartography, dynamic mapping, and data visualization to promote hypothesis generation and communication of ecological trends through space and time. Drawing on the vast holdings of the Neotoma Paleoecological Database (<http://neotomadb.org>), IAM is capable of displaying maps and information graphics of over 18 million occurrences of mammals, plants, and marine and freshwater organisms over the last 2.5 million years (Williams et al., in press). An early version of IAM is publicly available at <http://paleo.geography.wisc.edu>.

6.1 Use Cases

Ice Age Mapper is a multi-level interface designed for at least two key audiences: undergraduate students and professional ecological researchers. Here, I introduce the use case scenarios for each hypothetical user.

6.1.0.1 User 1: Undergraduate Student

User 1 is a second year undergraduate student currently enrolled in an introductory paleoecology/paleoclimatology class. Her use of Ice Age Mapper will include (1) viewing pre-configured maps that accompany important concepts presented in class and (2) a homework assignment designed to enhance understanding of climatic influence on biotic patterns in North America since the LGM. Preconfigured maps may be introduced by the professor to illustrate and contextualize important events that are difficult to place in time, space, and history. The homework assignment will be aimed at (1) a general understanding of the dynamic nature of biotic assemblages in the recent geologic past (e.g., northward shifts) and (2) identifying specific changes and/or climatic events (e.g., Younger Dryas). The application should facilitate finding answers to both broad questions (e.g., in general, how do ranges shift since the LGM?, how have climatic patterns influenced biotic patterns since the LGM, if at all?) and specific questions (e.g., what changes did you observe in _____'s range? Do you notice any patterns in _____'s relative abundance at different points in time?, How did _____'s presence in the Midwest's fossil record change?). The student will be expected to prepare written responses to the assignment but does not require access to / knowledge of the raw data. Throughout her use, User 1 will make heavy use of the contextual layers (ice sheets, northern hemisphere temperature) as a grounding point between what she knows and what she is learning.

6.1.0.2 User 2: Researcher

User 2 is a paleoecologist by trade and a PI on the Neotoma project. She will use Ice Age Mapper as (1) a hypothesis generation tool for her own research, primarily to reduce the time required to download and look at data from Neotoma. User 2 is a world expert in climate change and its influence of biotic patterns and has published both field work (i.e., contributed a dataset to Neotoma) and synthetic studies. Her use of Ice Age Mapper will include (a) identifying outliers (e.g., what's that site doing way out there?), (b) sampling priorities (e.g., why hasn't anyone found _____ here?), and (c) prior research efforts (e.g., formerly collected data and its PI). To facilitate this workflow, she requires fine-grained control over multiple filters at once. Once a possible hypothesis has been generated, she may wish to access the underlying data directly for use in statistical analyses or visualization in another software package.

6.2 Wireframes

6.3 Initial Proposals for Distributions

In this problem, we'll try to identify between our two key user groups:

- *Students*: Using the interface for a class assignment
- *Researchers*: Using the interface for an in-depth analysis of paleoecological patterns

Here, we will take the scenarios above and translate them into probabilities using the author's expert judgement and understanding of expected use. Each interface component will have a probability of being used. Panel components have additional estimates of their width, as this component may be resized interactively by the user. Finally, the map component will have a distribution modeling its zoom level and geographic center. Additional components, such as whether useful metadata will accompany the map and the entry point to arrive at the map are also modeled in this framework. The initial probabilities are shown in the tables below.

Variable	Interpretation
hasAuthor	Does the map have an author
hasOrganization	Does the map have an institution associated with it
hasTitle	Does the map have a title associated with it
hasDescription	Does the map have a description associated with it
usedTaxonIDNumber	Did the map create use the search by name function or the browse function to a
configdata.state.openSite	Is the site details panel open
configdata.state.layout.westPanelIsOpen	Is the site details panel open
configdata.state.layout.southPanelIsOpen	Is the bottom timeline panel open
configdata.state.layout.eastPanelIsOpen	Is the right dashboard panel open
configdata.state.doSearch	Does the map come preconfigured with data
configdata.state.map.zoom	What zoom level is the map at
configdata.config.timer.totalElapsed	How much time has elapsed since the session started?
configdata.config.timer.loadElapsed	How much time has elapsed since loading data onto the map
configdata.config.layout.eastPanelSize	Percentage of screen covered by dashboard
configdata.config.layout.westPanelSize	Percentage of screen covered by site details
configdata.config.layout.southPanelSize	Percentage of screen covered by bottom timeline
usedAgeFilter	Does the configuration use the time filter
usedLatitudeFilter	Does the configuration use the latitude filter
usedAbundanceFilter	Does the configuration use the abundance filter
usedPIFilter	Does the configuration use the investigator filter
usedRecordTypeFilter	Does the configuration use the record type filter?

Variable	Interpretation
hasAuthor	Does the map have an author
hasOrganization	Does the map have an institution associated with it
hasTitle	Does the map have a title associated with it
hasDescription	Does the map have a description associated with it
usedTaxonIDNumber	Did the map create use the search by name function or the browse function to a
configdata.state.openSite	Is the site details panel open
configdata.state.layout.westPanelIsOpen	Is the site details panel open
configdata.state.layout.southPanelIsOpen	Is the bottom timeline panel open
configdata.state.layout.eastPanelIsOpen	Is the right dashboard panel open
configdata.state.doSearch	Does the map come preconfigured with data
configdata.state.map.zoom	What zoom level is the map at
configdata.config.timer.totalElapsed	How much time has elapsed since the session started?

Variable	Interpretation
configdata.config.timer.loadElapsed	How much time has elapsed since loading data onto the map
configdata.config.layout.eastPanelSize	Percentage of screen covered by dashboard
configdata.config.layout.westPanelSize	Percentage of screen covered by site details
configdata.config.layout.southPanelSize	Percentage of screen covered by bottom timeline
usedAgeFilter	Does the configuration use the time filter
usedLatitudeFilter	Does the configuration use the latitude filter
usedAbundanceFilter	Does the configuration use the abundance filter
usedPIFilter	Does the configuration use the investigator filter
usedRecordTypeFilter	Does the configuration use the record type filter?

Once the probabilities have been assigned, they are read into R. R generated a probability density function from the model then draws N random samples from that distribution. In this way, we have N ‘users’ in each group that we can analyze as real data.

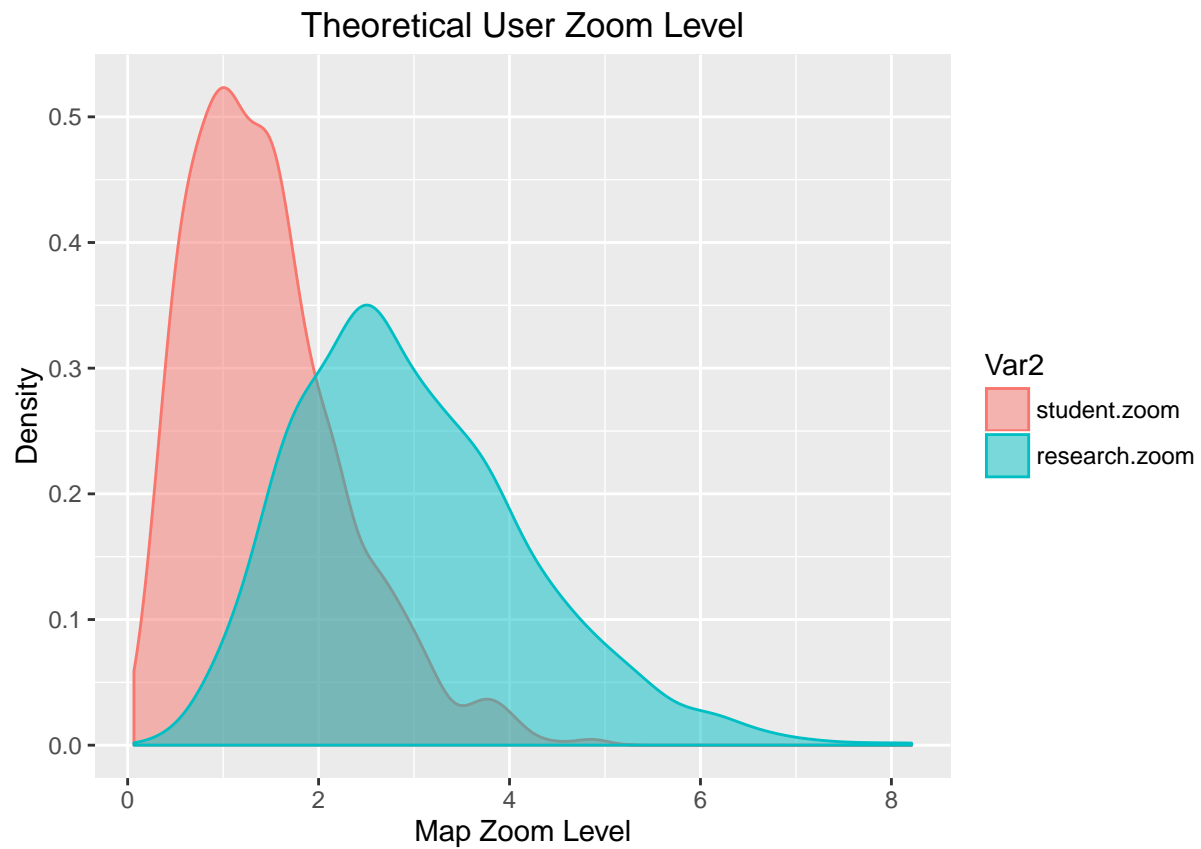
hasAuthor	hasOrganization	hasTitle	hasDescription	usedTaxonIDNumber
TRUE	TRUE	TRUE	FALSE	TRUE
TRUE	FALSE	TRUE	FALSE	TRUE
TRUE	TRUE	FALSE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	TRUE	TRUE	TRUE

6.4 Negotiation of Initial Distributions

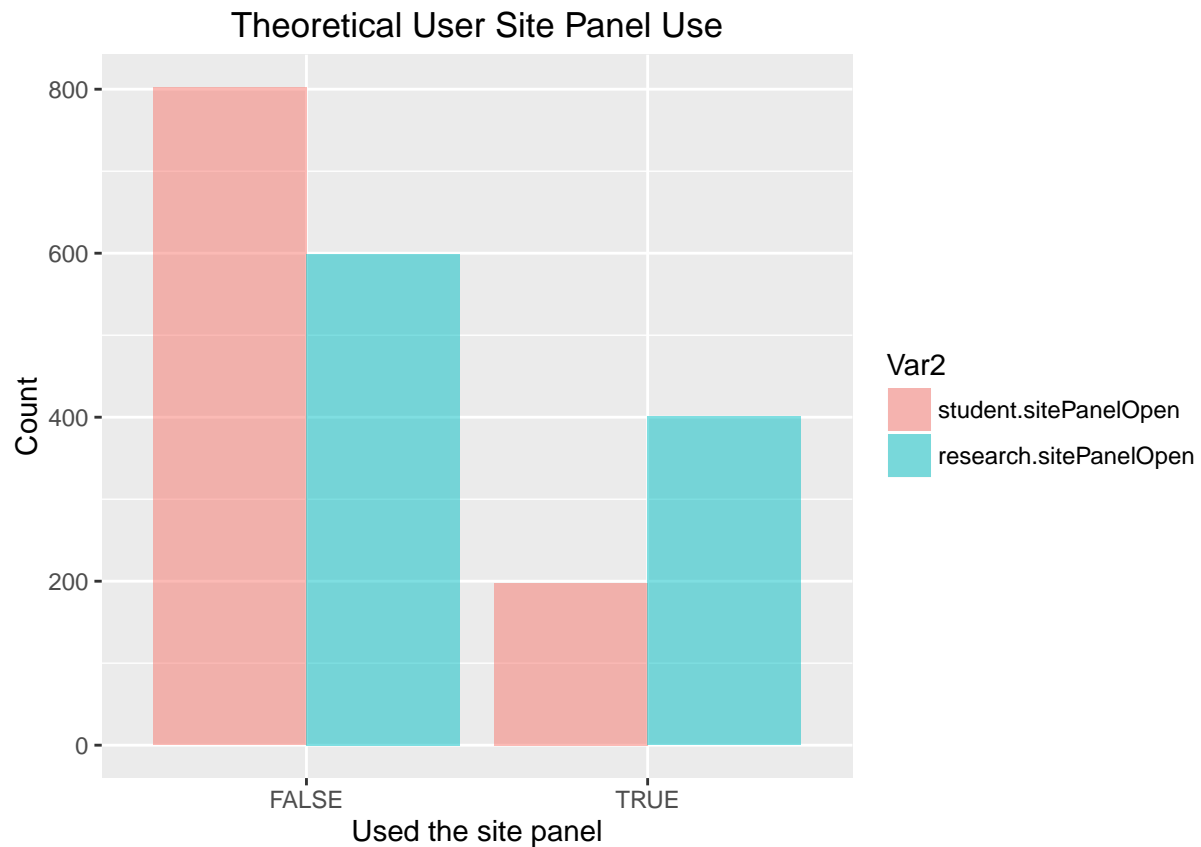
Distributions were negotiated in a set of informal interviews carried out in April, 2017. Research uses were negotiated with Jack Williams, a paleoecologist and lead investigator on the Neotoma Paleocology Database. Student uses were negotiated with _____. Distributions were also refined through informal interviews with Dr. Robert Roth.

6.5 Visually Assess the Data

Using this data, we can graphically assess whether it appears that there are differences in the two models (we built the models, so we know that they are – but we can see how that appears as a PDF here). When we get real data, we can see how the theoretical models align with the real usership data. Also, we can, I think, integrate the new data using Bayesian data assimilation. More on that as the idea develops.



Here, we can see the difference in zoom levels inherent between the two groups of users. Students are more interested in continental level inference, while researchers are more likely to focus on a study area they are already familiar with, or identify a new one.

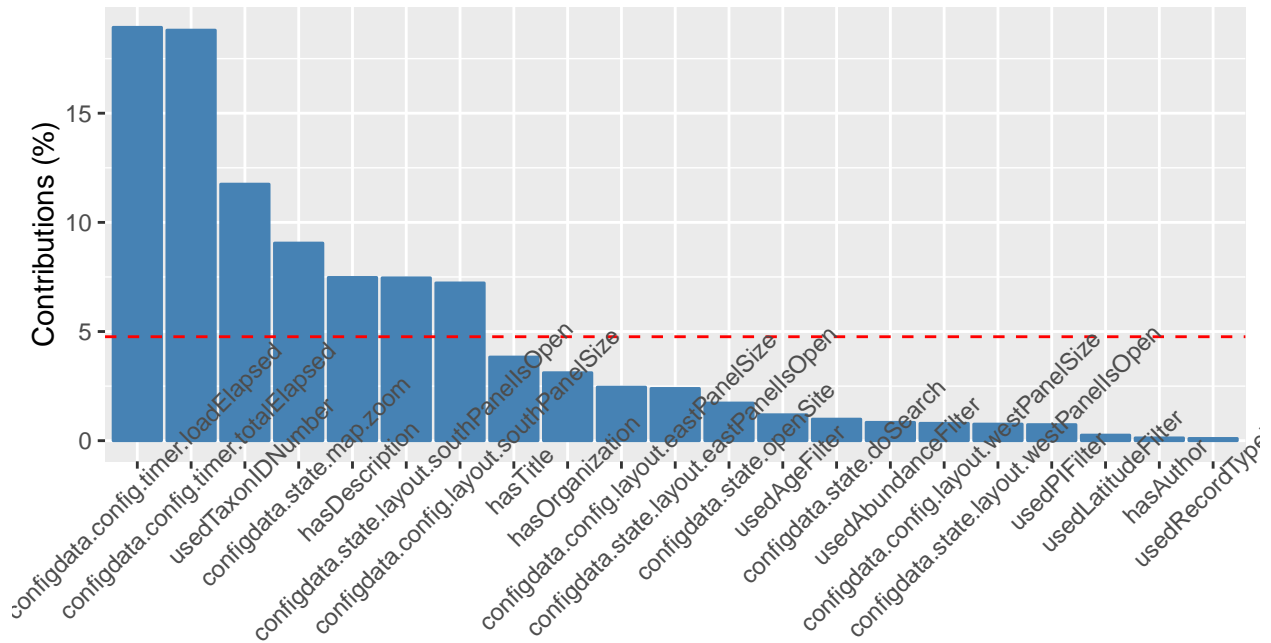


We can also see how likely one group is to use one of the components of our interface, such as the use of the site details panel.

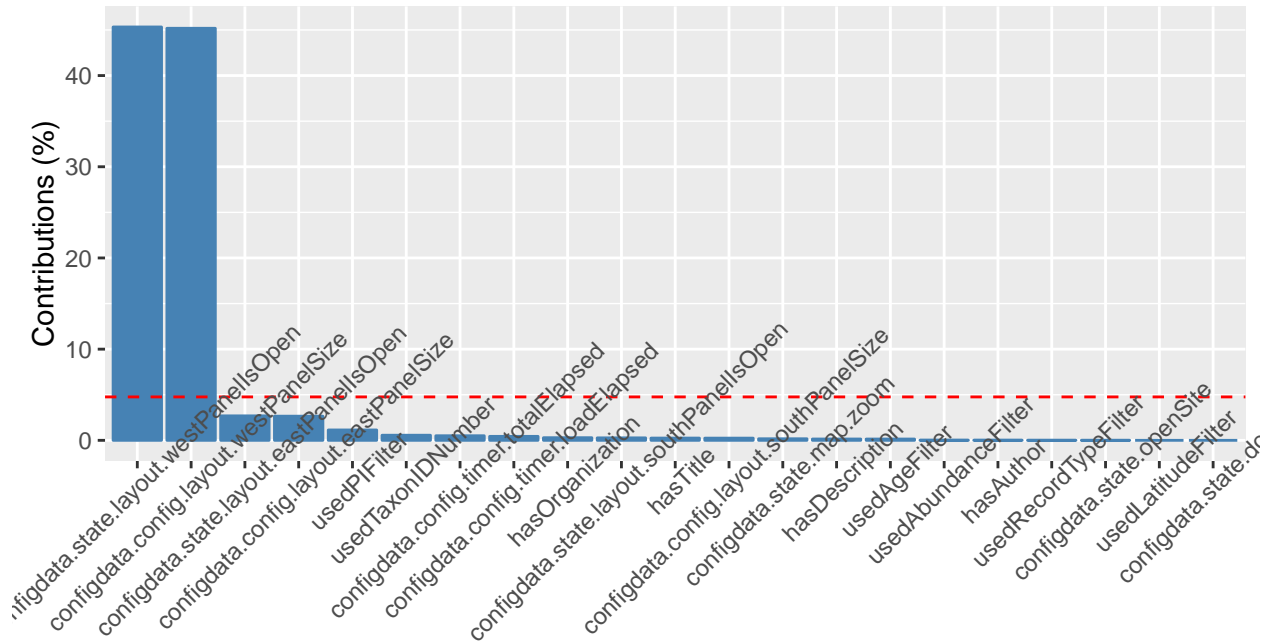
6.6 Principle Components Analysis

Here, we have many variables that may contribute to the variation in the interface design. First, we will do a PCA on the data to reduce its dimensionality and, hopefully, see meaningful differences between our groups of users.

Contribution of variables to Dim-1



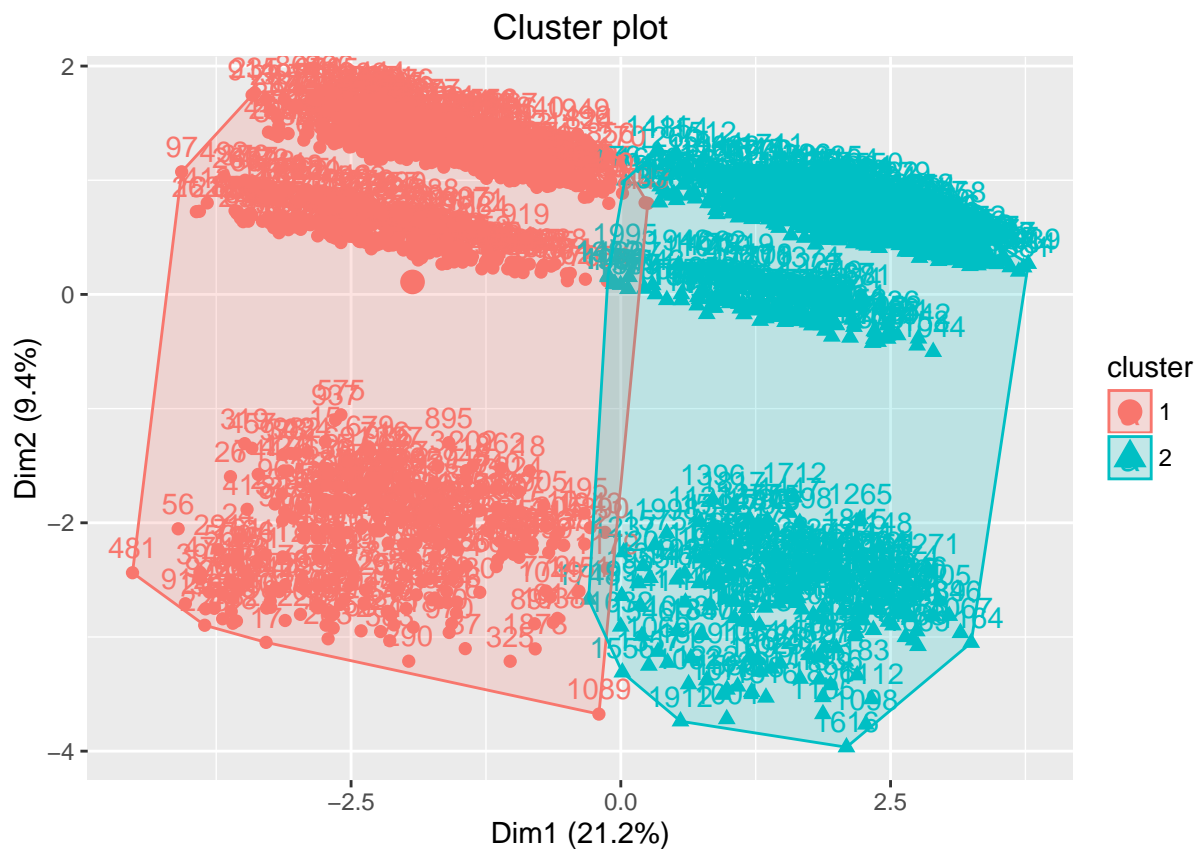
Contribution of variables to Dim-2



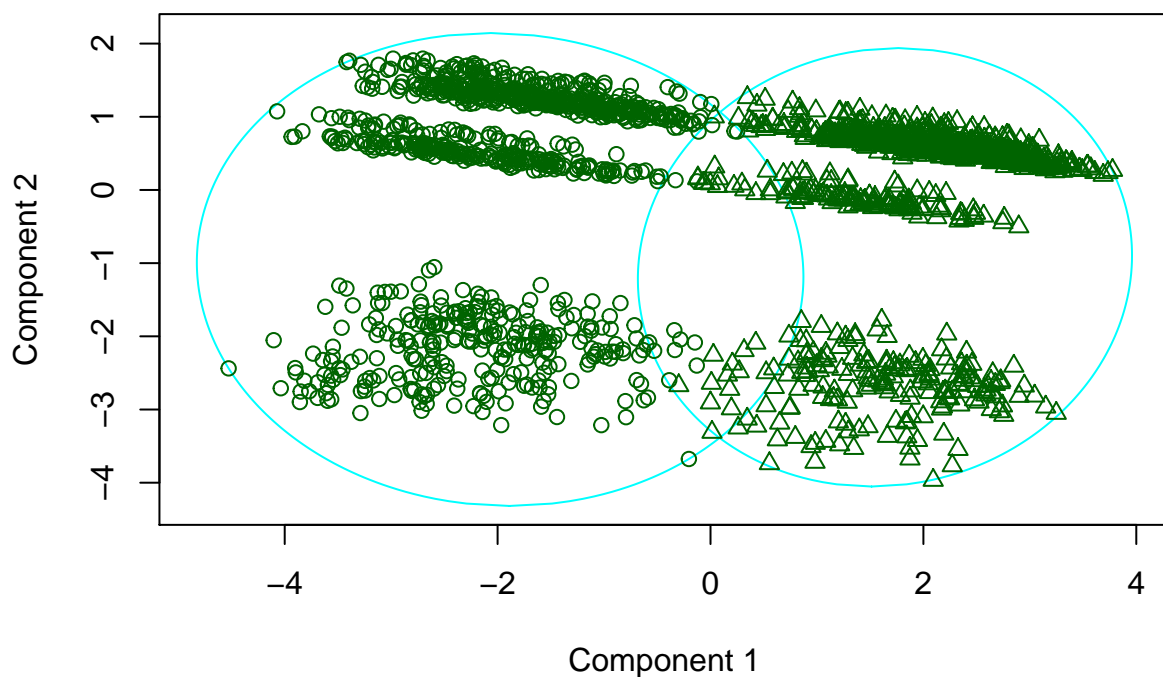
6.7 Cluster Analysis

Next, we can attempt to cluster the groups into meaningful groups using an unsupervised learning approach. In real life, users are not crisply grouped into classes. Instead, they may have some attributes on one class and some attributes of another. For example, is a graduate student a *researcher* or a *student*? In reality, she will likely display usership characteristics of both classes.

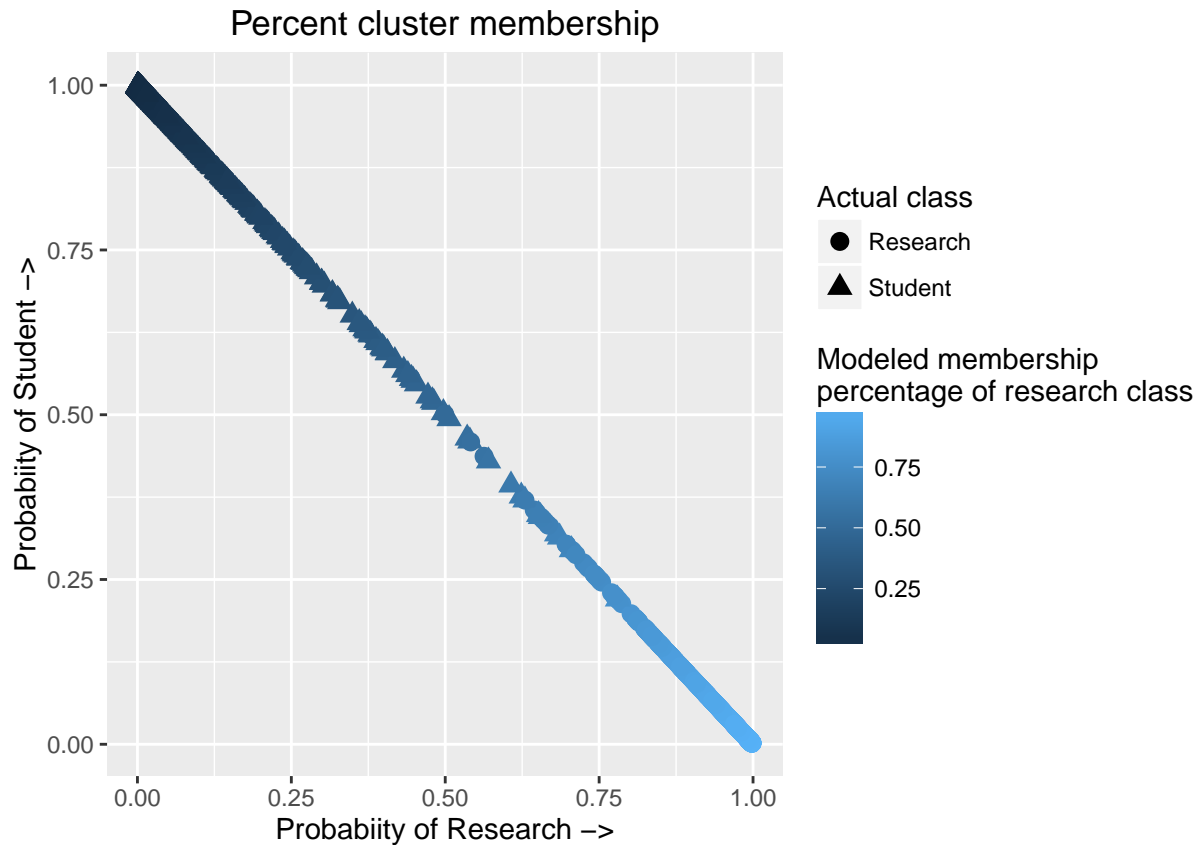
Here we will use fuzzy logic to cluster the users into two groups, letting each use case have a fuzzy degree of membership in each of the classes. For example, the graduate student may have a 25% membership in the student category and a 75% membership in the research class.



Cluster membership, k= 2



These two components explain 30.54 % of the point variability.



6.8 Bayesian Data Assimilation

7 Discussion

7.1 Statistical Testing of Usage Differences

7.2 Novel Visualization Potential

7.3 Bayesian Assimilation with Real Usage Patterns

```
library(reshape2)
library(ggplot2)

makePrior <- function(m,n){
  a = n * m
  b = n * (1 - m)
  dom <- seq(0,1,0.001)
  val <- dbeta(dom,a,b)
  return(data.frame('x'=dom, 'y'=val))
}

makeLikelihood <- function(N,Y){
```

```

a <- Y + 1
b <- N - Y + 1
dom <- seq(0,1,0.001)
val <- dbeta(dom,a,b)
return(data.frame('x'=dom, 'y'=val))
}

makePosterior <- function(m, n, N, Y, name){
  a <- Y + (n*m) -1
  b <- N - Y + (n*(1-m)) - 1
  dom <- seq(0,1,0.001)
  val <- dbeta(dom,a,b)
  return(data.frame('x'=dom, 'name' = name, 'y'=val))
}

mean_of_posterior <- function(m,n,N,Y){
  a <- Y + (n*m) -1
  b <- N - Y + (n*(1-m)) - 1
  E_posterior <- a / (a + b)
  return(E_posterior)
}

NObs <- 10
theta_true <- 0.5
successes <- rbinom(NObs, 1, theta_true)
obs <- data.frame(successes = successes, trials = rep(1, NObs))

mode_of_posterior <- function(m,n,N,Y){
  a <- Y + (n*m) -1
  b <- N - Y + (n*(1-m)) - 1
  mode_posterior <- (a-1)/(a+b-2)
  return(mode_posterior)
}

assimilate <- function(m, n, obs){
  output <- data.frame(x = c(), iter=c(), value=c())
  ctx <- 1
  for (ctx in 1:nrow(obs)){
    thisObs <- obs[ctx,]
    numTrials <- thisObs$trials
    numSuccess <- thisObs$successes
    thisIter <- paste("POSTERIOR", ctx, sep="_")
    thisPost <- makePosterior(m, n, numSuccess, numTrials, thisIter)
    me <- mean_of_posterior(m, n, numSuccess, numTrials)
    m <- me
    print(c(m, n, numTrials, numSuccess))
    output <- rbind(output, thisPost)
    n = n + 1
  }
}

```

```

    return(output)
}

assim_dist <- assimilate(0.5, 10, obs)

ggplot(aes(x = x), data=assim_dist) + geom_line(aes(y = y, group=name, col=name)) + scale_color_discrete()

n <- 100
m <- 0.25

NObs <- 25
theta_true <- 0.75
successes <- rbinom(NObs, 1, theta_true)
obs <- data.frame(successes = successes, trials = rep(1, NObs))

# output <- list()
# ctx <- 1
# for (ctx in 1:nrow(obs)){
#   thisObs <- obs[ctx,]
#   numTrials <- thisObs$trials
#   numSuccess <- thisObs$successes
#   print(numTrials)
#   print(numSuccess)
#   thisIter <- paste("POSTERIOR", ctx, sep="_")
#   thisPost <- makePosterior(m, n, numTrials, numSuccess, thisIter)
#   m <- mean_of_posterior(m, n, numTrials, numSuccess)
#   # print(c(m, n, numTrials, numSuccess))
#   # output <- rbind(output, thisPost)
#   n = n + 1
# }

```

8 Conclusion

9 References

```

library(LearnBayes)
findBeta <- function(quantile1,quantile2,quantile3)
{
  # find the quantiles specified by quantile1 and quantile2 and quantile3
  quantile1_p <- quantile1[[1]]; quantile1_q <- quantile1[[2]]
  quantile2_p <- quantile2[[1]]; quantile2_q <- quantile2[[2]]
  quantile3_p <- quantile3[[1]]; quantile3_q <- quantile3[[2]]

  # find the beta prior using quantile1 and quantile2
  priorA <- beta.select(quantile1,quantile2)
  priorA_a <- priorA[1]; priorA_b <- priorA[2]

  # find the beta prior using quantile1 and quantile3
  priorB <- beta.select(quantile1,quantile3)
  priorB_a <- priorB[1]; priorB_b <- priorB[2]
}

```



```

# find the best possible beta prior
diff_a <- abs(priorA_a - priorB_a); diff_b <- abs(priorB_b - priorB_b)
step_a <- diff_a / 100; step_b <- diff_b / 100
if (priorA_a < priorB_a) { start_a <- priorA_a; end_a <- priorB_a }
else { start_a <- priorB_a; end_a <- priorA_a }
if (priorA_b < priorB_b) { start_b <- priorA_b; end_b <- priorB_b }
else { start_b <- priorB_b; end_b <- priorA_b }
steps_a <- seq(from=start_a, to=end_a, length.out=1000)
steps_b <- seq(from=start_b, to=end_b, length.out=1000)
max_error <- 1000000000000000000
best_a <- 0; best_b <- 0
for (a in steps_a)
{
  for (b in steps_b)
  {
    # priorC is beta(a,b)
    # find the quantile1_q, quantile2_q, quantile3_q quantiles of priorC:
    priorC_q1 <- qbeta(c(quantile1_p), a, b)
    priorC_q2 <- qbeta(c(quantile2_p), a, b)
    priorC_q3 <- qbeta(c(quantile3_p), a, b)
    priorC_error <- abs(priorC_q1-quantile1_q) +
      abs(priorC_q2-quantile2_q) +
      abs(priorC_q3-quantile3_q)
    if (priorC_error < max_error)
    {
      max_error <- priorC_error; best_a <- a; best_b <- b
    }
  }
}
print(paste("The best beta prior has a=",best_a,"b=",best_b))
return(list(a=best_a, b=best_b))
}

quantile1 <- list(p=0.5, x=0.85) # we believe the median of the prior is 0.85
quantile2 <- list(p=0.75,x=0.95) # we believe the 99.999th percentile of the prior is 0.95
quantile3 <- list(p=0.25,x=0.60) # we believe the 0.001st percentile of the prior is 0.60

# findBeta(quantile1,quantile2,quantile3)

calcPosteriorForProportion <- function(successes, total, a, b)
{
  # Adapted from triplot() in the LearnBayes package
  # Plot the prior, likelihood and posterior:
  likelihood_a = successes + 1; likelihood_b = total - successes + 1
  posterior_a = a + successes; posterior_b = b + total - successes
  theta = seq(0.005, 0.995, length = 500)
  prior = dbeta(theta, a, b)
  likelihood = dbeta(theta, likelihood_a, likelihood_b)
  posterior = dbeta(theta, posterior_a, posterior_b)
}

```

```

m = max(c(prior, likelihood, posterior))
plot(theta, posterior, type = "l", ylab = "Density", lty = 2, lwd = 3,
      main = paste("beta(", a, ",", b, ") prior, B(", total, ",", successes, ") data,",
                    "beta(", posterior_a, ",", posterior_b, ") posterior"), ylim = c(0, m), col = "red")
lines(theta, likelihood, lty = 1, lwd = 3, col = "blue")
lines(theta, prior, lty = 3, lwd = 3, col = "green")
legend(x=0.8,y=m, c("Prior", "Likelihood", "Posterior"), lty = c(3, 1, 2),
      lwd = c(3, 3, 3), col = c("green", "blue", "red"))
# Print out summary statistics for the prior, likelihood and posterior:
calcBetaMode <- function(aa, bb) { BetaMode <- (aa - 1)/(aa + bb - 2); return(BetaMode); }
calcBetaMean <- function(aa, bb) { BetaMean <- (aa)/(aa + bb); return(BetaMean); }
calcBetaSd <- function(aa, bb) { BetaSd <- sqrt((aa * bb)/(((aa + bb)^2) * (aa + bb + 1))); return(BetaSd); }
prior_mode <- calcBetaMode(a, b)
likelihood_mode <- calcBetaMode(likelihood_a, likelihood_b)
posterior_mode <- calcBetaMode(posterior_a, posterior_b)
prior_mean <- calcBetaMean(a, b)
likelihood_mean <- calcBetaMean(likelihood_a, likelihood_b)
posterior_mean <- calcBetaMean(posterior_a, posterior_b)
prior_sd <- calcBetaSd(a, b)
likelihood_sd <- calcBetaSd(likelihood_a, likelihood_b)
posterior_sd <- calcBetaSd(posterior_a, posterior_b)
print(paste("mode for prior=", prior_mode, ", for likelihood=", likelihood_mode, ", for posterior=", posterior_mode))
print(paste("mean for prior=", prior_mean, ", for likelihood=", likelihood_mean, ", for posterior=", posterior_mean))
print(paste("sd for prior=", prior_sd, ", for likelihood=", likelihood_sd, ", for posterior=", posterior_sd))
return(posterior)
}

p <- calcPosteriorForProportion(45, 50, 52.22, 9.52)

makeQuantiles <- function(dist, quantiles=c(0.25, 0.5, 0.75)){
  x1 <- quantile(dist, quantiles[1])[[1]]
  q1 <- list(p=quantiles[[1]], x=x1)
  x2 <- quantile(dist, quantiles[2])[[1]]
  q2 <- list(p=quantiles[[2]], x=x2)
  x3 <- quantile(dist, quantiles[3])[[1]]
  q3 <- list(p=quantiles[[3]], x=x3)
  return(list(q1, q2, q3))
}

# q <- makeQuantiles(p)

# d <- findBeta(q[[1]], q[[2]], q[[3]])

p <- calcPosteriorForProportion(1, 10, 10, 10)

x <- beta.select(list(p=0.1, x=quantile(p, 0.1)), list(p=0.5, x=quantile(p, 0.5)))
p <- calcPosteriorForProportion(1, 10, x[[1]], x[[2]])

library(rjags)
n <- 2
Y <- 1
a <- 1

```

```

b      <- 2

model_string <- "model{

  # Likelihood
  Y ~ dbinom(theta,n)

  # Prior
  theta ~ dbeta(a, b)
}"

model <- jags.model(textConnection(model_string),
                    data = list(Y=Y,n=n,a=a,b=b))

update(model, 10000, progress.bar="none"); # Burnin for 10000 samples

samp <- coda.samples(model,
                     variable.names=c("theta"),
                     n.iter=20000, progress.bar="none")

summary(samp)

q <- as.numeric(as.mcmc(samp))
plot(density(q))
# t1 <- findBeta(list(x=0.25, p=quantile(q)[[2]]), list(x=0.75, p=quantile(q)[[4]]), list(x=0.5, p=quantile(q)[[6]]))
# lines(density(rbeta(10000, t1[[1]], t1[[2]])), col='green')
#
#
# p <- calcPosteriorForProportion(Y, n, a, b)
# t2 <- findBeta(list(x=0.25, p=quantile(p)[[2]]), list(x=0.75, p=quantile(p)[[4]]), list(x=0.5, p=quantile(p)[[6]]))
# lines(density(rbeta(10000, t2[[1]], t2[[2]])), col='red')

```