Bayesian Data Analysis - exercises

Exercise 1: Experimenting with priors and predictives

In our simple binomial model (/chapters/14-bayesian-data-analysis.html#a-simple-illustration), we compared the parameter priors and posteriors to the corresponding **predictives** which tell us what data we should expect given our prior and posterior beliefs. For convenience, we've reproduced that model here:

```
x

// observed data
var k = 1 // number of successes
var n = 20  // number of attempts
var priorDist = Uniform({a: 0, b: 1});

var model = function() {
   var p = sample(priorDist);

   // Observed k number of successes, assuming a binomial
   observe(Binomial({p : p, n: n}), k);

   // sample from binomial with updated p
   var posteriorPredictive = binomial(p, n);

   // sample fresh p (for visualization)
   var prior_p = sample(priorDist);
   // sample from binomial with fresh p (for visualization)
   var priorPredictive = binomial(prior_p, n);

   return {
       prior: prior_p, priorPredictive : priorPredictive,
       posterior : p, posteriorPredictive : posteriorPredictive
    };
}

var opts = {method: "MCMC", samples: 2500, lag: 50};
var posterior = Infer(opts, model);

viz.marginals(posterior)
```

run ▼

a. Notice that we used a uniform distribution over the interval [0,1] as our prior, reflecting our assumption that a probability must lie between 0 and 1 but otherwise remaining agnostic to which values are most likely to be the case. While this is convenient, we may want to represent other assumptions. The Beta distribution (https://en.wikipedia.org/wiki/Beta_distribution), expressed in WebPPL as `Beta({a:..., b:...})`' is a more general way of expressing beliefs over the interval [0,1].

Try different beta priors on `p`, by changing `priorDist = Uniform(...)` to `p = Beta({a: 10,b: 10})`, `Beta({a: 1, b: 5})` and `Beta({a: 0.1, b: 0.1})`. (Note that `beta(1,1)` is mathematically the same as `uniform(0,1)`.) Use the figures produced to describe the assumptions these priors capture, and how they interact with the same data to produce posterior inferences and predictions.

b. Predictive distributions are not restricted to exactly the same experiment as the observed data, and can be used in the context of any experiment where the inferred model parameters make predictions. In the current simple binomial setting, for example, predictive distributions could be found by an experiment that is different because it has `n' != n` observations. Change the model to implement an example of this.

Exercise 2: Parameter fitting vs. Parameter integration

One of the strongest motivations for using Bayesian techniques for model-data evaluation is in how "nuisance" parameters are treated. "Nuisance" parameters are parameters of no theoretical interest; their only purpose is to fill in a necessary slot in the model. Classically, the most prominant technique (from the frequentist tradition) for dealing with these parameters is to fit them to the data, i.e., to set their value equal to whatever value maximizes the model-data fit (or, equivalently, minimizes some cost function).

The Bayesian approach is different. Since we have *a priori* uncertainty about the value of our parameter, we will also have *a posteriori* uncertainty about the value (though hopefully the uncertainty will be reduced). What the Bayesian does is *integrate over* her posterior distribution of parameter values to make predictions. Intuitively, rather than taking the value corresponding to the peak of the distribution (i.e., the maximum), she's considering all values with their respective probabilites.

MAP是点估计，虽然在求解参数w的时候也将w视作随机变量，但在做预测时，是以后验分布P(w|D)中的峰值w'作为w的取值。

Bayesian Inference在得到w的后验分布P(w|D)后, 在这个分布上对y求期望，integrate over posterior distribution of parameter values to make predictions。

MAP不需要求出后验分布，找到峰值就行；Bayesian Inference是要对后验分布积分。

Why might this be important for model assessment? Imagine the following situation. You are piloting a task and want to use Bayesian Data Analysis because you hear it is useful when you have few data points. You think that the task you've designed is a little too difficult for subjects. (Let's imagine that you're a psychophysicist, and your task pertains to contrast discriminiation in the peripheral visual field.) You think the current task design is too difficult, but you're not sure. It may well be that it's fine for subjects.

Here is your prior.

```

// Prior on task difficulty is uniform on [0, ..., 0.9], with a spike on 0.9
// i.e., you think it's likely that the task is too difficult
var sampleTaskDifficulty = function() {
  return flip() ? .9 : randomInteger(10) / 10;
};

var model = function() {
  return sampleTaskDifficulty();
};
```

```

viz.hist(Infer({method: 'enumerate'}, model), {numBins: 9})
```

run ▼

比如正确率反应时和任务难度相关，但是你其实是不知道难度和这些指标到底是什么关系。

You have a model of how subjects perform on your task. You could have a structured, probabilistic model here. For simplicity, let's assume you have the simplest model of task performance. It is a direct function of task-difficulty: subjects perform well if the task isn't too difficult.

```
var subjectPerformWell = !flip(taskDifficulty)
```

在大部分实验中，不是一个难或者不难而是一个连续的量。

There's a lot of training involved in your task and that it's very time consuming for you to collect data. You run one subject through your training regime and have them do the task. The subject performs well! Soon after, your adviser drops by and wants you to make a decision to collect more data or tweak your experimental paradigm. You thought beforehand that your task was too difficult. Do you still think your task is too hard?

真实………

Since you wrote down your prior beliefs, we can examine how much the data update those beliefs about the `taskDifficulty` parameter. How does your degree of belief in task difficult change as a result of your one pilot subject performing well?

```

// Prior on task difficulty is uniform on [0, ..., 0.9], with a spike on 0.9
var sampleTaskDifficulty = function() {
  return flip() ? .9 : randomInteger(10) / 10;
};

// Compute posterior after seeing one subject perform well on the task
var taskDifficultyPosterior = Infer({method: 'enumerate'}, function(){
  var taskDifficulty = sampleTaskDifficulty();

  // subject will perform well if the task is not too difficult
  var subjectPerformsWell = !flip(taskDifficulty)

  // observe that they perform well (i.e. this value is true)
  condition(subjectPerformsWell)
  return taskDifficulty;
})

// Most likely task-difficulty is still .9
taskDifficultyPosterior.MAP().val

// But a lot of probability mass is on lower values
viz.hist(taskDifficultyPosterior, {numBins: 9})

// Indeed, the expected subject ability is around .4
expectation(taskDifficultyPosterior)
```

run ▼

A. Would you proceed with more data collection or would you change your paradigm? How did you come to this conclusion?

B. In part A, you probably used either a value of task-difficulty or the full distribution of values to decide about whether to continue data collection or tweak the paradigm. We find ourselves in a similar situation when we have models of psychological phenomena and want to decide whether or not the model has fit the data (or, equivalently, whether our psychological theory is capturing the phenomenon). The traditional approach is the value (or “point-wise estimate”) approach: take the value that corresponds to the best fit (e.g., by using least-squares or maximum-likelihood estimation; here, you would have taken the Maximum A Posteriori (or, MAP) estimate, which would be 0.9). Why might this not be a good idea? Provide two answers. One that applies to the data collection situation above, and one that applies to the metaphor of model or theory evaluation.

Exercise 3: BDA of Bayesian Cognitive Models

We saw in this chapter how to analyze our models of cognition by using Bayesian statistical techniques. Compare and contrast the results of our cognitive model of tug-of-war with our regression models. Please consider each question and provide a couple sentences describing your answer.

A. Was there a phenomena in the data that either model was better able to capture?

B. What, if anything did the models in part A fail to capture?

C. Are there other aspects of the model you could 'lift' into the Bayesian Data Analysis (i.e. fixed parameters that you could put a prior on and include in your joint inference)?

D. How does WebPPL expose commonalities between these two models?

BDA：模型的可解释性
BDA把每个过程都写出来，再对condition on observation，得到可能的参数

LR：模型的解释率
LR只是直接去fit方程的参数，minimize cost function，参数并没有任何意义

Exercise 4

Let's continue to explore the inferences you (as a scientist) can draw from the posterior over parameter values. This posterior can give you an idea of whether or not your model is well-behaved. In other words, do the predictions of your model depend heavily on the exact parameter value?

To help us understand how to examine posteriors over parameter settings, we're going to revisit the example of the blicket detector from the chapter on `Conditional Dependence`.

Here is the model, with slightly different names than the original example, and written in a parameter-friendly way. It is set up to display the "backwards blocking" phenomenon.

```
var blicketBaseRate = 0.4

var blicketPower = 0.9

var nonBlicketPower = 0.05

var machineSpontaneouslyGoesOff = 0.05



var blicketPosterior = function(evidence) {

  return Infer({method: 'enumerate'}, function() {

    var blicket = mem(function(block) {return flip(blicketBaseRate)})

    var power = function(block) {return blicket(block) ? blicketPower : nonBlicketPowe

    var machine = function(blocks) {

      return (blocks.length == 0 ?
              flip(machineSpontaneouslyGoesOff) :
              flip(power(first(blocks))) || machine(rest(blocks)))
    }
    // Condition on each of the pieces of evidence making the machine go off
    map(function(blocks){condition(machine(blocks))}, evidence)
    return blicket('A')
  });
};

// A&B make the blicket-detector go off
viz(blicketPosterior([['A', 'B']]))

// A&B make the blicket-detector go off, and then B makes the blicket detector go off
viz(blicketPosterior([['A', 'B'], ['B']]))
```

run ▼

A. What are the parameters of this model? In the plainest English you can muster, interpret the current values of the parameters. What do they mean?

Let's analyze this model with respect to some data. First, we'll put priors on these parameters, and then we'll do inference, conditioning on some data we might have collected in an experiment on 4 year olds, a la Sobel, Tenenbaum, & Gopnik (2004). [The data used in this exercise is schematic data].

```
///fold:
...

```

```
// 5 experiment conditions / stimuli
var possibleEvidenceStream = [
  [['A']],
  [['A', 'B']],
  [['A', 'B'], ['B']],
  [['A', 'B'], ['A', 'B']],
  [[]]
];

// for each condition.
// note: always the question "is A a blicket?"
var data = [
  repeat(10, function(){return true}).concat(false),
  repeat(6 , function(){return true}).concat(repeat(5, function(){return false})),
  repeat(4, function(){return true}).concat(repeat(7, function(){return false})),
  repeat(8, function(){return true}).concat(repeat(3, function(){return false})),
  repeat(2, function(){return true}).concat(repeat(9, function(){return false}))
];

// Same model as above, but parameterized
var detectingBlickets = mem(function(evidence, params) {
  return Infer({method: 'enumerate'}, function() {
    var blicket = mem(function(block) {return flip(params.blicketBaseRate)})
    var power = function(block) {return blicket(block) ? params.blicketPower : params.nonBlick
    var machine = function(blocks) {
      return (blocks.length == 0 ?
              flip(params.machineSpontaneouslyGoesOff) :
              flip(power(first(blocks))) || machine(rest(blocks)))
    }
    map(function(blocks){condition(machine(blocks))}, evidence)
    return blicket('A')
  })
})

var dataAnalysis = Infer({method: 'MCMC', samples: 5000, callbacks: [editor.MCMCProgress()]},
  var params = {
    blicketBaseRate: sample(Uniform({a: 0, b: 1}), {driftKernel: uniformKernel}),
    blicketPower: sample(Uniform({a: 0, b: 1}), {driftKernel: uniformKernel}),
    nonBlicketPower: sample(Uniform({a: 0, b: 1}), {driftKernel: uniformKernel}),
    machineSpontaneouslyGoesOff: sample(Uniform({a: 0, b: 1}), {driftKernel: uniformKernel})
  }

  var cognitiveModelPredictions = map(function(evidence) {
    return detectingBlickets(evidence,params);
  }, possibleEvidenceStream);

  // observe each data point under the model's predictions
  map2(function(dataForStim, modelPosterior) {
    map(function(dataPoint) {
      observe(modelPosterior, dataPoint);
    }, dataForStim)
  }, data, cognitiveModelPredictions)

  var predictives = toProbs(cognitiveModelPredictions)
  return _.extend(params, predictives)
})

viz.marginals(dataAnalysis);
viz.scatter(predictiveSummary(dataAnalysis), dataSummary(data),
            {xLabel: 'model', yLabel: 'data'})
```

run ▼

Before running this program, answer the following question:

给定模型和数据，参数的分布是怎么样的

B. What does the `Infer` statement in `dataAnalysis` return? What does the `Infer` statement in `detectingBlickets` return? Why are there two queries in this program?

给定模型和参数，数据是怎么样的

C. Now, run the program. [Note: This will take between 15-30 seconds to run.] Interpret each of the resulting plots.

D. How do your interpretations relate to the parameter values that were set in the original program?

E. Look carefully at the priors (in the code) and the posteriors (in the plots) over blicketPower and nonBlicketPower. Did we impose any a priori assumptions about the relationship between these parameters? Think about the experimental setup. Do you think we would be justified in imposing any assumptions? Why or why not? What do the posteriors tell you? How was the data analysis model able to arrive at this conclusion?

没有

根据实验，可以加

告诉我这些参数的分布

F. Do you notice anything about the scatter plot? How would you interpret this? Is there something we could add to the data analysis model to account for this?

x轴是模型认为A是blicket的可能性（predict）
y轴是人认为A是blicket的可能性（data）
两者相关程度很高，说明这个4参数的生成模型能很好地解释数据。

G. Now, we’re going to examine the predictions of the model if we had done a more traditional analysis of point-estimates of parameters (i.e. fitting parameters). Examine your histograms and determine the “maximum a posteriori” (MAP) value for each parameter. Plug those into the code below and run it.

```
///fold:
...

var params = {
  blicketBaseRate : ...,
  blicketPower: ...,
  nonBlicketPower: ...,
  machineSpontaneouslyGoesOff: ...
};

var bestFitModelPredictions = map(function(evidence) {
  return Math.exp(detectingBlickets(evidence, params).score(true));
}, possibleEvidenceStream)

```

```
viz.scatter(bestFitModelPredictions, dataSummary(data))
```

run ▼

H. What can you conclude about the two ways of looking at parameters in this model's case? Do you think the model is relatively robust to different parameter settings?