for *Handbook of Market Research*, Homburg, Christian, Martin Klarmann and Arnd Vomberg, eds.
August 2018

Fusion Modeling

Elea McDonnell Feit and Eric T. Bradlow

Abstract This chapter introduces readers to applications of data fusion in marketing from a Bayesian perspective. We will discuss several applications of data fusion including the classic example of combining data on media viewership for one group of customers with data on category purchases for a different group, a very common problem in marketing. While many missing data approaches focus on creating "fused" data sets that can be analyzed by others, we focus on the overall inferential goal, which, for this classic data fusion problem is to determine which media outlets attract consumers who purchase in a particular category and are therefore good targets for advertising. The approach we describe is based on a common Bayesian approach to missing data, using data augmentation within MCMC estimation routines. As we will discuss, this approach can also be extended to a variety of other data structures including mismatched groups of customers, data at different levels of aggregation and more general missing data problems that commonly arise in marketing. This chapter provides readers with a step-by-step guide to developing Bayesian data fusion applications, including an example fully-worked out in the Stan modeling language. Readers who are unfamiliar with Bayesian analysis and MCMC estimation may benefit by reading the chapter in this handbook on Bayesian Models first.

Key words: Data fusion, Data augmentation, Missing data, Bayesian, Markov-chain Monte Carlo

Elea McDonnell Feit
LeBow College of Business, Drexel University, Philadelphia, PA e-mail: efeit@drexel.edu

Eric T. Bradlow
The Wharton School, University of Pennsylvania, Philadelphia, PA e-mail: ebradlow@wharton.upenn.edu

1 Introduction

1.1 The classic data fusion problem in marketing

Like many other fields, numerous situations arise in marketing where the ideal data for analysis is not readily available. For example, in media planning, marketers want to know whether viewers of a particular media (e.g. television channels or shows, magazines, websites, etc.) purchase a particular product (e.g. breakfast cereal or video games), so that they can decide where to place advertising or estimate the association between exposures to ads and purchases. (See the chapter in this handbook on Return on Media Models for more on marketing response modeling.). Ideally, we would like a data set where the media consumption and purchase behavior is tracked for the same set of customers. However, such data is seldom available. Typically, a media tracking firm (e.g. Comscore, Rentrak) collects data on media usage for one set of consumers while another firm tracks data on product purchases (e.g. IRI, Niesen or Dunnhumby for CPG products, Polk for automobiles in the US or IMS Health for pharmaceuticals in the US). Even when media and purchase data are collected by the same firm, it is often impractical to collect that data for the same group of customers and so firms like Nielsen and Kantar, which collect both purchase and media usage data, typically maintain separate panels for media tracking and purchase tracking. Thus, fusing these separate data sources is a classic problem in marketing analytics (cf. Kamakura and Wedel, 1997; Gilula et al, 2006).

While the goal is to measure the correlation between media usage and product purchase, the data structure that we are faced with is like that shown in Fig. 1, where each row (indexed by i) in data set 1 represents a user in the media panel and the variables in data set 1 describe which content (e.g., channels, shows or websites) user i views. (We let y_{i1} denote the vector of observed variables for each user in data set 1). Data set 2 describes a different set of customers in a purchase panel with observed variables describing which products each consumer has purchased (denoted as y_{i2}). The marketing objective is to estimate what types of products the viewers of some content purchase. The key that makes this possible is a vector of common linking variables which are observed both for customers in data set 1 and customers in data set 2 (y_{ib}), where the subscript b indicates "both". These variables are often demographics that are collected for users in both data sets. Typically, these demographics are correlated with both media consumption and product purchases (i.e. new parents may be more likely to visit a parenting website and more likely to buy diapers), which enables data fusion.

Beyond the media and purchase data fusion problem, the data structure depicted in Fig. 1 arises in many other contexts in marketing where we observe one set of variables for one group of customers and another set of variables for another group of customers. For example, two retailers considering a co-branding agreement might want to fuse their separate customers lists to estimate how often customers purchase both brands. And even within companies, growing privacy concerns have led firms to avoid maintaining data sets with personal identifiers (e.g., names or addresses)

for individual customers. The data fusion methods we describe here can be used to link two data sets which have been de-identified to protect user privacy (Qian and Xie, 2014).

In some cases marketing analysts may even plan to create such unmatched data; for instance, in split questionnaire design, marketing researchers minimize the burden on survey respondents by creating a pair of complementary surveys which each can be answered by a subset of the respondents, and later fused back together (Adigüzel and Wedel, 2008). Beyond marketing, in educational testing, the data structure in Fig. 1 occurs when data set 1 is students who take the SAT in 2015, data set 2 is those who take it in 2016, and linking variables are test questions (items) that overlap between the two tests.

In all of these examples, the analysis goal is to understand the multivariate relationships between y_{i1} and y_{i2}. The key to linking the two sets of survey data is a set of questions that is common to both surveys, providing the linking variables described in Fig. 1 (y_{ib}).

There are also several closely-related data settings where similar methods can be employed including survey subsampling and time sampling (cf. Kamakura and Wedel, 2000). In subsampling, some of variables are only collected for a subset of the population, often because those variables are more difficult to collect, e.g. expensive medical tests in population health surveys. In time sampling, a subset of respondents answer a repeated survey at each point in time, avoiding potential respondent fatigue, while still collecting data at the desired time interval.

Early approaches to data fusion tackled it as a database integration problem, where in a first stage we match records in data set 1 with records in data set 2 to create a complete database. Statistical modeling and estimation can then proceed as usual with the now-complete data set. For example, the hot deck procedure (Ford, 1983), and its more sophisticated variants, can be used to match records in the two data sets using a set of *ad hoc* rules to match customer i in data set 1 with

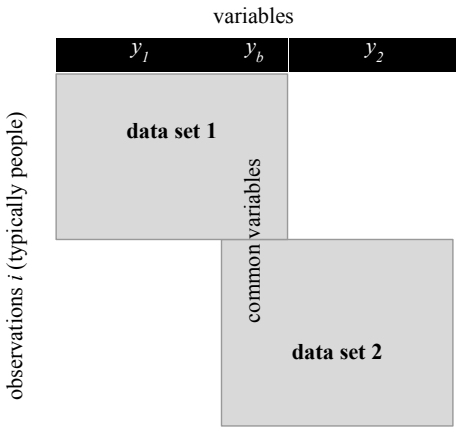

Fig. 1 The traditional data fusion problem is to combine two multivariate data sets with different, but overlapping, sets of variables. This data structure occurs in a number of marketing settings and can be addressed as a Bayesian missing data problem.

a customer j in data set 2 who has the same values of y_{ib}. If there are more than one candiate match, the match is selected randomly. If there are a large number of common variables y_{ib}, such that a perfect match to customer i is not always available, then a nearest neighbor approach can be used to match to customers who are similar. In both hotdeck and nearest neighbor, once all the customers i in data set 1 are matched to a customer j in data set 2, analysis proceeds as if y_{i1} and y_{j2} were observed from the same customer.

A challenging and often ignored aspect of these two-step imputation-then-analysis approaches is that the uncertainty in the imputation is not propagated forward to the statistical modeling stage (Andridge and Little, 2010). In marketing, two-step approaches have become largely superseded by approaches which cast data fusion as a Bayesian missing data problem (Kamakura and Wedel, 1997; Gilula et al, 2006; Qian and Xie, 2014), which is the approach we will focus on in this chapter.

We focus on analyzing data like that in Fig. 1 as a Bayesian missing data problem: y_{i2} are missing for individuals in data set 1 and y_{i1} are missing for data set 2. Thus, while this chapter resides in the section of this book on Data, the approach is more of "a modeling method to handle data that is less than ideal."

A critical step in analyzing any missing data problem is to consider the process by which the missing data came to be missing. Ideally, data is Missing Completely at Random (MCAR) , which means that missingness is unrelated to the observed data or the values of the missing data. When data is MCAR we can ignore the missing data mechanism in data fusion problems.

In data fusion, this assumption would be violated if one or both of the data sets was a biased sample from the target population. For instance, if a media usage data set contains mostly lower-income respondents and the relationship between media usage and product usage is different for low- and high-income respondents, then the missing data in Fig. 1 would not be ignorable for overall population-level inferences. This can happen due to poor sampling methods or survey non-response in one or both of the panels. Respondents frequently avoid answering sensitive questions particularly when the true answer is socially undesirable, e.g. viewing content that one might be embarrassed to admit watching. In these instances, the likelihood of a particular survey response being missing depends on the missing response. In these cases the process that created the missingness can be modeled to avoid bias (Bradlow and Zaslavsky, 1999; Ying et al, 2006).

When fusing two data sets that have been carefully sampled from the same target population, we can assume the data is missing by design, which is a special case of Missing Complete at Random (Little and Rubin, 2014, Ch. 1). In this case, inference can proceed without explicitly modeling the process that led to the missingness. We are not aware of any published examples of data fusion in marketing where sampling bias or non response is modeled, although this is a potential area for future research.

The procedure for handling the missing data in Fig. 1 is as follows. If $f(y_i|\theta)$ is the model for the complete vector of responses $y_i = (y_{i1}, y_{i2}, y_{ib})$ with parameters θ, our inference is based on the likelihood of the observed data y^{obs}, which is given by:

$$f(y^{\text{obs}}|\theta) = \int_{y_1^{\text{mis}}} \int_{y_2^{\text{mis}}} \prod_i f(y_i|\theta) dy_2^{\text{mis}} dy_1^{\text{mis}} \quad (1)$$

where y_1^{mis} are the missing observations of y_{i2} in data set 1 and y_2^{mis} are the missing observations of y_{i1} in data set 2.

One way to estimate θ in Eq. 1 is to create a Bayesian MCMC sampler that samples simultaneously from the posterior of θ, and the posteriors of the missing data elements y_1^{mis} and y_2^{mis}. This approach is referred to as data augmentation (Tanner and Wong, 1987). We will illustrate data augmentation for two alternative specifications of $f(y|\theta)$ in Sec. 2, but first we introduce another closely-related missing data problem that occurs when merging data from separate sources.

1.2 Mixed levels of data aggregation

A second problem that can arise when trying to combine data from two data sources is that the data is provided for individual customers in one data set, but is only available in aggregate in another. In analyzing media usage data, this problem occurs because usage of some media channels like websites and mobile apps are easily tracked and linked at the user-level, while data on exposure to broadcast media like radio, television or outdoor signage is only available in aggregate. For example, we might know from a representative panel (e.g., Nielsen People Meter) that approximately 5.3% percent of a group of users watched a television show, but we do not know exactly which users those were. Media planners would like to understand the co-usage of media channels – are users who watch some content on TV also likely to watch it on mobile or the web – but we can not directly observe the co-usage at the consumer level (Feit et al, 2013).

The resulting mixed aggregate-disaggregate data structure is depicted in Fig. 2 where we observe one set of disaggregate variables, y_{i1t} at the individual-level and only totals, Y_{2t}, for a set of aggregate variables. The managerial goal is to infer the correlations across users between the aggregate and the disaggregate variables, which requires repeated observations of y_{i1t} and Y_{2t} over t.

Beyond the media-planning problem described above, this mixed aggregate-disaggregate data structure occurs in other marketing settings, often due to the limitations of tracking systems. For example, a retailer may have detailed customer-level data on visits to an online store, but only aggregate counts of customer visits in physical stores. Even though this data is deficient, one can still use it to infer how many multi-channel customers there are and often which customers those are. Similarly, retailers often have customer-level data on coupon redemption (tracked as part of the transaction), but only aggregate data on how many coupons are in circulation. Musalem et al (2008) show how to use a Bayesian missing data approach with this data to infer "who has the coupon?," in turn leading to more accurate inference about the effect of coupons on purchases.

Inference for the mixed aggregate-disaggregate data structure described in Fig. 2 can also be viewed as a missing data problem, where the individual-level ob-

servations for the aggregated variables, y_{i1t}, are missing; we only observe a total $Y_{2t} = \sum y_{i2t}$ for each period t. As with the traditional data fusion problem, the y_{i2t} are missing by design and inference can be based on specifying a likelihood for the complete data and then integrating out the missing observations. Specifically, if $f(y_{i1t}, y_{i2t}|\theta)$ is the likelihood for the complete individual-level observations that we do not observe, then inference is based on:

$$f(y_{1t}, Y_{2t}|\theta) = \prod_i \int_{y_{i2t}} f(y_{i1t}, y_{i2t}|\theta) dy_{i2t} \qquad \text{s.t.} \sum_i y_{i2t} = Y_{2t} \tag{2}$$

The key nuance in Eq. 2 is that the integral over the missing data must conform to the constraint implied by the observed marginal totals $\sum y_{i2t} = Y_{2t}$. By observing co-variation between media channels over *time*, and posing a model limiting the co-variation structure in the model so that it is driven by user-level behavior, one can estimate the user-level co-variation in media usage and make inference about which customers are most likely to have been consuming the aggregated media channel on a given day. By contrast, if we were to aggregate all the data and fit an aggregate time-series model with $\sum y_{i1t} = Y_{1t}$ and $\sum y_{i1t} = Y_{1t}$, it would be impossible to attribute co-variation in *aggregate* media usage to co-usage by individual users.

An MCMC sampler can be developed to sample from the posterior of the model in Eq. 2 by developing a way to sample the missing individual-level y_{i2t} such that they conform to the constraint. Thus the method is closely related to the approach used to estimate choice models from aggregate data proposed by Chen and Yang (2007) and Musalem et al (2008). In fact, in this case, the aggregated data (i.e. constraint) provides information that makes the imputed y_{i2t} even more plausible.

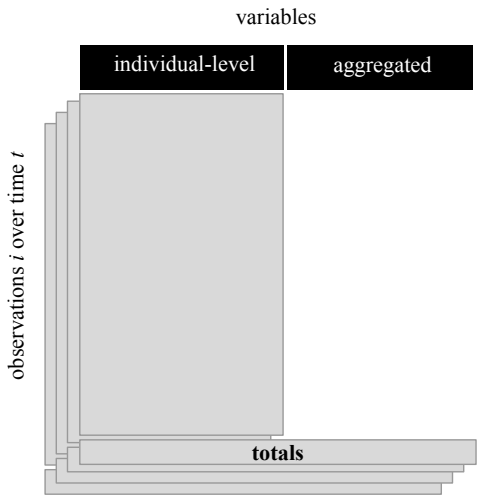

Fig. 2 In mixed aggregate and disaggregate data, only marginal totals are observed for some variables. Repeated observations of the marginal totals make it possible to identify the individual-level correlations, even when they are not directly observed.

2 Developing and estimating fusion models

This section provides readers with a step-by-step guide to developing and estimating fusion models by walking the reader through the computation for two examples. These examples are intentionally simplified to allow the reader to focus on the core ideas in data fusion. Our hope is that readers who master these examples will be well-prepared to move on to the more sophisticated examples we discuss in the literature review in Sec. 3.

2.1 Ex. 1: Fusing data using a multivariate normal model

We begin with an example of data like that in Figure 1, where a vector of K_1 variables, y_{i1}, are only observed in the first data set, while another vector of K_2 variables, y_{i2} are only observed in second data set. As we discussed in the introduction, this is the data structure for split questionnaire designs (where data set 1 and data set 2 represent sub-surveys administered to separate people) and for the classic problem of fusing media consumption data with product purchase data. While creating a complete fused data set (i.e., imputing the missing data in Fig. 1) is often an intermediate step in the analysis, it is important to recognize the ultimate inferential goal in both of these examples is to understand the association between y_{i1} and y_{i2} despite the fact that those variables are never observed together for the same respondent.

The key to making this inference is that there is a vector of K_b variables that are observed in both data sets, y_{ib}. As we will illustrate, it is vital that these linking variables be correlated with y_{i1} and y_{i2}. If they are independent, then the observed data provide no information about the association between y_{i1} and y_{i2}.

The first step in building a fusion model is to specify a likelihood for the complete observation that we wish we had for all respondents $y_i = (y_{i1}, y_{i2}, y_{ib})$. One simple model for a vector response like this is a multivariate normal distribution:

$$y_i \sim N_{(K_1+K_2+K_b)}(\mu, \Sigma) \tag{3}$$

where N_K denotes the multivariate normal distribution of dimension K and μ and Σ are the mean vector and covariance matrix to be estimated from the data. The multivariate normal model is computationally convenient to work with and so is commonly used in the statistical literature (Little and Rubin, 2014; Rässler, 2002). It also allows us to estimate correlations between elements of y_i through the covariance matrix Σ. In data fusion problems, we are particularly interested in the correlations between elements of y_{i1} and y_{i2}, which are never observed for the same subject. For example, when combining media consumption and purchase data, these correlations tell us that users who use a particular media channel are likely to purchase a particular product.

While we begin with the simpler multivariate normal model, we should note that the variables we observe in marketing are often binary or discrete, which may

not be suitably modeled with a multivariate normal model. In most real cases, an appropriate model is chosen such as a latent cut-point model. However, these other models are a relatively straightforward extension of the multivariate model as we will show in Ex. 2.

The core idea of fusion modeling is to estimate the model in (3) using Bayesian methods. Bayesian inference readily handles missing data, including the missing data that is created here due to the fact that some elements of y_i are unobserved for each respondent in the data set. Of course, it would be impossible to follow the approach of estimating the model in (3) using only complete cases, as there are no complete cases.

While our goal is to evaluate the likelihood in Eq. 1, we handle the integral by treating the missing data as unknown and computing the joint posterior of the missing data and the model parameters. In Bayesian inference, all unknowns - parameters, missing data, latent variables - are treated similarly. Conditional on the model and priors, we compute a posterior distribution for each unknown quantity based on Bayes theorem. Once this joint posterior is obtained, the marginal posteriors of the parameter Σ can be evaluated to understand the correlations in the data. The posteriors of the missing data elements in y_i can also be used to generate a fused data set, if that is desired.

It should be emphasized that by fitting a Bayesian fusion model we *simultaneously* obtain estimates for both the parameters of interest – in this case μ and Σ – and impute the missing values in y_i for each respondent, i. Unlike other approaches to missing data which impute in a first stage (often using *ad hoc* methods) and then estimate parameters in a second stage, the posteriors obtained using Bayesian inference use all the information available in the data and reflect all of the posterior uncertainty resulting from the imputing the missing data.

For all but the most simple Bayesian models, the posterior is obtained by developing an algorithm that will generate random draws from the joint posterior distribution of all unknown parameters. These random samples are then analyzed to estimate the posterior distributions for both the model parameters and the missing data. There are a variety of algorithms for sampling from the posterior distribution that can be adapted to any model including the broad class of Markov chain Monte Carlo (MCMC) algorithms. When writing MCMC samplers for a fusion model it is necessary to work out the full conditionals for the missing data and create Gibbs steps to explicitly draw them. An alternative to building the sampler directly is to use a tool like Stan (Carpenter et al, 2016), which allows the user to specify a likelihood using a modeling language, and then automatically produces an MCMC algorithm to generate posterior draws from that model. We will illustrate this example using Stan. This code can be run in R, after the Stan software and the `RStan` R package are installed; see the RStan Getting Started guide (Team', 2016) for installation instructions.

The first step in estimating the fusion model using Stan is to lay out the Stan model code, which describes the data and the likelihood. We provide this code in Figure 3.

```
data {
  int<lower=0> N1;  //observations in data set 1
  int<lower=0> N2;  //observations in data set 2
  int<lower=0> K1;
  int<lower=0> K2;
  int<lower=0> Kb;
  vector[K1] y1[N1];
  vector[K2] y2[N2];
  vector[Kb] yb[N1 + N2];
}
parameters {
  // mean of complete vector
  vector[K1 + K2 + Kb] mu;
  // correlation matrix for complete vector
  corr_matrix[K1 + K2 + Kb] Omega;
  // variance of each variable
  vector<lower=0>[K1 + K2 + Kb] tau;
  // missing elements in data set 1 (observed in y2)
  vector[K2] y1mis[N1];
  // missing elements in data set 2 (observed in y1)
vector[K1] y2mis[N2];
}
transformed parameters{
  // create the complete data
  vector[K1 + K2 + Kb] y[N1 + N2];
  for (n in 1:N1) {
    for (k in 1:K1) y[n][k] = y1[n][k];
    for (k in 1:K2) y[n][K1 + k] = y1mis[n][k];
    for (k in 1:Kb) y[n][K1 + K2 + k] = yb[n][k];
  }
  for (n in 1:N2) {
    for (k in 1:K1) y[N1+n][k] = y2mis[n][k];
    for (k in 1:K2) y[N1+n][K1+k] = y2[n][k];
    for (k in 1:Kb) y[N1+n][K1+K2+k] = yb[N1+n][k];
  }
}
model {
  //priors
  mu ~ normal(0, 100);
  tau ~ cauchy(0,2.5);
  Omega ~ lkj_corr(2);
  //likelihood
  y ~ multi_normal(mu, quad_form_diag(Omega, tau));
}
```

Fig. 3 Stan code for multivariate normal data fusion model.

The `data` block in the code in Fig. 3 tells Stan what data is observed: `N1` observations of a vector of length `K1` called `y1`, `N2` observations of a vector of length `K2` called `y2`, and `N1 + N2` observations of a vector of length `Kb` called `yb`. These correspond to the variables observed only in data set 1, the variables observed only in data set 2 and the common linking variables.

The `parameters` block in Fig. 3 defines the variables for which we want to obtain a posterior. This includes the parameters `mu`, `tau` and `Omega`, where `mu` is the mean vector for y_i, `tau` is a vector of variances and `Omega` is the correlation matrix. (This is the preferred parameterization of the multivariate normal in Stan. Note that other MCMC tools like WinBUGS (Spiegelhalter et al, 2003) parameterize the multivariate normal with a precision matrix, the inverse of the covariance matrix.) The `parameters` block also defines the missing elements of y_i: `y1mis` for the missing variables in data set 1 and `y2mis` for the missing variables in data set 2. In Stan, the term `parameters` is used for any unknown quantity including both traditional parameters and missing data; following the Bayesian approach to inference, Stan makes no distinction between these two types of unknowns.

In the `transformed parameters` block there is a bit of code that maps the observed data and the missing data into the full `y` array. This is simply bookkeeping; the known and unknown elements of `y` from the `data` and `parameters` blocks are mapped into a single vector. (Note that WinBUGS does not require this step and instead simply assumes that any declared data that is not provided is missing data.) Stan requires the user to explicitly declare the observed and missing data and then use the `transformed parameters` block to define the combined "wished for" data.

In the final `model` block, the model is specified, along with priors for the parameters. The complete `y` vector is modeled as a multivariate normal with mean vector `mu` and covariance matrix `quad_form_diag(Omega, tau)`, which transforms `Omega` and `tau` to the covariance matrix Σ. The prior on `mu` is the conjugate normal prior and the priors on `tau` and `Omega` are the cauchy and the LKJ prior for correlation matrices, as recommended by the Stan Modeling Language User's Guide and Reference Manual (Stan Development Team, 2017). Note that Stan provides a wide variety of other possible models and priors and the code in Fig. 3 can be easily modified. More details on how Stan models are specified can be found in the User's Guide.

To estimate the model, the Stan code above is saved in a file and then called using the `stan` function from the `RStan` package in R. (Alternatively, Stan can be called from other languages including Python, MATLAB, Mathematica and Stata.) In R, the data is passed to Stan as an R list object of elements with the same names and dimensions as defined in the `data` block in the Stan model code, i.e., `d1` is a list with `N1`, `N2`, `K1`, `K2`, `Kb`, `y1`, `y2`, and `yb`. For example, if the data is stored in the R object `d1$data`, its structure would be as follows:

```
> str(d1$data)
List of 8
 $ K1: num 1
 $ K2: num 1
```

```
$ Kb: num 2
$ N1: num 100
$ N2: num 100
$ y1: num [1:100, 1] 1.037 -0.798 0.318 -0.322 0.323 ...
$ y2: num [1:100, 1] 0.401 -1.821 -1.701 0.726 -0.228 ...
$ yb: num [1:200, 1:2] 0.607 0.53 1.759 0.49 0.406 ...
```

In this example, the combined vector y_i consists of four variables where the y_{1i} vector is a single variable observed for 100 respondents, y_{2i} is a second single variable observed for a different 100 respondents and y_{bi} consists of two variables observed for all 200 respondents. Complete code to generate synthetic data and run this example is included in the Appendix and is available online at `https://github.com/eleafeit/data_fusion`.

If the code above is saved in the file `Data_Fusion.stan` in the working directory of R, then we can obtain draws from the posterior distribution of all unknowns with the following command in R:

```
library(rstan)
m1 <- stan(file="Data_Fusion.stan", data=d1,
           iter=10000, warmup=2000, chains=1)
```

The result is a set of samples from the posterior distribution for `mu`, `tau`, `Omega` and the missing values of y_i, which are called `y1mis` and `y2mis`. Note the inputs `iter` and `warmup` inputs to Stan, which specify that Stan should throw away the first 2000 draws (`warmup`) and then treat the next 10000 - 2000 = 8000 draws as samples from the posterior. See the chapter in this handbook on Bayesian Models for more detail.

Once this call to Stan from R is completed the posterior draws are stored in the `m1` object in R. Note the computation may take minutes or hours depending on the size of the data set and the speed of the computer; MCMC algorithms tend to be quite computationally intensive. These draws can be analyzed (typically within R) to make statements about the posteriors of the parameters and the missing data. For instance, a summary of the estimated correlations can be produced with the command:

```
> summary(m1, par=c("Omega"))
```

which results in the output:

```
$summary
           stats
parameter          mean           sd        2.5%         25%         50%         75%      97.5%
  Omega[1,1]  1.00000000 0.000000e+00  1.00000000  1.00000000  1.00000000  1.00000000  1.0000000
  Omega[1,2]  0.41684930 2.061013e-01  0.02423465  0.26268973  0.41374668  0.57872494  0.7835050
  Omega[1,3] -0.28081755 7.906684e-02 -0.42740820 -0.33640737 -0.28380772 -0.22730667 -0.1226349
  Omega[1,4]  0.64837176 5.226765e-02  0.53581533  0.61582170  0.65217926  0.68441334  0.7424671
  Omega[2,1]  0.41684930 2.061013e-01  0.02423465  0.26268973  0.41374668  0.57872494  0.7835050
  Omega[2,2]  1.00000000 8.368726e-17  1.00000000  1.00000000  1.00000000  1.00000000  1.0000000
  Omega[2,3] -0.69550156 4.772961e-02 -0.77889871 -0.72881140 -0.69929899 -0.66573197 -0.5916050
...
```

In this example, the primary inferential goal is to understand the correlation between y_{1i} and y_{2i}, which are the first two elements in the vector `y`. This corresponds

to the `Omega[1,2]` correlation reported in the second row of the above summary. The summary shows that the correlation has a posterior mean of 0.417 with a standard deviation of 0.206, suggesting that the correlation is between 0.024 and 0.784, which is fairly diffuse, but clearly suggests a positive correlation between y_{i1} and y_{i2}. While this correlation is the key parameter of interest in the data fusion problem, posterior summaries of other parameters can be obtained with similar commands. See the R code in the Appendix for details.

The posterior distributions for μ and Σ are shown graphically in Figs. 4 and 5 and code to produce these graphs is included in the Appendix. The posterior distribution of the key correlation between y_{i1} and y_{i2} is shown in the bottom of the plot in Fig. 5, labeled `Omega.12`. Again, the figure clearly shows that the posterior of this correlation is quite diffuse. However, despite the fact that we never observe y_{1i} and y_{2i} for the same individual, we can infer that that y_{1i} and y_{2i} are positively

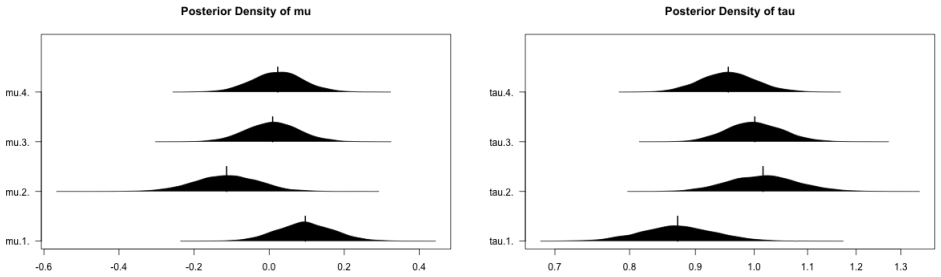

Fig. 4 Posterior distribution of `mu` (means of multivariate normal for y_i) and `tau` (variances of y_i) in Ex. 1.

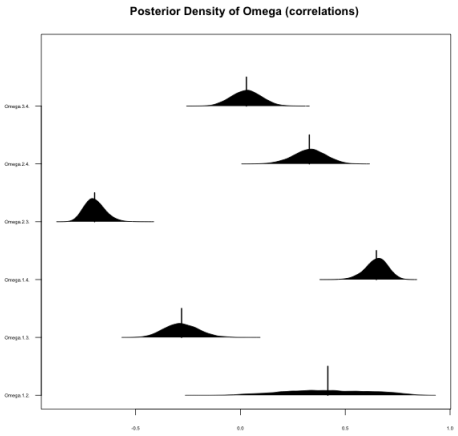

Fig. 5 Posterior distribution of `Omega` (correlations in the multivariate normal for y_i) in Ex. 1.

correlated, which was precisely the goal of our data fusion. Since we generated this data synthetically, we happen to know that the true correlation is 0.3, which the model has recovered reasonably well, despite the fact that y_{1i} and y_{2i} are never observed for the same i and the data set is rather small.

Another important feature to notice in Fig. 5 is that the posterior for the correlation between y_{i1} and y_{i2} (labeled `Omega.12`) is much more diffuse than the other correlations. Since the first two elements of y_i are never directly observed together, the data contains only *indirect* information about the correlation. The other correlations are directly observed and therefore better identified resulting in narrower posteriors for the other correlations in Fig. 5. This can be understood by noting that for the correlations where the variables are never observed together, the MCMC sampler is integrating over possible missing values which creates greater diffuseness (appropriately so) in the posterior.

Although our primary goal is to understand the association between y_{i1} and y_{i2} which can be assessed with the posterior of `Omega.12`, the MCMC sampler also produces posterior samples for the missing elements of y_i. An example of one of these posterior distributions is shown in Fig. 6. Although the overall mean of y_{i1} across all respondents (observed and unobserved) is around 0.1 (see Fig. 4), the posterior for this particular respondent is substantially lower and is centered at -1.22 (2.5%-tile = -2.45, 97.5%-tile = 0.00). Even though we don't observe y_{i1} for this respondent, the posterior for the missing data tells us the likely range of reasonable values of y_{i1} *for this respondent*, based on his or her observed values for y_{i2} and y_{ib}. The posterior of y_{i1} can be summarized by the mean or median to obtain a "best estimate" of the missing data for this respondent. These estimates depend on the observed data for individual i, as well as the estimated mean and covariance across the population.

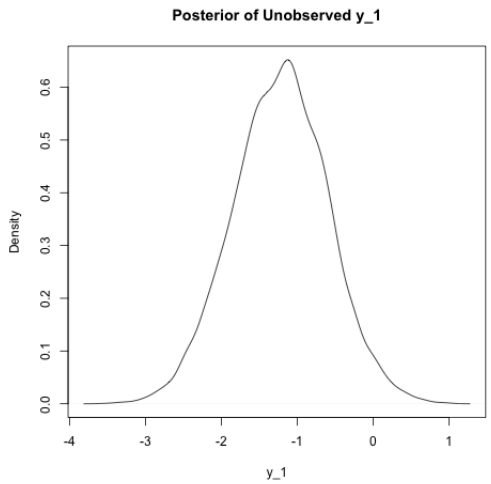

Fig. 6 Posterior distribution for one of the unobserved elements of y

Importantly, because the posterior for the unobserved elements of y_i and the parameters are evaluated simultaneously, the posterior uncertainty in the missing elements of y_i fully accounts for the posterior uncertainty in μ and Σ and, similarly, the posterior uncertainty in μ and Σ fully accounts for posterior uncertainty in the unobserved elements of y_i. That is, we can say that there is a 95% chance that the missing value of y_{i1} for this respondent is between -2.45 and 0.00, conditional on the model and our priors.

The MCMC sampler has produced posterior samples for all 100 missing values of y_{i1} and 100 missing values of y_{i2} and the posterior draws could be summarized to produce a fused data set where the missing values are imputed with posterior means or medians. Although this is unnecessary; if the inferential goal was to measure the correlations in Σ we can interpret the posteriors for Σ directly. If the goal is to produce a fused data set, then to carry forward the posterior uncertainty into any future analysis, the missing values should be multiply imputed (Rubin, 1996), simply by sampling a subset of the posterior draws to create multiple fused data sets. We strongly recommend this approach as opposed to plugging in posterior means or medians (even when appropriately obtained) as biased estimates of non-linear parameters would occur.

Depending on the context, the imputed individual-level data may also be used to target individual customers. For instance, if y_{2i} represents usage of a particular product, then the imputed values of y_{2i} could be used to target specific customers who are likely to use the product, even if we have never observed those customers' product usage. This scoring application is useful in any CRM application where the customers in the data set can be re-targeted, such as fusing the product purchase data between two retailers to identify customers that are good prospects for cross-selling.

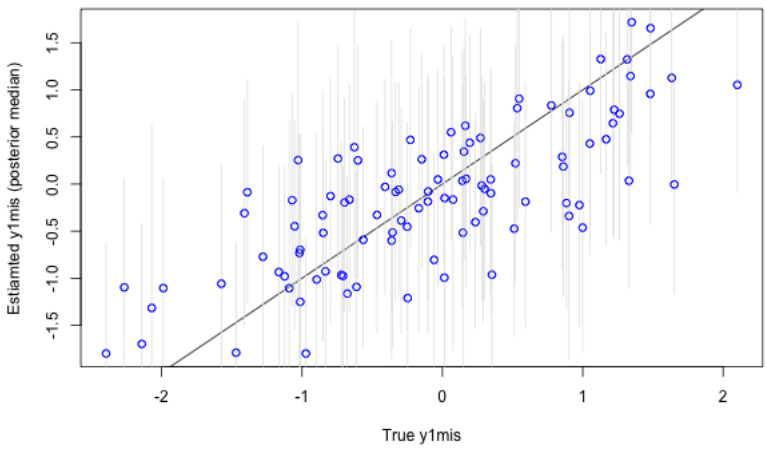

Fig. 7 Posterior estimates of missing elements of y_1 are accurately recovered by the fusion model.

To summarize the overall ability of the model to recover the unobserved values in y_{i1}, we plot the posterior medians for all 100 missing observations of y_{i1} against the true value used to generate the data in Fig. 7. (We know the true values because when we generated the data, we drew y_i from a multivariate normal distribution and then removed the "unobserved" elements of y_i.) Fig. 7 also shows the posterior uncertainty in the imputation by plotting error bars representing the 2.5 and 97.5 %-tiles of the posterior distribution, illustrating the full range of values that the missing data might take. The posterior medians are generally consistent with the true values and the true value is always contained within the posterior interval. Thus, the fusion model is able to accurately recover the unobserved value of y_{i1}.

Fig. 7 shows that the posterior medians for the unobserved y_{1i} tend to be somewhat closer to zero than the true values. (The slope of a best-fit line thorough the points in Fig. 7 is somewhat less than 1.) This is an example of Bayesian shrinkage, where Bayesian posteriors for individuals tend to be closer to the the overall mean and should be expected.

With this example, we have illustrated that fusion modeling is straightforward to execute, however, a word of caution is due. Inference about unobserved values of y_{1i} and y_{2i} and the correlation between them (including non-Bayesian inference) depends critically on there being correlations between the linking variables and y_{1i} and y_{2i}. To illustrate this, we re-estimated the fusion model with two other synthetic data sets that were identical in dimension to the first. One was generated where Σ was diagonal (i.e. no correlations in y_i) and one generated where all correlations in Σ were 0.9. Complete R code for replicating these analyses is included in the Appendix.

When the correlations are high, as shown on the bottom panel in Fig. 8, the missing elements of y_{1i} can be recovered very precisely. The posterior medians are close to the true values and the posteriors are quite narrow, which reflects the fact that when there are high correlations the data is more informative about the unobserved elements of y_{1i} than our first example, where the correlations were moderate. However, when the correlations are zero, the data is completely uninformative of the missing observations of y_{1i}. As shown in the top panel of Fig. 8, there is no discernible relationship between the posterior medians and the true values and the posteriors are so wide that they run off the edges of the plot. This extends to inference about the correlation between y_{i1} and y_{i2}, which has a posterior that is close to uniform between -1 and 1, which is essentially the same as the prior, reflecting that fact that the data contains no information about the correlation between y_{i1} and y_{i2}. So, even with a substantial number of observations, it is possible that the missing elements of y_{i1} and y_{i2} can not be recovered if the linking variables y_{ib} are not correlated with y_{i1} and y_{i2}.

This discussion relates to the clear distinction between the fraction of missing data and fraction of missing information in the missing data literature (Little and Rubin, 2014) where the number of rows that are incomplete may be a poor indicator of how much missing information there is.

The two cases presented in Fig. 8 illustrate why it is extremely important when doing data fusion to carefully choose the linking variables in y_{ib}. For example, in

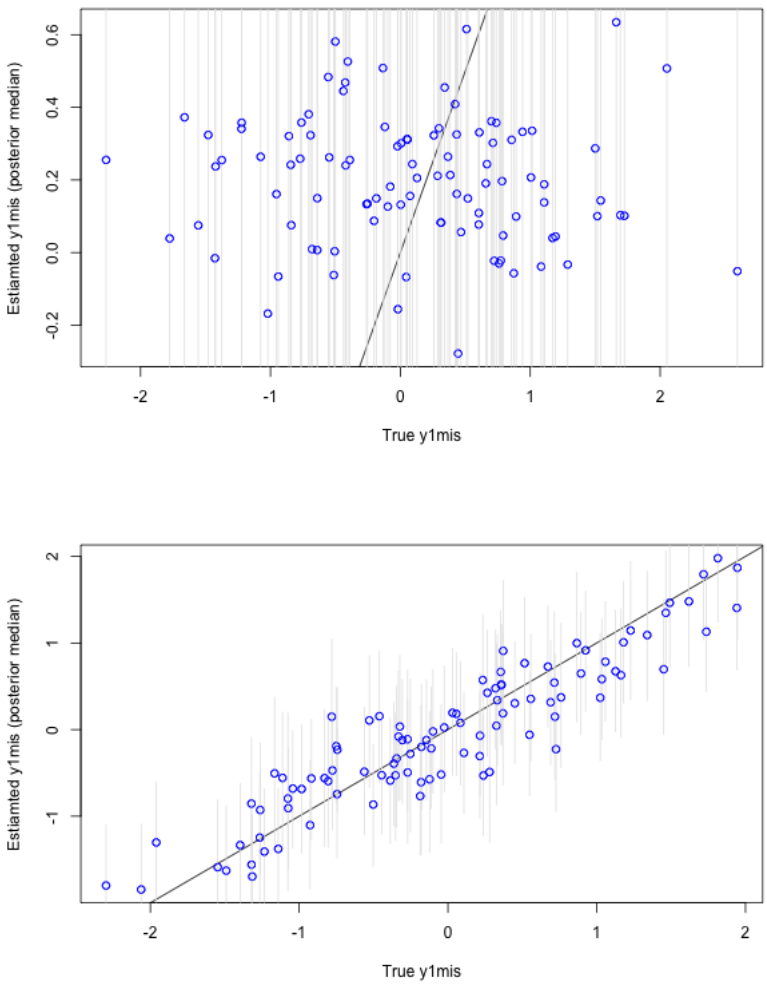

Fig. 8 Inference about missing elements of y_1 depends critically on the correlations between the elements of y. When Σ is diagonal, there is no information in the data about the missing elements of y_{1i} (top). When correlations between elements of y_i are high, unobserved elements of y_{1i} are precisely recovered (bottom).

designing a split questionnaire survey, it is important that responses to the questions that are answered by all respondents are correlated with responses to the questions that are only answered by some respondents. (See Adigüzel and Wedel (2008) for more extensive discussion of split questionnaire design.) In fusing media consumption and product purchase data – the classic data fusion example – demographics are usually used as the linking variables. This will work best when demographics

are correlated with media consumption (which is very likely, but perhaps less so in today's highly-fragmented media landscape) and product purchase (which is likely at the category level, but perhaps not at the brand level).

However, even if a poor choice is made for the linking variables, the Bayesian posteriors for the parameters and the missing data will always reflect whatever uncertainty remains. Thus, unlike ad hoc imputation approaches, Bayesian fusion modeling will identify when the linking variables are weak by reporting diffuse posteriors for the individual-level imputations.

2.2 Ex. 2: Fusing data using a multivariate probit model

As we mentioned earlier, the multivariate normal model is inappropriate for most marketing data, where there are many binary or categorical variables. This is easily accommodated by specifying a latent variable model where an underlying latent vector is normally distributed and then each element of of that vector is appropriately transformed to suit the observed data. For example, if the data is binary, which is quite common in marketing for instance with "check all that apply" type questions in a survey or with behavioral variables that track incidence, one can use a multivariate probit model. Assuming that $y_i = (y_{1i}, y_{2i}, y_{bi})$ contains all binary variables, the model for the complete data is:

$$y_{ik} = \begin{cases} 1 & \text{if } z_{ik} > 0 \\ 0 & \text{if } z_{ik} < 0 \end{cases} \tag{4}$$

$$z_i = (z_{1i}, ..., z_{Ki}) \sim N_K(\mu, \Sigma) \tag{5}$$

where k indexes the elements in y_i from 1 to $K = K_1 + K_2 + K_b$. Complete Stan model code for this model is provided in the Appendix. Note that the variances in Σ are not identified in the multivariate probit model, but associated correlations (`Omega`) are identified.

We estimated this model using a data set with similar structure as that in Ex. 1. As in the previous example, there is one observed variable in the first data set (y_{i1}), one in the second (y_{i2}) and two linking variables in y_{ib}, however, these variables are now all binary. As in the previous example, the key inferential goal is to estimate the correlation between y_{i1} and y_{i2}.

Complete R code for running this model is provided in the Appendix; we focus here on the resulting posterior inference. The posterior distribution for the correlations are shown in Fig. 9 which shows that the correlation between y_{i1} and y_{i2} has a posterior mean of 0.368 with a rather wide posterior relative to our first example (2.5%-tile = 0.024, 97.5%-tile=0.784). This is unsurprising; as in the previous example y_{i1} and y_{i2} are never observed for the same subject and, in addition, the binary data used in this example is less informative than the data used in Ex. 1.

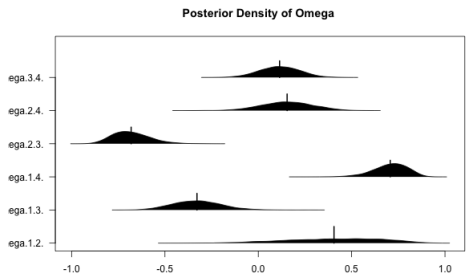

Fig. 9 Posterior distribution of correlations (`Omega`) from a data fusion model for binary data in Ex. 2.

For each of the unobserved y_{1i}, we obtain a set of posterior draws that is either 0 or 1. Summarizing these, we can get a probability that a particular missing value is 1. For example, the first missing element of data set 1 is equal to one in 0.296 of the posterior draws, indicating that there is a 0.296 probability that $y_{1,1}$ is one. Consequently, our best estimate of the missing value of $y_{1,1}$ is that it is equal to zero.

We can summarize these best estimates across all individuals in the data set. Comparing these to the true values that generated the data, we get the following confusion matrix :

Table 1 Confusion matrix for estimated missing values of y_{1i} in fusion model for binary data.

		True Value of y_{1i}	
		0	1
Estimated y_{1i}	0	38	17
	1	14	31

So, even with binary data, which is less informative, it is still possible to estimate a fusion model and recover the unobserved variables from each data set reasonably well.

The multivariate probit sampler also produces draws for the underlying continuous normal variables, z_i. In Fig. 10 we plot the posterior means of those estimated latent variables. Those z's which are associated with an observed binary y are plotted in black and those z for which the y is missing are plotted in red. In Fig. 10 all of the black points are in the upper right or lower left quadrants, reflecting the fact that when the associated binary variable y is observed, the posterior means for z are consistent with the observed y which is in turn consistent with the sign of the true z. In contrast, the red points appear in all four quadrants, reflecting the same "confusion" we saw in Tab 1. However, the points in Fig. 10 do generally follow a diagonal line, reflecting the model's ability to recover the missing values of y_i for most users and the latent z_i.

Exs. 1 and 2 illustrate simple data fusion models for continuous data and binary data. Ex. 2 uses a continuous normal latent variable to model a binary observation and this strategy can be extended to allow for ordinal responses, truncated continu-

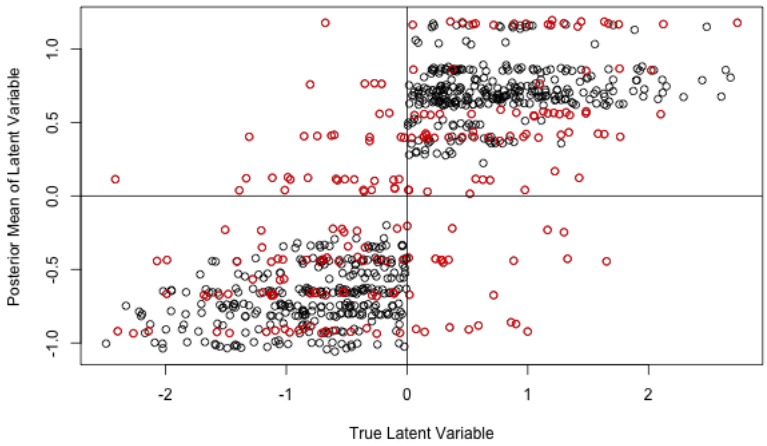

Fig. 10

ous responses or a combination of different variable types. These models can also be extended to allow for mixed levels of aggregation as we discussed in Sec. 1. Additionally, one could build any number of model structures to relate the data in both data sets. The Bayesian framework and tools like Stan allow analysts the flexibility to build models that reflect the data generating process.

2.3 Summary of the process for developing a fusion model

To summarize, the general process for developing a fusion model is as follows:

1. Cast the fusion problem as one of missing data.
2. Consider how the missing data came to be missing. In most data fusion problems, the missing data is *missing by design*, which means we do not need to model the process by which the data became missing as in other missing data settings.
3. Specify a parametric model for the complete "wished-for" or "fused" data.
4. Develop an MCMC sampler for the model. In these examples, we have used Stan which automatically produces a MCMC sampler based on a specified model. Programs similar to Stan include WinBUGS and JAGS. One may also code the sampler directly in a statistical language like R, MATLAB , Python or Gauss.
5. Treat the missing data as unknowns and estimate them using data augmentation. In the above example, we used Stan to define the missing data as a Stan `parameter`, which resulted in Stan producing a posterior sample for the missing data. When building a Gibbs sampler from scratch, one would find a way to

draw the missing parameters from their full conditional distributions based on the model parameters and the observed data.
6. Analyze the posterior samples to make inferences about the model parameters. Often those model parameters correspond directly to the inferential goals of the project.
7. If desired, create a multiply-imputed "fused" data set by taking several random draws from the posterior for the imputed missing data. The fused data can be used as a basis for targeting individual customers.

A point that should be emphasized is that this approach, like all Bayesian inference, conditions on the specified model for the fused data. Our first example used a multivariate normal model and our second used a multivariate probit model. Models based on the multivariate normal are computationally convenient and common in the literature. For instance, in the context of split questionnaires, Raghunathan and Grizzle (1995) and Adigüzel and Wedel (2008) use a cut-point model with an underlying multivariate normal distribution. Rässler (2002) also focuses primarily on data fusion with the multivariate normal. However, as with all model-based inference, a model should be chosen that is appropriate for the data and obtaining a posterior based on that model and the observed data is generally easy to do using modern Bayesian computational methods.

We focused here on methods that propose a model for the joint distribution of the fused data, (y_1, y_2, y_b), but Gilula et al (2006) point out that it is actually only necessary to specify the joint distribution of y_1 and y_2 *conditional* on y_b. They further point out that most of the two-stage matching approaches implicitly assume independence of y_1 and y_2 conditional on y_b, i.e. $f(y_1,y_2|y_b) = f(y_1|y_b)f(y_2|y_b)$. Relying on this assumption, one can specify and estimate models for $f(y_1|y_b)$ and $f(y_2|y_b)$ directly, then integrate over the observations of y_b in the data to find the joint distribution of y_1 and y_2. This simplifies the modeling task, eliminating the need to specify a model for the linking variables, y_b. The likelihood of $f(y_1|y_b)$ and $f(y_2|y_b)$ can be modeled using off-the-shelf methods such as generalized linear models. Qian and Xie (2014) expand on this approach by proposing an alternative non-parametric model for $f(y_1|y_b)$ and $f(y_2|y_b)$ that is highly-flexible and suitable for both continuous and discrete data.

By contrast, the approaches like that illustrated in Exs. 1 and 2 model the full vector (y_1, y_2, y_b) and do not make the assumption of conditional independence directly. Then instead identify the conditional dependence through the prior which yields dependence in the marginal distribution. As we illustrated in Ex. 1, the empirical identification of these the full joint distribution can be weak, that is, some parameters of the joint distribution are only identified by the prior. The level of identification depends, sometimes in subtle ways, on the data; in our example identification was weak when y_{ib} was not correlated with y_{i1} and y_{i2}. Empirical identification should always be checked by comparing the prior to the posterior uncertainty; if they are the same then the data has not provided any information.

3 Summary of related literature

We conclude with a brief summary of the literature in marketing on data fusion and then expand to a number of related papers that use Bayesian missing data methods. Our hope is that the examples provided in the previous section will provide a solid base from which students can tackle the more challenging data fusion problems described in the literature.

3.1 Literature on data fusion

Tab. 2 organizes several key papers on data fusion into three related problem domains: (1) the classic data fusion problem, (2) split questionnaires, (3) mixed aggregate-disaggregate data.

The classic problem of fusing media and purchace data (see Fig. 1) was first recognized by Kamakura and Wedel (1997). They cast the problem as a Bayesian missing data problem, recognizing that the missing data mechanism is missing by design and so is ignorable. They propose a joint model for the fused categorical data (y_{i1}, y_{i2} and y_{ib}) that is a discrete mixture model where incidence is independent across y_i within each latent group. They also show that it is important to account for the uncertainty caused by the data fusion process and propose a multiple imputation approach that is a predecessor to the Bayesian posterior samples we have described in this chapter. Kamakura and Wedel (2000) build on this work by proposing an alternative factor model which can be used in data fusion. They also point out there there are a number other related problems where data is missing by design (including subsampling and time sampling, which we discussed in Sec. 1) where the same Bayesian missing data approach may be employed.

Gilula et al (2006) simplified the data fusion problem by making the assumption of conditional independence between the fused variables. If $p(y_{i1}|y_{ib})$ is assumed to be independent of $p(y_{i2}|y_{ib})$, then it becomes unnecessary to specify the full joint distribution of y_{i1}, y_{i2} and y_{ib}. Instead, the $p(y_1|y_b)$ and $p(y_2|y_b)$ can be estimated separately (using standard models) and then the joint distribution can be approximated by averaging over the observed empirical distribution of y_{ib}, i.e.,

$$p(y_{i1}, y_{i2}|D) \approx E_{\theta|D}\left[\frac{1}{N}\sum_{obs} p(y_{i1}|y_{ib},\theta)p(y_{i2}|y_{ib},\theta)\right] \tag{6}$$

where D is the observed data and N is the number of observations in the complete data set. This so-called direct approach to data fusion reduces the potential for misspecification and is computationally simpler than the joint modeling approach. This approach works well for the standard data fusion problem, yet the joint modeling approach is often desirable when the data fusion problem is embedded withing a more complex model (eg. Musalem et al, 2008; Feit et al, 2010).

Most recently, Qian and Xie (2014) developed a non-parametric odds-ratio model, which they show performs better than the parametric models typical of the prior literature and apply this model using both the direct and the joint modeling approaches. They also identify a new application area for data fusion: combining data collected anonymously on a sensitive behavior with data collected non-anonymously on other behaviors. In their specific application they fuse data on customers use of counterfeit products with other shopping and product attitudes.

Table 2 Summary of key data fusion papers

	Paper	Fusion	Contribution
Media and Purchase	Kamakura and Wedel (1997)	Joint	Recognizing data fusion as a missing data problem and a discrete mixture model for data fusion with categorical variables
	Kamakura and Wedel (2000)	Joint	Factor model for data fusion with continuous and categorical variables
	Gilula et al (2006)	Direct	Direct approach to data fusion applied with several off-the-shelf models
	Qian and Xie (2014)	Direct or Joint	Non-parametric odds ratio model for data fusion with continuous and categorical variables
Split Quest.	Raghunathan and Grizzle (1995)	Joint	Split questionnaire as a missing data problem and a model for continuous and discrete data
	Adigüzel and Wedel (2008)	Joint	Method to *design* a split questionnaire and a normal multivariate cut-point model for data fusion
Agg.	Feit et al (2013)	Joint	Fusion model for mixed aggregate-disaggregate binary data

At about the same time that data fusion was recognized as an important problem in marketing Raghunathan and Grizzle (1995) proposed similar techniques for analyzing split questionnaires in the statistical literature. They propose a model for combined continuous and categorical data and analyze that model with a fully-Bayesian approach, using a Gibbs sampler, as we described in Sec. 2. Adigüzel and Wedel (2008) extend the work on split questionnaire, focusing on the problem of split questionnaire *design*, using a pilot-sample of complete data to determine which questions should be included in each block of the split questionnaire to obtain the most precise posteriors for the missing (by design) data. They use a normal multivariate cut-point model for the data fusion.

Building on this prior work on data fusion, Feit et al (2013) brought the problem of combining mixed aggregate and disaggregate data into the marketing literature. Their approach involves building a posterior sampler for the complete individual-level data that is constrained to be consistent with the aggregate data.

3.2 Related missing data problems

The application of the Bayesian approach to missing data extends far beyond the data fusion problem. Since the reader of this chapter has, by this point, become familiar with the Bayesian approach to missing data, in this section, we provide a brief overview of other applications of this approach. Tab. 3 lists a few key papers in this area.

Table 3 Summary of related work on Bayesian missing data problems

Problem	**Paper**	**Missing Data Mechanism**
Missing regressors	Feit et al (2010)	Ignorable
	Qian and Xie (2011)	Ignorable
Aggregated regressors	Musalem et al (2008)	Ignorable
Survey selection	Bradlow and Zaslavsky (1999)	Non-ignorable
	Ying et al (2006)	Non-ignorable
	Cho et al (2015)	Non-ignorable
Anonymous vists	Novak et al (2015)	Non-ignorable

Both Feit et al (2010) and Qian and Xie (2011) propose solutions to the common problem that the analyst wishes to estimate a regression model, but has some missing regressors. Regressors are not typically included in the probability model and so Feit et al (2010) illustrate how these missing regressors can be handled by including a model specification for them. Inference then proceeds by simulating from the joint posterior for the regression model parameters, regressor model parameters and the missing regressors. Their work illustrates how, under the Bayesian framework, the posterior for missing regressors is informed by the observed regression outcomes. Specifically, they show that you can impute consumers product needs (typically modeled as a regressor) from their observed choices in a conjoint study. While Feit et al (2010) use a standard multivariate probit model for the missing regressors, Qian and Xie (2011) propose a more flexible non-parametric model that can handle a variety of missing regressors. Both papers illustrate the point that researchers should specify a model that reflects their beliefs about the data generating process, whether that model is one of those proposed in the papers in Tab. 2 specifically for data fusion or a regression model or a more complex structural model. Once the model is specified, model parameters and missing data are estimated simultaneously, rather than treating missing data as a problem that should be handled prior to data analysis.

Musalem et al (2008) develop a model and estimation routine for a similar problem where individual-level regressors are missing, but are observed in aggregate. The specific problem they study is the situation where purchase histories are observed for individual customers and those customers are observed redeeming coupons, but we don't know which customers have a coupon that they chose not to redeem. Instead, we only observe how many coupons were distributed in aggregate. They simultaneously estimate a model that relates the (unobserved) coupon

availability to purchases and and imputes "who has the coupon" in a way that is consistent with the aggregate observation.

All the previously discussed literature deals with situations where data is missing by design. That is, the data is missing because the researcher planned not to collect it and and the missingness is therefore not random and is ignorable. But many missing data problems in marketing address situations where the missingness is stochastic and related to the missing value, which is non-ignorable. The classic example of this is survey non-response. Bradlow and Zaslavsky (1999) impute individual users' missing satisfaction ratings under the assumption that a user will be less likely to answer a satisfaction question when they do not hold a strong opinion. Similar, Ying et al (2006) study individual users' movie ratings under the assumption that the likelihood that a user will not rate a movie (probably because they didn't watch it) is related their (unobserved) rating for that movie. Ying et al (2006) illustrate that when the correlation between a movie being not-rated and the likely rating is ignored, predicted ratings are less accurate, leading to less a less effective movie recommendation system. More recently, Cho et al (2015) have revisited missing data in customer satisfaction surveys.

Finally, in a recent application of Bayesian missing data methods, Novak et al (2015) estimate a model of repeat transactions using customer relationship management (CRM) data, which often has the problem that there are a number of visits where the customer is not identified. These transactions may have been made by an existing customer or by a new customer. They show that when there are so-called "anonymous visits" a Bayesian missing data approach can be used to impute the missing user ids and identify the customer who made the anonymous visit.

4 Conclusion

As readers can see, the general problem of missing data in marketing is very broad. The Bayesian framework can be used for missing Ys, missing Xs, data sets where there is individual and aggregate data and so on. In fact, the broad class of missing data and data fusion problems, we would argue, is one of the most prevalent among practitioners today who want to leverage all the data that they have even when disparate data sources can not be directly linked. However, as a final note, we warn again that for those who use these sophisticated methods, one should always pay attention to the mechanism (or hopefully lack thereof) that generated the missing data. If the mechanism is non-ignorable, then then one would have to build a likelihood for the missing data process and that process is often hard to observe and verify. While much research has been done for over 30 years in this area, as new data sets emerge we expect this area to remain one of high activity going forward.

Acknowledgements We would like to thank the many co-authors with whom we have have had discussions while developing and troubleshooting fusion models and other Bayesian missing data methods, especially, Andres Musalem, Fred Feinberg, Pengyuan Wang and Julie Novak.

Appendix

This appendix provides the code used to generate all examples in this chapter. It is also available online at `https://github.com/eleafeit/data_fusion`. Note that the results in the chapter were obtained with Stan 2.17. If you use a different version of Stan, you may obtain slightly different results even when using the same random number seed.

R Code for generating synthetic data and running Ex. 1 with Stan

R commands for Ex. 1 (requires utility functions below to be sourced first)

```
library(MASS)
library(coda)
library(beanplot)
library(rstan)

# Example 1a: MVN ================================================================
# Generate synthetic data
set.seed(20030601)
Sigma <- matrix(c(1, 0.3, -0.2, 0.7, 0.3, 1, -0.6, 0.4,
                  -0.2, -0.6, 1, 0.1, 0.7, 0.4, 0.1, 1), nrow=4)
d1 <- data.mvn.split(K1=1, K2=1, Kb=2, N1=100, N2=100, mu=rep(0,4), Sigma=Sigma)
str(d1$data)
# Call to Stan to generate posterior draws
m1 <- stan(file="Data_Fusion_MVN.stan", data=d1$data,
           iter=10000, warmup=2000, chains=1, seed=12)
# Summaries of posterior draws for population-level parameters
summary(m1, par=c("mu"))
summary(m1, par=c("tau"))
summary(m1, par=c("Omega"))
plot.post.density(m1, pars=c("mu", "tau"), prefix="Ex1",
                  true=list(d1$true$mu, sqrt(diag(d1$true$Sigma)),
                            cov2cor(d1$true$Sigma)))
draws <- As.mcmc.list(m1, pars=c("Omega"))
png(filename="Ex1PostOmega.png", width=600, height=600)
beanplot(data.frame(draws[[1]][,c(2:4, 7:8, 12)]),
         horizontal=TRUE, las=1, what=c(0, 1, 1, 0), side="second",
         main=paste("Posterior Density of Omega (correlations)", log=""),
         cex.axis=0.5)
dev.off()
# Summaries of posterior draws for missing data
summary(extract(m1, par=c("y1mis"))$y1mis[,3,])
png("Ex1y13mis.png")
plot(density(extract(m1, par=c("y1mis"))$y1mis[,3,]),
     main="Posterior of Unobserved y_1", xlab="y_1")
dev.off()
summary(m1, par=c("y"))  # posteriors of observed data place a point mass at the observed value
plot.true.v.est(m1, pars=c("y1mis", "y2mis"), prefix="Ex1",
                true=list(d1$true$y1mis, d1$true$y2mis))

# Example 1b: MVN with zero correlations =========================================
# Generate synthetic data
set.seed(20030601)
Sigma <- matrix(0, nrow=4, ncol=4)
diag(Sigma) <- 1
# Call to Stan to generate posterior draws
d2 <- data.mvn.split(K1=1, K2=1, Kb=2, N1=100, N2=100, mu=rep(0,4), Sigma=Sigma)
m2 <- stan(file="Data_Fusion_MVN.stan", data=d2$data,
```

```
           iter=10000, warmup=2000, chains=1, seed=12)
# Summarize posteriors of population-level parameters
summary(m2, par=c("mu"))
summary(m2, par=c("tau"))
summary(m2, par=c("Omega"))
plot.post.density(m2, pars=c("mu", "tau"), prefix="Ex2",
                  true=list(d1$true$mu, sqrt(diag(d1$true$Sigma)),
                            cov2cor(d1$true$Sigma)))
draws <- As.mcmc.list(m2, pars=c("Omega"))
png(filename="Ex2PostOmega.png", width=600, height=400)
beanplot(data.frame(draws[[1]][,c(2:4, 7:8, 12)]),
         horizontal=TRUE, las=1, what=c(0, 1, 1, 0), side="second",
         main=paste("Posterior Density of Omega", log=""), cex.axis=0.5)
dev.off()
# Summaries of posterior draws for missing data
plot.true.v.est(m2, pars=c("y1mis", "y2mis"), prefix="Ex2",
                true=list(d2$true$y1mis, d2$true$y2mis))

# Example 1c: MVN with strong positive correlations ============================
# Generate synthetic data
set.seed(20030601)
Sigma <- matrix(0.9, nrow=4, ncol=4)
diag(Sigma) <- 1
# Call to Stan to generate posterior draws
d3 <- data.mvn.split(K1=1, K2=1, Kb=2, N1=100, N2=100, mu=rep(0,4), Sigma=Sigma)
m3 <- stan(file="Data_Fusion_MVN.stan", data=d3$data,
           iter=10000, warmup=2000, chains=1, seed=12)
# Summaries of population-level parameters
summary(m3, par=c("mu"))
summary(m3, par=c("tau"))
summary(m3, par=c("Omega"))
plot.post.density(m3, pars=c("mu", "tau"), prefix="Ex3",
                  true=list(d1$true$mu, sqrt(diag(d1$true$Sigma))))
draws <- As.mcmc.list(m3, pars=c("Omega"))
png(filename="Ex3PostOmega.png", width=600, height=400)
beanplot(data.frame(draws[[1]][,c(2:4, 7:8, 12)]),
         horizontal=TRUE, las=1, what=c(0, 1, 1, 0), side="second",
         main=paste("Posterior Density of Omega", log=""))
dev.off()
# Summaries of posterior draws for missing data
plot.true.v.est(m3, pars=c("y1mis", "y2mis"), prefix="Ex3",
                true=list(d3$true$y1mis, d3$true$y2mis))
```

Utility functions for Ex. 1

```
data.mvn.split <- function(K1=2, K2=2, Kb=3, N1=100, N2=100,
                           mu=rep(0, K1+K2+Kb), Sigma=diag(1, K1+K2+Kb))
{
  y <- mvrnorm(n=N1+N2, mu=mu, Sigma=Sigma)
  list(data=list(K1=K1, K2=K2, Kb=Kb, N1=N1, N2=N2,
                 y1=as.matrix(y[1:N1, 1:K1], col=K1),
                 y2=as.matrix(y[N1+1:N2, K1+1:K2], col=K2),
                 yb=as.matrix(y[,K1+K2+1:Kb], col=Kb)),
       true=list(mu=mu, Sigma=Sigma,
                 y1mis=y[1:N1, K1+1:K2],
                 y2mis=y[N1+1:N2, 1:K1]))
}

data.mvp.split <- function(K1=2, K2=2, Kb=3, N1=100, N2=100,
                           mu=rep(0, K1+K2+Kb), Sigma=diag(1, K1+K2+Kb))
{
  z <- mvrnorm(n=N1+N2, mu=mu, Sigma=Sigma)
  y <- z
  y[y>0] <- 1
  y[y<0] <- 0
  y1mis <- y[1:N1, K1+1:K2]
  y2mis <- y[N1+1:N2, 1:K1]
  y[1:N1, K1+1:K2] <- NA
```

```
  y[N1+1:N2, 1:K1] <- NA
  true=list(mu=mu, Sigma=Sigma, z=z, y=y, y1mis=y1mis, y2mis=y2mis)
  y[is.na(y)] <- 0
  data=list(K1=K1, K2=K2, Kb=Kb, N1=N1, N2=N2, y=y)
  list(data=data, true=true)
}

plot.post.density <- function(m.stan, pars, true, prefix=NULL){
  for (i in 1:length(pars)) {
    draws <- As.mcmc.list(m.stan, pars=pars[i])
    if (!is.null(prefix)) {
      filename <- paste(prefix, "Post",  pars[i], ".png", sep="")
      png(filename=filename, width=600, height=400)
    }
    beanplot(data.frame(draws[[1]]),
             horizontal=TRUE, las=1, what=c(0, 1, 1, 0), side="second",
             main=paste("Posterior Density of", pars[[i]]))
    if (!is.null(prefix)) dev.off()
  }
}

plot.true.v.est <- function(m.stan, pars, true, prefix=NULL){
  for (i in 1:length(pars)) {
    draws <- As.mcmc.list(m.stan, pars=pars[i])
    est <- summary(draws)
    if (!is.null(prefix)) {
      filename <- paste(prefix, "TrueVEst", pars[i], ".png", sep="")
      png(filename=filename, width=600, height=400)
    }
    plot(true[[i]], est$quantiles[,3], col="blue",
         xlab=paste("True", pars[i]),
         ylab=paste("Estiamted", pars[i], "(posterior median)"))
    abline(a=0, b=1)
    arrows(true[[i]], est$quantiles[,3], true[[i]], est$quantiles[,1],
           col="gray90", length=0)
    arrows(true[[i]], est$quantiles[,3], true[[i]], est$quantiles[,5],
           col="gray90", length=0)
    points(true[[i]], est$quantiles[,3], col="blue")
    if (!is.null(prefix)) dev.off()
  }
}
```

Stan Model for Ex. 2 (Split Multivariate Probit Data)

```
functions {
  int mysum(int[,] a) {
    int s;
    s = 0;
    for (i in 1:size(a))
      s = s + sum(a[i]);
    return s;
  }
}
data {
  int<lower=0> K1;     // number of vars only observed in data set 1
  int<lower=0> K2;     // number of vars only observed in data set 2
  int<lower=0> Kb;     // number of vars observed in both data sets
  int<lower=0> N1;     // number of observations in data set 1
  int<lower=0> N2;     // number of observations in data set 2
  int<lower=0,upper=2> y[N1+N2, K1+K2+Kb];  // should contain zeros in missing positions
}
transformed data {
  int<lower=1, upper=N1+N2> n_pos[mysum(y)];
```

```
  int<lower=1, upper=K1+K2+Kb> k_pos[size(n_pos)];
  int<lower=1, upper=N1+N2> n_neg[(N1+N2)*(K1+K2+Kb) - K2*N1 - K1*N2 - mysum(y)];
  int<lower=1, upper=K1+K2+Kb> k_neg[size(n_neg)];
  int<lower=0> N_pos;
  int<lower=0> N_neg;
  N_pos = size(n_pos);
  N_neg = size(n_neg);
  {
    int i;
    int j;
    i = 1;
    j = 1;
    for (n in 1:N1) {                       //positions in observed y1
      for (k in 1:K1) {
        if (y[n,k] == 1) {
          n_pos[i] = n;
          k_pos[i] = k;
          i = i + 1;
        } else {
          n_neg[j] = n;
          k_neg[j] = k;
          j = j + 1;
        }
      }
      for (k in (K1+K2+1):(K1+K2+Kb)) {
        if (y[n,k] == 1) {
          n_pos[i] = n;
          k_pos[i] = k;
          i = i + 1;
        } else {
          n_neg[j] = n;
          k_neg[j] = k;
          j = j + 1;
        }
      }
    }
    for (n in (N1+1):(N1+N2)) {               //positions in observed y2
      for (k in (K1+1):(K1+K2+Kb)) {
        if (y[n,k] == 1) {
          n_pos[i] = n;
          k_pos[i] = k;
          i = i + 1;
        } else {
          n_neg[j] = n;
          k_neg[j] = k;
          j = j + 1;
        }
      }
    }
  }
}
parameters {
  vector[K1 + K2 + Kb] mu;
  corr_matrix[K1 + K2 + Kb] Omega;
  vector<lower=0>[N_pos] z_pos;
  vector<upper=0>[N_neg] z_neg;
  vector[K2] z1mis[N1];
  vector[K1] z2mis[N2];
}
transformed parameters{
  vector[K1 + K2 + Kb] z[N1 + N2];
  vector[K2] y1mis[N1];
  vector[K1] y2mis[N2];
  for (i in 1:N_pos)
    z[n_pos[i], k_pos[i]] = z_pos[i];
  for (i in 1:N_neg)
    z[n_neg[i], k_neg[i]] = z_neg[i];
```

```
  for (n in 1:N1) {
    for (k in 1:K2) {
      z[n, K1 + k] = z1mis[n, k];
      if (z1mis[n, k] > 0)
        y1mis[n, k] = 1;
      if (z1mis[n, k] < 0)
        y1mis[n, k] = 0;
    }
  }
  for (n in 1:N2) {
    for (k in 1:K1) {
      z[N1 + n, k] = z2mis[n, k];
      if (z2mis[n, k] > 0)
        y2mis[n, k] = 1;
      if (z2mis[n, k] < 0)
        y2mis[n, k] = 0;
    }
  }
}
model {
  mu ~ normal(0, 3);
  Omega ~ lkj_corr(1);
  z ~ multi_normal(mu, Omega);
}
```

R commands for Ex. 2

```
# Generate synthetic data
set.seed(20030601)
Sigma <- matrix(c(1, 0.3, -0.2, 0.7, 0.3, 1, -0.6, 0.4,
                  -0.2, -0.6, 1, 0.1, 0.7, 0.4, 0.1, 1), nrow=4)
d1 <- data.mvp.split(K1=1, K2=1, Kb=2, N1=100, N2=100, mu=rep(0,4), Sigma=Sigma)
# Call to Stan to generate posterior draws
m1 <- stan(file="Data_Fusion_MVP.stan", data=d1$data,
           iter=10000, warmup=2000, chains=1, seed=35)
# Summaries of posteriors of population-level parameters
summary(m1, par=c("mu", "Omega"))
plot.post.density(m1, pars=c("mu"), prefix="Ex1MVP", true=list(d1$true$mu))
png(filename="Ex1MVPPostOmega.png", width=600, height=400)
draws <- As.mcmc.list(m1, pars=c("Omega"))
beanplot(data.frame(draws[[1]][,c(2:4, 7:8, 12)]),
         horizontal=TRUE, las=1, what=c(0, 1, 1, 0), side="second",
         main=paste("Posterior Density of Omega", log=""))
dev.off()
# Summarize posteriors for one of missing values
y1mis.draws <- extract(m1, par=c("y1mis"))[[1]][,1,1]  # draws for third respondent
mean(y1mis.draws > 0)
# Confusion matrix for missing data
y1mis.est <- summary(m1, par=c("y1mis"))$summary[, "50%"]>0
xtabs(~y1mis.est + (d1$true$y1mis>0))
y2mis.est <- summary(m1, par=c("y1mis"))$summary[, "50%"]>0
xtabs(~y2mis.est + (d1$true$y2mis>0))
z.est <- data.frame(z.true=as.vector(t(d1$true$z)), y=as.vector(t(d1$true$y)),
                    z.postmed=summary(m1, pars=c("z"))$summary[,"50%"])
png(filename="Ex1MVPTrueVEstz.png", width=600, height=400)
plot(z.est[,c(1,3)], xlab="True Latent Variable", ylab="Posterior Mean of Latent Variable")
points(z.est[is.na(z.est$y), c(1,3)], col="red")
abline(h=0, v=0)
dev.off()
```

References

Adigüzel F, Wedel M (2008) Split questionnaire design for massive surveys. Journal of Marketing Research 45(5):608–617

Andridge RR, Little RJ (2010) A review of hot deck imputation for survey non-response. International Statistical Review 78(1):40–64

Bradlow ET, Zaslavsky AM (1999) A hierarchical latent variable model for ordinal data from a customer satisfaction survey with no answer responses. Journal of the American Statistical Association 94(445):43–52

Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker MA, Guo J, Li P, Riddell A (2016) Stan: a probabilistic programming language. Journal of Statistical Software

Chen Y, Yang S (2007) Estimating disaggregate models using aggregate data through augmentation of individual choice. Journal of Marketing Research 44(4):613–621

Cho J, Aribarg A, Manchanda P (2015) The value of measuring customer satisfaction. Available at SSRN 2630898

Feit EM, Beltramo MA, Feinberg FM (2010) Reality check: Combining choice experiments with market data to estimate the importance of product attributes. Management Science 56(5):785–800

Feit EM, Wang P, Bradlow ET, Fader PS (2013) Fusing aggregate and disaggregate data with an application to multiplatform media consumption. Journal of Marketing Research 50(3):348–364

Ford BL (1983) An overview of hot-deck procedures. Incomplete data in sample surveys 2(Part IV):185–207

Gilula Z, McCulloch RE, Rossi PE (2006) A direct approach to data fusion. Journal of Marketing Research 43(1):73–83

Kamakura WA, Wedel M (1997) Statistical data fusion for cross-tabulation. Journal of Marketing Research pp 485–498

Kamakura WA, Wedel M (2000) Factor analysis and missing data. Journal of Marketing Research 37(4):490–498

Little RJ, Rubin DB (2014) Statistical analysis with missing data. John Wiley & Sons

Musalem A, Bradlow ET, Raju JS (2008) Who's got the coupon? estimating consumer preferences and coupon usage from aggregate information. Journal of Marketing Research 45(6):715–730

Novak J, Feit EM, Jensen S, Bradlow E (2015) Bayesian imputation for anonymous visits in crm data. Available at SSRN 2700347

Qian Y, Xie H (2011) No customer left behind: A distribution-free bayesian approach to accounting for missing xs in marketing models. Marketing Science 30(4):717–736

Qian Y, Xie H (2014) Which brand purchasers are lost to counterfeiters? an application of new data fusion approaches. Marketing Science 33(3):437–448

Raghunathan TE, Grizzle JE (1995) A split questionnaire survey design. Journal of the American Statistical Association 90(429):54–63

Rässler S (2002) Statistical matching: A frequentist theory, practical applications, and alternative Bayesian approaches, vol 168. Springer Science & Business Media

Rubin DB (1996) Multiple imputation after 18+ years. Journal of the American Statistical Association 91(434):473–489

Spiegelhalter D, Thomas A, Best N, Lunn D (2003) Winbugs user manual

Stan Development Team (2017) Stan modeling language user's guide and reference manual, version 2.17.0 URL http://mc-stan.org

Tanner MA, Wong WH (1987) The calculation of posterior distributions by data augmentation. Journal of the American Statistical Association 82(398):528–540

Team' SD (2016) Rstan getting started. https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started

Ying Y, Feinberg F, Wedel M (2006) Leveraging missing ratings to improve online recommendation systems. Journal of Marketing Research 43(3):355–365

Index