Estimating the Parameters of Circles and Ellipses Using Orthogonal Distance Regression and Bayesian Errors-in-Variables Regression

Jolene Splett* Amanda Koepke* Felix Jimenez†

Abstract

In ordinary least-squares regression, independent variables are assumed to be known without error. However, in many real-life situations this assumption is not valid. Both orthogonal distance regression and Bayesian errors-in-variables regression can be used to estimate model parameters when there are errors in the dependent and independent variables.

To illustrate the use of the maximum-likelihood and Bayesian approaches, we use both methods to estimate the parameters of a circle. The data used for circle fitting were taken from the cross section of an optical fiber. The shape of optical fibers is important when joining two fibers, so accurate dimensional measurements are critical to minimizing coupling loss. We then compare the results of the two techniques when fitting an ellipse to simulated data.

Circle and ellipse fitting using maximum-likelihood methods have been well documented; however, Bayesian methods for these tasks are less developed. As expected, we found that the Bayesian approach for circle fitting is more intuitive and easier to implement than the maximum-likelihood approach, but generalizing the Bayesian approach to ellipse fitting was surprisingly difficult.

Key Words: Bayesian statistics, errors-in-variables regression, orthogonal distance regression, circle fitting, ellipse fitting

1. Introduction

In many real-life regression problems, errors are present in both the independent and dependent variables. Circles and ellipses are classic examples of data that have errors in both the x and y variables.

We examine two methods of fitting circle and ellipse data. The first method is orthogonal distance regression (ODR). In ODR, maximum-likelihood estimates are obtained by minimizing distances between data points and the fitted curve. The second method is Bayesian errors-in-variables regression (EIV). In general, the Bayesian technique allows for easy and intuitive explicit modeling of errors in both the x and y variables for simple models. In this paper, we seek to extend the Bayesian EIV method for the purpose of estimating the parameters of circles and ellipses. While maximum-likelihood based approaches to the circle-fitting problem are well documented (Boggs et al. [1992], Chernov [2011]), very little has been written about a Bayesian approach to this classic errors-in-variables problem.

Data representing the cross section of an optical fiber (Wang et al. [1997], Mamileti et al. [1992]) are used to demonstrate the ODR and EIV methods for estimating the parameters of a circle. We then use both methods to fit an ellipse to simulated data.

Sections 2 and 3 describe the data being fit and detail our circle fitting and ellipse fitting techniques. Results are provided in Section 4, and Section 5 summarizes our findings.

*National Institute of Standards and Technology, 325 Broadway MC898.03, Boulder, CO 80305
†University of Colorado, Department of Physics, 390 UCB, Boulder, CO 80309

2. Circle Fitting

2.1 Optical Fiber Data

Dimensional measurements of an optical fiber cross section are obtained by circle fitting. Accurate dimensional measurements are critical to providing the best possible transmission of light through the fiber; as much as 1 μm offset in joining two fibers will result in about 5 % loss of signal (Wang et al. [1997]).

The data were obtained by first generating a gray-scale image of a fiber cross section (Figure 1). Next, an edge-detection algorithm was applied to the image to define the outside edge of the fiber, producing the (x, y) coordinate pairs required for the analysis.

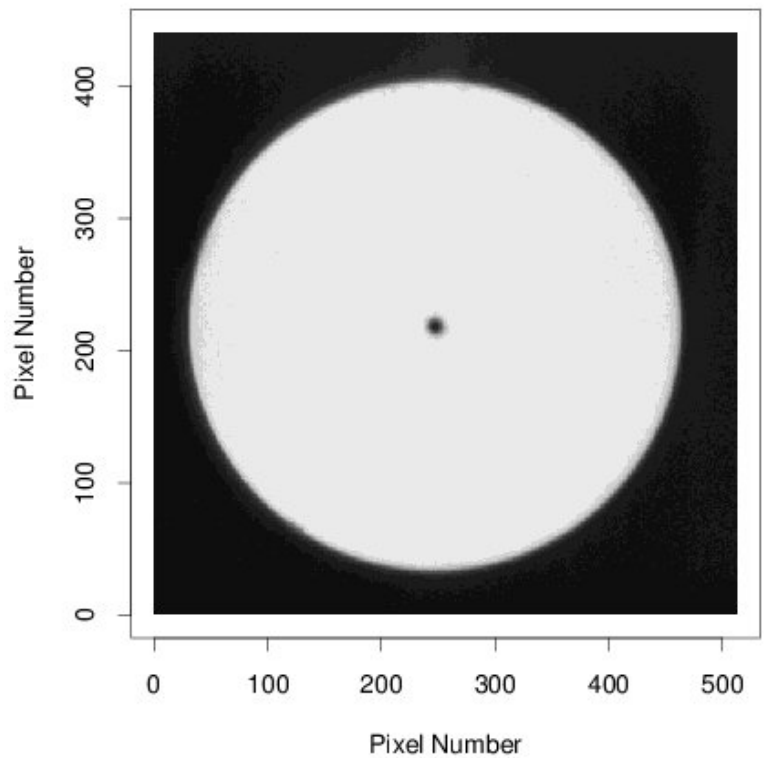


Figure 1: Gray-scale image of an optical-fiber cross section. Data and figure from Wang et al. [1997].

For reasons discussed in 2.2.1, two different representations of a circle are used to perform model fitting. For ODR, we use the general equation of a circle, $r^2 = (x_i - x_c)^2 + (y_i - y_c)^2$. The parameters of interest are the radius (r), the x-coordinate of the center (x_c), and the y-coordinate of the center (y_c).

For EIV, we define a general circle in terms of (x, y) coordinates. Consider a unit circle, centered at $(0, 0)$ with radius $r = 1$. We denote points on the unit circle as $(\tilde{x}_i, \tilde{y}_i)$, where $\tilde{x}_i = \cos(\theta_i)$, $\tilde{y}_i = \sin(\theta_i)$, and $\theta_i \in (-\pi, \pi]$. From the unit circle, we can obtain points on a standard circle (x'_i, y'_i) by multiplying by a constant r, so $x'_i = r\cos(\theta_i)$ and $y'_i = r\sin(\theta_i)$. This standard circle is centered at zero and has radius r. We can move this circle to a new center (x_c, y_c), which gives us the general form for points on a circle

$$(X_i = x_c + r\cos(\theta_i), \;\; Y_i = y_c + r\sin(\theta_i)) \,. \tag{1}$$

2.2 Bayesian Estimates

2.2.1 The model

To obtain parameter estimates in the frequentist framework, the sum-of-squared residuals is minimized subject to the constraint from the general equation of a circle; creating a similar framework for Bayesian EIV regression is not obvious. Keksel et al. [2018] modeled a circle using Bayesian methods by defining their likelihood using $(x_i - x_c)^2 + (y_i - y_c)^2 - r^2$, assigning this a normal distribution with zero mean and some variance, but the approach does not allow the errors in the x and y variables to be modeled separately. The method used by Werman and Keren [2001] also has this limitation. Since the ODR circle parameterization makes it difficult to determine the likelihood for the Bayesian EIV regression approach, we propose a more intuitive parameterization, rewriting the model in terms of x and y coordinates.

We use the general form of a circle (1) to model our observed data. First, for notational and computational convenience, we roughly center our observed circle by subtracting the means, so our new data are represented as $x_{i,new} = x_i - \bar{x}$ and $y_{i,new} = y_i - \bar{y}$, where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. The center of the new circle is then $x_0 = x_c - \bar{x}$ and $y_0 = y_c - \bar{y}$, where (x_c, y_c) is the center of the observed circle data as defined in Section 2.1. Thus for the EIV model, (1) becomes

$$(X_i^* = x_0 + r\cos(\theta_i), \;\; Y_i^* = y_0 + r\sin(\theta_i)) \,. \tag{2}$$

We assume $x_{i,new}$ and $y_{i,new}$ are normally distributed about the points of some true circle defined by (2) with variances σ_x^2 and σ_y^2, so

$$x_{i,new} \sim N(X_i^*, \sigma_x^2) \qquad y_{i,new} \sim N(Y_i^*, \sigma_y^2). \tag{3}$$

Now we must assign prior distributions to all of the parameters in our model. We know that the deviations of the points about the circle are small, encouraging us to use relatively concentrated prior distributions for σ_x and σ_y. We assume tight gamma prior distributions for σ_x and σ_y,

$$\sigma_x \sim \text{Gamma}(\text{shape} = 2, \text{rate} = 50) \qquad \sigma_y \sim \text{Gamma}(\text{shape} = 2, \text{rate} = 50).$$

Since we center our data, x_0 and y_0 should be close to zero, so we assume *a priori*

$$x_0 \sim N(0, 1) \qquad y_0 \sim N(0, 1).$$

The radius is required to be positive, and we assume *a priori*

$$r \sim N(60, 10^2) I_{r \geq 0}.$$

This prior distribution is centered about a realistic value of r, but has large variance.

We use a von Mises prior distribution for the θ_i, bounded between $-\pi$ and π, with location equal to zero and concentration, κ, equal to 0.1. This distribution is plotted for various concentration values in Figure 2. The concentration determines how concentrated the distribution is about the location. The distribution becomes uniform over the interval as κ goes to zero. However, in practice the concentration is restricted to be greater than zero for computational reasons, so we give κ a small value. The von Mises distribution is a common circular distribution that has desirable inferential properties (Mardia and Jupp [2000]).

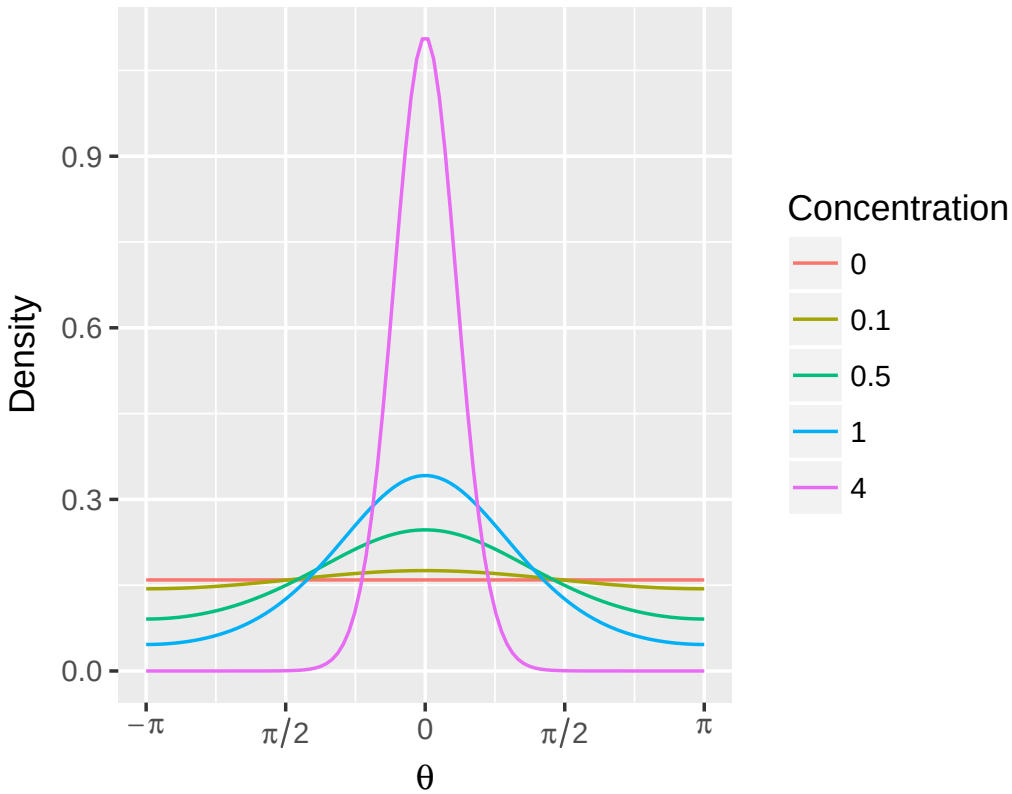


Figure 2: Probability density functions for von Mises distributions on θ, bounded between $-\pi$ and π, with different values for the concentration. As the concentration approaches zero, the distribution becomes more uniform.

2.2.2 MCMC

Now that a model, a set of priors, and the observed data have been defined, we turn our attention to estimating the parameters. To accomplish this, we use Hamiltonian Monte Carlo (HMC), implemented via `Stan` using the `R` package `rstan` [Carpenter et al., 2017, Stan Development Team, 2018], to sample from the posterior distribution of the parameters given the observed data. In particular, we run three chains, each with a burn in of 500 samples and a total of 1000 iterations.

The large number of parameters make this model difficult to fit without reasonable initial values for HMC. We initialize all three chains as follows. The initial radius value is $r_{init} = 60$, the standard deviations (σ_x, σ_y) are both initialized at 0.01, and the center is $(x_{0,init}, y_{0,init}) = (0, 0)$. To initialize the θ_i, we relate them to the data $(x_{i,new}, y_{i,new})$ and the other initial values, so

$$\theta_{i,init} = \arctan\left(\frac{\tilde{y_i}}{\tilde{x_i}}\right) = \arctan\left(\frac{(y_{i,new} - y_{0,init})/r_{init}}{(x_{i,new} - x_{0,init})/r_{init}}\right),$$

which reduces to $\theta_{i,init} = \arctan\left(\frac{y_{i,new}}{x_{i,new}}\right)$.

2.3 Maximum-Likelihood Estimates

We obtain maximum-likelihood estimates of the parameters for the circle model using the Fortran package, ODRPACK (Boggs et al. [1992]). The model for circle fitting,

$$0 = (x_i - x_c)^2 + (y_i - y_c)^2 - r^2, \tag{4}$$

is an implicit model because there is no explicit independent variable. For implicit multivariate orthogonal distance regression using ODRPACK, define $z_i = (x_i, y_i)$ and $\delta_i = (\delta_{x_i}, \delta_{y_i})$, where δ_{x_i} and δ_{y_i} represent the errors in x_i and y_i, respectively. Parameters are estimated using

$$\min_{\beta,\delta} \sum_{i=1}^{n} w_{\delta_i} {\delta_i}^2 \tag{5}$$

subject to the constraint

$$f_i(z_i + \delta_i; \beta) = ((x_i - \delta_{x_i}) - x_c)^2 + ((y_i - \delta_{y_i}) - y_c)^2 - r^2 = 0, \qquad i = 1, \ldots, n,$$

where w_{δ_i} are weights and $\beta = (x_c,\, y_c,\, r)$. For our problem, $w_{\delta_i} = 1$ for all i.

3. Ellipse Fitting

3.1 Simulated Data

To understand the models for fitting the ellipse, we review some geometry. An ellipse can be defined in terms of two circles with radii r_x and r_y, with $r_x > r_y$. This is depicted in Figure 3. As with the circle model, we first consider the unit circle (green circle in the figure). In the figure, A denotes a point on the unit circle, and $A = (\tilde{x_i}, \tilde{y_i}) = (\cos(\theta_i), \sin(\theta_i))$, where $\theta_i \in [0, 2\pi)$. Using points on the unit circle, we obtain points on the red circle by multiplying by r_y, so point B in the figure equals $(r_y\cos(\theta_i), r_y\sin(\theta_i))$. Similarly, we obtain points on the blue circle by multiplying by r_x, so point C in the figure equals $(r_x\cos(\theta_i), r_x\sin(\theta_i))$. Using these two circles, we obtain a point (D) on the standard ellipse as $(x_i', y_i') = (r_x\cos(\theta_i), r_y\sin(\theta_i))$. The standard ellipse is centered at zero, has major axis r_x, and has minor axis r_y.

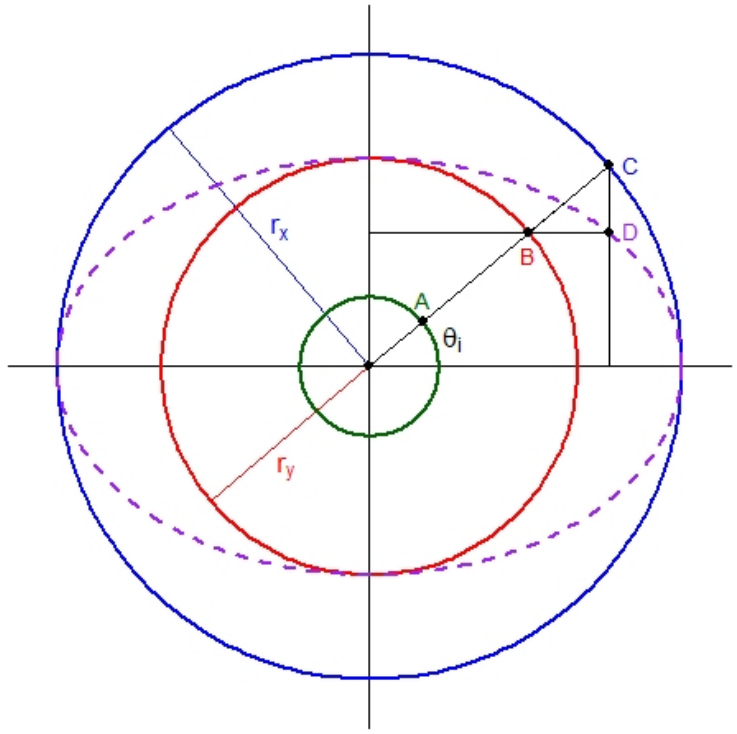


Figure 3: Geometric interpretation of an ellipse based on two circles with different radii.

The standard ellipse can be rotated by angle α and moved to a new center (x_c, y_c), which gives us the general form for points on an ellipse

$$\begin{aligned} X_i &= x_c + r_x \cdot \cos(\theta_i) \cdot \cos(\alpha) - r_y \cdot \sin(\theta_i) \cdot \sin(\alpha) \\ Y_i &= y_c + r_y \cdot \sin(\theta_i) \cdot \cos(\alpha) + r_x \cdot \cos(\theta_i) \cdot \sin(\alpha). \end{aligned} \tag{6}$$

For ellipse fitting, we simulate 1000 noisy data pairs as $x_i \sim N(X_i, \sigma_x^2)$ and $y_i \sim N(Y_i, \sigma_y^2)$ using the values shown in Table 1. The $\theta_1, \ldots, \theta_{1000}$ are an evenly spaced sequence from $-\pi + \epsilon$ to $\pi - \epsilon$, with $\epsilon = 0.00001$. Figure 4 displays the data points used in the analysis.

Table 1: Parameters used to generate simulated ellipse data.

Parameter	Value
x_c	71
y_c	74
r_x	62
r_y	50
α	-0.55 rad
σ_x	0.05
σ_y	0.07

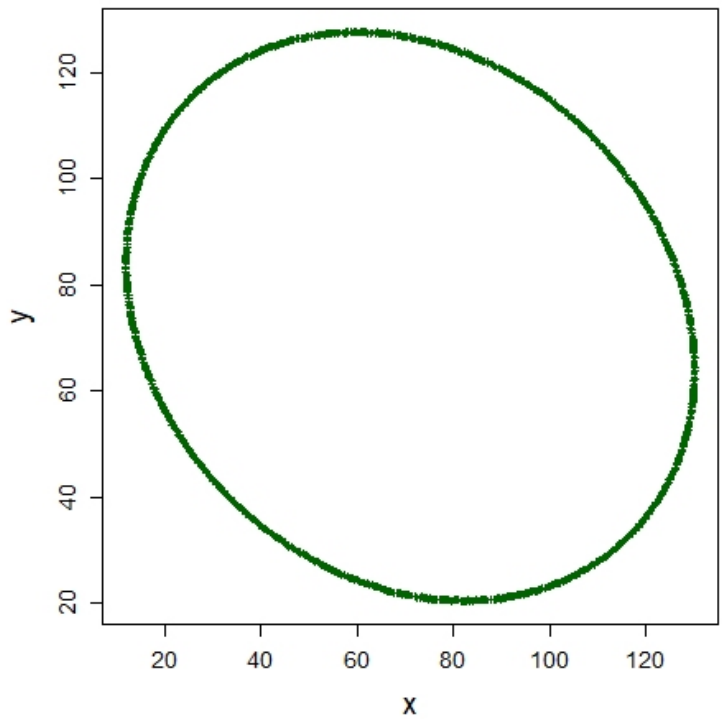


Figure 4: Simulated ellipse data (n=1000) generated using (6) and the parameters listed in Table 1.

3.2 Bayesian Estimates

3.2.1 The Model

As before, we use the general form of the ellipse (6) to model our observed data. We again roughly center our observed ellipse by subtracting the means, so our new data are represented as $x_{i,new} = x_i - \bar{x}$ and $y_{i,new} = y_i - \bar{y}$, and the center of the new ellipse is $x_0 = x_c - \bar{x}$ and $y_0 = y_c - \bar{y}$, where (x_c, y_c) is the center of the observed ellipse data as defined in Section 3.1. Thus for the EIV model, (6) becomes

$$\begin{aligned} X_i^* &= x_0 + r_x \cdot \cos(\theta_i) \cdot \cos(\alpha) - r_y \cdot \sin(\theta_i) \cdot \sin(\alpha) \\ Y_i^* &= y_0 + r_y \cdot \sin(\theta_i) \cdot \cos(\alpha) + r_x \cdot \cos(\theta_i) \cdot \sin(\alpha). \end{aligned} \tag{7}$$

As with the circle model, we assume $x_{i,new}$ and $y_{i,new}$ are normally distributed about the points of some true ellipse defined by X_i^* and Y_i^*, so

$$x_{i,new} \sim N(X_i^*, \sigma_x^2) \qquad y_{i,new} \sim N(Y_i^*, \sigma_y^2). \tag{8}$$

We use many of the same prior distributions assumed in Section 2.2. Since we center our data, we assume *a priori* that the center of our ellipse should be close to (0,0), so

$$x_0 \sim N(0, 1) \qquad y_0 \sim N(0, 1).$$

The standard deviations of x_i and y_i, both of which are positive, are defined by

$$\sigma_x \sim \text{Gamma}(\text{shape} = 2, \text{rate} = 50) \qquad \sigma_y \sim \text{Gamma}(\text{shape} = 2, \text{rate} = 50).$$

Both the x radius (r_x) and y radius (r_y) must be positive. We assume

$$r_x \sim N(60, 10^2)I_{[r_x>0]} \qquad r_y \sim N(60, 10^2)I_{[r_y>0]}.$$

Again, this prior distribution expresses a lot of uncertainty about these parameter values, even though we think they should be around 60. The larger of the r_x and r_x values corresponds to the major axis (M) and the smaller value corresponds to the minor axis (m).

The rotation of the ellipse, α, is constrained to cover a 90 degree range to ensure a unique solution for the parameter estimates. This restriction on α allows us to avoid imposing a restriction on the relative sizes of r_x and r_y due to the fact that the major axis must be greater than the minor axis. If α covered an entire 180 degree range, then two different values of α could be used to describe the same ellipse if the major and minor axes are switched. *A priori* we assume α has a uniform distribution between $-\pi/2$ and zero.

Finally, we again use a von Mises prior distribution for the θ_i, bounded between $-\pi$ and π, with location equal to zero and concentration equal to 0.1.

3.2.2 MCMC

Our results are very sensitive to initial values, particularly for the angle of rotation. Thus, we use the data to estimate a rough angle of rotation to initialize the model. Using the roughly centered data $(x_{i,new}, y_{i,new})$, we calculate $r_i = \sqrt{x_{i,new}^2 + y_{i,new}^2}$. The maximum r_i should correspond to a point close to the major axis, so the angle that corresponds to that point, constrained to be between $-\pi$ and zero, is a good initial value for the angle of rotation of our ellipse. Specifically, $\alpha_{init} = \arctan\left(\frac{y_{j,new}}{x_{j,new}}\right)$, where j is the index for the maximum value of the r_i. If this is not between $-\pi/2$ and zero, we find the corresponding angle that is in this range.

For the center, we use $(x_{0,init}, y_{0,init}) = (0, 0)$. Given $(x_{i,new}, y_{i,new})$, we obtain a standard ellipse roughly centered at $(0, 0)$ and rotated by angle, $-\alpha_{init}$, using

$$\begin{aligned} x_i' &= (x_{i,new} - x_{0,init}) \cdot \cos(-\alpha_{init}) - (y_{i,new} - y_{0,init}) \cdot \sin(-\alpha_{init}) \\ y_i' &= (y_{i,new} - y_{0,init}) \cdot \cos(-\alpha_{init}) + (x_{i,new} - x_{0,init}) \cdot \sin(-\alpha_{init}). \end{aligned} \tag{9}$$

From the standard ellipse (9), we can initialize r_x and r_y as

$$r_{x,init} = \frac{|\min(x_i')| + |\max(x_i')|}{2}$$
$$r_{y,init} = \frac{|\min(y_i')| + |\max(y_i')|}{2}.$$

To initialize the θ_i, we set

$$\theta_{i,init} = \arctan\left(\frac{y_i'/r_{y,init}}{x_i'/r_{x,init}}\right).$$

The standard deviations σ_x and σ_y are both initialized at 0.01. We use these values to initialize three chains, each with a burn in of 300 and a total of 600 iterations, again using HMC to sample from the posterior distribution.

3.3 Maximum-Likelihood Estimates

A general formula for an ellipse that is translated and rotated in the x, y plane (Fuller [1987]) is

$$\beta_1(y_i - y_c)^2 + 2\beta_2(y_i - y_c)(x_i - x_c) + \beta_3(x_i - x_c)^2 - 1 = 0. \tag{10}$$

The center of the ellipse is $(x_c,\ y_c)$; however, the remaining parameters, $(\beta_1,\ \beta_2,\ \beta_3)$, have no geometric interpretation. Thus, we compute the angle of rotation, the major axis, and minor axis, $(\alpha,\ M,\ m)$, from the estimated model parameters $(\hat{x}_c,\ \hat{y}_c,\ \hat{\beta}_1,\ \hat{\beta}_2,\ \hat{\beta}_3)$.

We use ODRPACK to obtain the maximum-likelihood estimates of the ellipse model parameters by the same procedure described in Section 2.3. As in circle fitting for an implicit model (10), ODRPACK minimizes (5) subject to

$$\begin{aligned} g_i(z_i + \delta_i; \beta) &= \beta_1((y_i - \delta_{y_i}) - y_c)^2 + 2\beta_2((y_i - \delta_{y_i}) - y_c)((x_i - \delta_{x_i}) - x_c) \\ &+ \beta_3((x_i - \delta_{x_i}) - x_c)^2 - 1 = 0, \quad i = 1, \ldots, n, \end{aligned}$$

where $z_i = (x_i,\ y_i)$ and $\beta = (x_c,\ y_c,\ \beta_1,\ \beta_2,\ \beta_3)$.

The standard errors of $(\hat{x}_c,\ \hat{y}_c)$ are obtained directly from ODRPACK. However, the standard errors of the additional parameters of interest, $(\hat{\alpha},\ \hat{M},\ \hat{m})$, are computed from Monte Carlo simulations (Lafarge and Possolo [2015]) based on $\hat{\beta}$ and the estimated variance-covariance matrix of $\hat{\beta}$.

4. Results

Parameter and interval estimates determined by maximum-likelihood and Bayesian methods are shown in Tables 2 and 3 for the circle and ellipse models, respectively. There is very little difference between estimated parameters for maximum-likelihood and Bayesian methods for both circle and ellipse models. Plots of the parameter estimates and their 95 % confidence intervals and 95 % credible intervals are shown in Figures 5 and 6.

Table 2: Circle parameters estimated by ODR and Bayesian EIV.

Parameter	ODR Estimate (95 % Confidence Interval)	Bayesian EIV Estimate (95 % Credible Interval)
x_0, μm	71.89 (71.89, 71.90)	71.90 (71.89, 71.90)
y_0, μm	74.26 (74.26, 74.27)	74.27 (74.26, 74.27)
r, μm	62.47 (62.47, 62.47)	62.48 (62.47, 62.48)

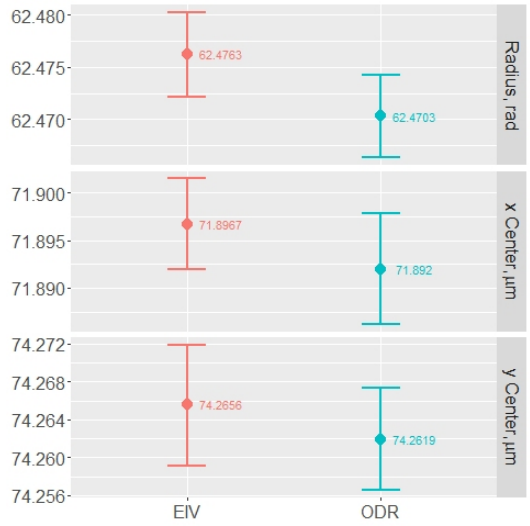
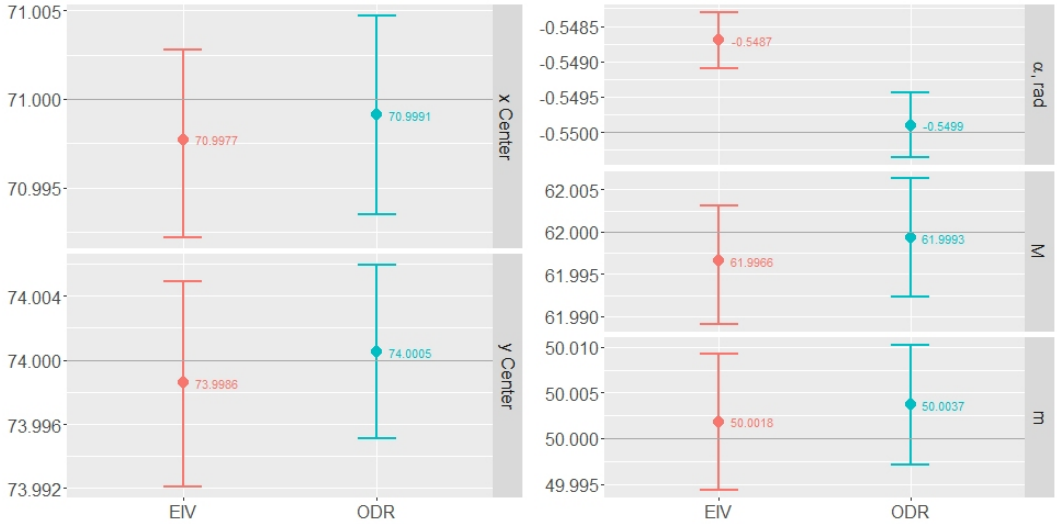


Figure 5: Bayesian and maximum-likelihood interval estimates for circle parameters.

Table 3: Ellipse parameters estimated by ODR and Bayesian EIV.

Parameter	ODR Estimate (95 % Confidence Interval)	Bayesian EIV Estimate (95 % Credible Interval)
x_0	70.9991 (70.9935, 71.0047)	70.9977 (70.9922, 71.0028)
y_0	74.0005 (73.9952, 74.0058)	73.9986 (73.9921, 74.0049)
α, rad	-0.5499 (-0.5505, -0.5496)	-0.5487 (-0.5491, -0.5483)
M	61.9993 (61.9924, 62.0062)	61.9966 (61.9891, 62.0031)
m	50.0037 (49.9973, 50.0102)	50.0018 (49.9943, 50.0093)



(a) (b)

Figure 6: Bayesian and maximum-likelihood interval estimates for ellipse parameters. The horizontal gray reference lines indicate the true parameter values.

Figure 7 displays a portion of the measurement data and the circle fitting results for both methods. Plotted on the scale of the raw data, the two methods produce nearly identical circles. The plot shows how the raw data deviates from a circle due to imperfections in the optical fiber.

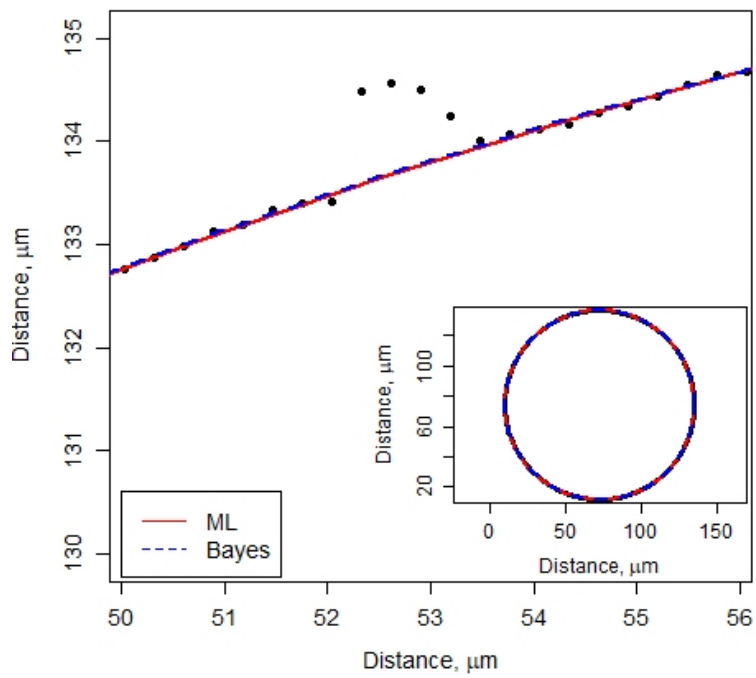


Figure 7: Maximum likelihood and Bayesian circle fitting results.

5. Discussion

We fit a circle model to a real data set and an ellipse model to a simulated data set using both maximum-likelihood and Bayesian approaches. The two methods produced similar parameter and interval estimates. The Bayesian approach is more intuitive for the error-in-variables problem than the maximum-likelihood method. Although we expected the Bayesian approach to be easier to implement than the maximum-likelihood method, we encountered many issues in practice when fitting an ellipse model to the simulated data using Bayesian EIV regression. For instance, the credible interval for the angle of rotation, α, does not cover the true value.

The two methods were further tested when we fit an ellipse to the optical fiber data. The maximum-likelihood method was able to produce parameter estimates, but the Bayesian EIV regression approach could not due to lack of convergence of the chains sampled using HMC. However, parameter estimates were obtained using Bayesian EIV regression when the angle of rotation was fixed at -0.55 rad. The failure of the Bayesian method stems from an inability to estimate α given the current model framework. The angle of rotation and the angles θ_i that correspond to each data point are likely not separable in this model formulation, especially when the data are very circular as is the case for the optical fiber data. Thus, a different model formulation is likely required. The maximum-likelihood method proceeds in a different fashion, minimizing the objective function subject to the

constraint of the circle model. Imposing a constraint in an intuitive way, while maintaining the error-in-variables structure of the problem, seems much more difficult in a Bayesian framework. More work is needed to develop new models for Bayesian EIV regression.

Another assumption that should be examined in future Bayesian models is the independence of the errors in the x and y values. For most Bayesian EIV regression problems, this assumption makes sense, but for optical fiber data these errors are likely correlated. It is plausible that a large error in the x coordinate suggests a deformation in the fiber at that point, and would likely have a corresponding y value with a large error as well. The model for the data will depend on how the data are acquired. Future work will include formulating a more accurate model of the error structure for our data. Additionally, in the future we will try using more prior distributions that do not assume independence between the center, radius, and angles.

A. Code

A.1 Circle

A.1.1 R code

```
library(rstan)

df = read.table("J1-1E.DAT")

niters=1000
n.chains = 3
in.dat = list(
  N=dim(df)[1],
  x=df$V1-mean(df$V1),
  y=df$V2-mean(df$V2)
)

x0init=0
y0init=0
truexinit = (in.dat$x-x0init)
trueyinit = (in.dat$y-y0init)

angle = atan2(trueyinit,truexinit)

inits = list(
  list(
    r = 60,
    x0 = x0init,
    y0 = y0init,
    sdy = .01,
    sdx = .01,
    theta = atan2(trueyinit,truexinit)
  ),
  list(
    r = 60,
    x0 = x0init,
    y0 = y0init,
    sdy = .01,
    sdx = .01,
    theta = atan2(trueyinit,truexinit)
  ),
```

```
  list(
    r = 60,
    x0 = x0init,
    y0 = y0init,
    sdy = .01,
    sdx = .01,
    theta = atan2(trueyinit,truexinit)
  )
)


fit = stan(
  file = "circle.stan",
  data = in.dat,
  init = inits,
  iter = niters,
  warmup = floor(niters/2),
  chains = n.chains,
  control = list(adapt_delta = .8)
)
```

A.1.2 Stan code

```
data {
  int<lower=0> N;
  vector[N] x;
  vector[N] y;
}

parameters {
  real<lower=0> sdy;
  real<lower=0> sdx;

  real x0;
  real y0;

  real<lower = 0> r;

  vector<lower=-pi(), upper = pi()>[N] theta;
}

transformed parameters {
  vector[N] x_tran;
  vector[N] y_tran;

  for(i in 1:N){
    x_tran[i] = x0 + r * cos(theta[i]);
    y_tran[i] = y0 + r * sin(theta[i]);
  }
}

model {
  r ~ normal(60, 10);

  x0 ~ normal(0,1);
  y0 ~ normal(0,1);
```

```
  sdy ~ gamma(2,50);
  sdx ~ gamma(2,50);

  theta ~ von_mises(0,0.1);

  x~normal(x_tran,sdx);
  y~normal(y_tran,sdy);
}
```

A.2 Ellipse

A.2.1 R code

```
library(rstan)

######################### Simulate the data
set.seed(4)

n=1000

x0 = 71
y0 = 74
alphasim = -.55 #couterclockwise rotation
rx= 62
ry = 50

theta = seq(-pi+.00001,pi-.00001,length.out = n)

Xi = x0 + rx*cos(theta)*cos(alphasim) - ry*sin(theta)*sin(alphasim)

Yi = y0 + ry*sin(theta)*cos(alphasim) + rx*cos(theta)*sin(alphasim)

df=data.frame(V1=numeric(n))

sdx_true = .05
sdy_true = .07
df$V1=rnorm(length(Xi),Xi,sdx_true)
df$V2=rnorm(length(Yi),Yi,sdy_true)

######################### Stan settings
niters=600
n.chains = 3
in.dat = list(
  N=dim(df)[1],
  x=df$V1-mean(df$V1),
  y=df$V2-mean(df$V2)
)

# Initial values
r=sqrt(in.dat$x^2+in.dat$y^2)
alphainit=atan2(in.dat$y[which.max(r)],in.dat$x[which.max(r)])
possibleAlphas=c(alphainit,alphainit+pi/2,alphainit+pi,
                 alphainit-pi/2,alphainit-pi)

myAlphaInit=possibleAlphas[possibleAlphas<0 & possibleAlphas>-pi/2 ]
```

```
truexinit = (in.dat$x-0)*cos(-myAlphaInit)-(in.dat$y-0)*sin(-myAlphaInit)
trueyinit = (in.dat$y-0)*cos(-myAlphaInit)+(in.dat$x-0)*sin(-myAlphaInit)

rxinit=sum(abs(range(truexinit)))/2
ryinit=sum(abs(range(trueyinit)))/2

angle=atan2(rxinit*trueyinit,ryinit*truexinit)

inits = list(
  list(
    rx = rxinit,
    ry = ryinit,
    x0 = 0,
    y0 = 0,
    sdy =.01,
    sdx =.01,
    alpha = myAlphaInit,
    theta = angle
  ),
  list(
    rx = rxinit,
    ry = ryinit,
    x0 = 0,
    y0 = 0,
    sdy =.01,
    sdx =.01,
    alpha = myAlphaInit,
    theta = angle
  ),
  list(
    rx = rxinit,
    ry = ryinit,
    x0 = 0,
    y0 = 0,
    sdy =.01,
    sdx =.01,
    alpha = myAlphaInit,
    theta = angle
  )
)

fit = stan(
  file = "ellipse_VM.stan",
  data = in.dat,
  init = inits,
  iter = niters,
  warmup = floor(niters/2),
  chains = n.chains,
  control = list(adapt_delta = .85,max_treedepth=11)
)
```

A.2.2 Stan code

```
data {
```

```
  int<lower=0> N;
  vector[N] x;
  vector[N] y;
}

parameters {
  real<lower=-pi()/2, upper = 0> alpha;
  real<lower=0> sdy;
  real<lower=0> sdx;

  real x0;
  real y0;

  real<lower = 0> rx;
  real<lower = 0> ry;

  vector<lower=-pi(), upper = pi()>[N] theta;
}

transformed parameters {
  vector[N] x_temp;
  vector[N] y_temp;

  vector[N] x_tran;
  vector[N] y_tran;

  for(i in 1:N){
    x_temp[i] = rx * cos(theta[i]);
    y_temp[i] = ry * sin(theta[i]);

    x_tran[i] = x0 + x_temp[i] * cos(alpha) - y_temp[i] * sin(alpha);
    y_tran[i] = y0 + y_temp[i] * cos(alpha) + x_temp[i] * sin(alpha);
  }
}

model {
  rx ~ normal(60, 10);
  ry ~ normal(60, 10);

  x0 ~ normal(0,1);
  y0 ~ normal(0,1);

  sdy ~ gamma(2,50);
  sdx ~ gamma(2,50);

  alpha ~ uniform(-pi()/2,0);

  theta ~ von_mises(0,0.1);

  x ~ normal(x_tran,sdx);
  y ~ normal(y_tran,sdy);
}
```

Acknowledgements

The authors would like to thank their colleagues: Adam Pintar, Blaza Toman, Kevin Coakley, Mike Frey, and Antonio Possolo for their very helpful comments.

REFERENCES

P. T. Boggs, R. H. Byrd, J. E. Rogers, and R. B. Schnabel. *User's Reference Guide for ODRPACK Version 2.01 Software for Weighted Orthogonal Distance Regression*. NIST Internal Report 4834, 1992.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 2017.

N. Chernov. *Circular and Linear Regression Fitting Circles and Lines by Least Squares*. CRC Press, Boca Raton, FL, 2011.

W. A. Fuller. *Measurement Error Models*. John Wiley and Sons, 1987.

A. Keksel, F. Ströer, and J. Seewig. Bayesian approach for circle fitting including prior knowledge. *Surface Topography: Metrology and Properties*, 6(3):035002, 2018.

T. Lafarge and A. Possolo. The NIST Uncertainty Machine. *NCLSI: Measure The Journal of Measurement Science*, 10(3):20–27, 2015.

L. Mamileti, C.-M. Wang, M. Young, and D. F. Vecchia. Optical fiber geometry by gray-scale analysis with robust regression. *Applied Optics*, 31(21):4182–4185, 1992.

K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley and Sons, Ltd., New York, NY, 2000.

Stan Development Team. RStan: the R interface to Stan, 2018. URL http://mc-stan.org/. R package version 2.18.2.

C.-M. Wang, D. F. Vecchia, M. Young, and N. A. Brilliant. Robust regression applied to optical-fiber dimensional quality control. *Technometrics*, 39(1):25–33, 1997.

M. Werman and D. Keren. A Bayesian method for fitting parametric and nonparametric models to noisy data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):528–534, 2001.