Term Project for Machine Learning: Censored Bayesian Polynomial Regression

Mikhail Mishin

April 2017

1 Project Description

Survival analysis is used to analyze data where the outcome variable is the time until occurrence of an event of interest; we call this time as survival time. The event of interest can be, for example, death, outbreak of a disease, divorce, or termination of a computer program.

Data of this kind are often incomplete. Typically, test subjects (e.g., patients) are followed for a fixed time observation period and the time of the occurrence of the event of interest is recorded. However, it may be that in some cases the event of interest did not occur during the period. In these cases, we do not know the exact time of the occurrence but only a lower bound. A datum is called censored if the information of its survival time is incomplete.

In this project we construct a simple Bayesian regression model for censored data. We assume that our data $\mathcal{D}$ consist of N observations. Each observation is a triple $(\boldsymbol{x}_i, y_i, c_i)$ where $\boldsymbol{x}_i \in \mathbb{R}^{D'}$ are covariates (predictors, independent variables, input), y_i is the observed survival time and c_i is the censoring time (the end of the observation period). Let z_i be the (unknown) true survival time. Our data are right-censored, that is,

$$y_i = \min(z_i, c_i)$$

We notice that if $y_i < c_i$ we know that $z_i = y_i$. However, if $y_i = c_i$ we know only that $z_i \geq c_i$.

We want to model the true survival times using regression. To this end, we specify a regression model. First, let $\phi(\boldsymbol{x}_i) : \mathbb{R}^{D'} \to \mathbb{R}^D$ be a basis function and let $\boldsymbol{w} \in \mathbb{R}$ be the parameters. Now, we can write

$$z_i = \boldsymbol{w}^T \phi(\boldsymbol{x}_i) + \eta_i,$$

where $\eta_i \sim \mathcal{N}(0, \beta^{-1})$ is the residual noise term and β is the noise precision. It follows that $z_i \sim \mathcal{N}(\boldsymbol{w}^T \phi(\boldsymbol{x}_i), \beta^{-1})$.

Finally, we specify a prior for the parameters $\boldsymbol{w}$. We use a Gaussian prior

$$\boldsymbol{w} \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I}),$$

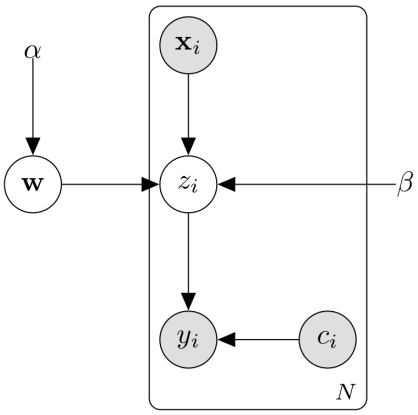

Figure 1: Plate diagram of the censored regression model.

where α is the prior precision.

Ignoring the terms that do not depend on the unknown quantities $\boldsymbol{z} = \{z_1, z_2, \ldots, z_N\}$ and $\boldsymbol{w}$, we have the following joint distribution

$$p(\boldsymbol{w}, \boldsymbol{z}, \mathcal{D}, \alpha, \beta) \propto p(\boldsymbol{w}|\alpha) \prod_{i=1}^{N} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i, \beta) p(y_i|z_i, c_i).$$

This distribution is illustrated in Figure 1. We use hyperparameter values $\alpha = \beta = 1$ and omit them from conditional distributions for simplicity.

In this project, we consider polynomial regression. We assume that $\boldsymbol{x}_i = x_i$ is a singleton, that is, $D' = 1$. To obtain d:th degree polynomial regression, we specify a basis function $\phi^{(d)}(x) = [1, x^1, x^2, \ldots, x^d]^T$. The constant term corresponds to the intercept. As a result, we have $D = d + 1$.

Due to the latent variables z_i , it is not possible to obtain the posterior distribution in analytic form here. Fortunately, it is possible to use the EM algorithm to obtain a maximum a posteriori (MAP) estimate for the parameters $\boldsymbol{w}$. In Bayesian EM, we want to find

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{w}|\mathcal{D}) = \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{w}, \mathcal{D})$$

The task for this project is to derive, implement, and test an EM algorithm for censored Bayesian polynomial regression.

2 Derivations

First, we derive the log-posterior functions for complete data and incomplete data. Then, we derive the EM update equations for the model.

2.1 Complete data log-posterior

We can treat data $\mathcal{D}$ as constant. Thus, the posterior is proportional to the joint distribution. The joint distribution function for the complete data is given by:

$$p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}) \propto p(\boldsymbol{w}, \boldsymbol{z}, \mathcal{D}) = p(\boldsymbol{w}) \prod_{i=1}^{N} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) p(y_i|z_i, c_i)$$

Taking the logarithm we obtain:

$$\log p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}) = \log p(\boldsymbol{w}) + \sum_{i=1}^{N} [\log p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) + \log p(y_i|z_i, c_i)] + \text{const} \quad (1)$$

2.2 Incomplete data log-posterior

The log-posterior function for the incomplete data is given by:

$$\begin{aligned}
\log p(\boldsymbol{w}|\mathcal{D}) &= \log \int_{-\infty}^{\infty} p(\boldsymbol{w}, \boldsymbol{z}|\mathcal{D}) d\boldsymbol{z} \\
&\propto \log \int_{-\infty}^{\infty} p(\boldsymbol{w}, \boldsymbol{z}, \mathcal{D}) d\boldsymbol{z} \\
&= \log \int_{-\infty}^{\infty} p(\boldsymbol{w}) \prod_{i=1}^{N} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) p(y_i|z_i, c_i) d\boldsymbol{z} \\
&= \log p(\boldsymbol{w}) + \log \int_{-\infty}^{\infty} \prod_{i=1}^{N} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) p(y_i|z_i, c_i) d\boldsymbol{z}
\end{aligned}$$

Moving the product out of the integral

$$\begin{aligned}
&= \log p(\boldsymbol{w}) + \log \prod_{i=1}^{N} \int_{-\infty}^{\infty} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) p(y_i|z_i, c_i) dz_i \\
&= \log p(\boldsymbol{w}) + \sum_{i=1}^{N} \log \int_{-\infty}^{\infty} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) p(y_i|z_i, c_i) dz_i \qquad (2)
\end{aligned}$$

where

$$\log p(\boldsymbol{w}) = -\frac{1}{2} \log |2\pi\alpha^{-1}\mathbf{I}| - \frac{1}{2} \boldsymbol{w}^T \alpha \mathbf{I} \boldsymbol{w} \qquad (3)$$

Observe that $p(y_i|z_i, c_i)$ is a delta distribution. Consider two cases:

- $y_i < c_i$:

$$p(y_i|z_i, c_i) = \begin{cases} 1, & \text{if } z_i = y_i, \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

- $y_i = c_i$:

$$p(y_i|z_i, c_i) = \begin{cases} 1, & \text{if } z_i >= c_i, \\ 0, & \text{otherwise} \end{cases} \qquad (5)$$

Therefore, the integral in (2) evaluates to:

$$\int_{-\infty}^{\infty} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i)p(y_i|z_i, c_i)dz_i = \begin{cases} p(Z_i = y_i|\boldsymbol{w}, \boldsymbol{x}_i), \text{ if } y_i < c_i, \\ \int_{c_i}^{\infty} p(z_i|\boldsymbol{w}, \boldsymbol{x}_i)dz_i, \text{ if } y_i = c_i \end{cases} \tag{6}$$

We can write the final formula for the incomplete data log-posterior using an indicator function:

$$\begin{aligned} \log p(\boldsymbol{w}|\mathcal{D}) = \log p(\boldsymbol{w}) + \sum_{i=1}^{N} [& (1 - \mathcal{I}(y_i = c_i)) \log f(y_i|\boldsymbol{w}^T \boldsymbol{x}_i, \beta^{-1}) \\ & + \mathcal{I}(y_i = c_i) \log(1 - F(c_i|\boldsymbol{w}^T \boldsymbol{x}_i, \beta^{-1}))] \end{aligned} \tag{7}$$

where $f(u|\mu, \sigma^2)$ and $F(u|\mu, \sigma^2)$ are the PDF and CDF of the Gaussian distribution with the mean μ and variance σ^2, respectively.

2.3 EM Update Equations

In Bayesian EM, we want to find $\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{w}, \mathcal{D})$.

- **E-step** To update z_i consider two cases:
  - If $y_i < c_i$,

$$\left\langle z_i \right\rangle_{Z|\mathcal{D}, \boldsymbol{w}_t} = \left\langle z_i | z_i = y_i, y_i \right\rangle = y_i \tag{8}$$

  - If $y_i = c_i$,

$$\left\langle z_i \right\rangle_{Z|\mathcal{D}, \boldsymbol{w}_t} = \left\langle z_i | z_i \geq c_i, \boldsymbol{w}_t, \boldsymbol{x}_i \right\rangle = \boldsymbol{w}_t^T \phi(\boldsymbol{x}_i) + \beta^{-\frac{1}{2}} \, \mathrm{H}\left(\frac{c_i - \boldsymbol{w}_t^T \phi(\boldsymbol{x}_i)}{\beta^{-\frac{1}{2}}} \right) \tag{9}$$

    , where

$$\mathrm{H}(u) = \frac{f(u)}{1 - F(u)}$$

    The last equation follows from the formula for the expectation of a normally distributed random variable bounded below ($z_i \geq c_i$).

- **M-step** Expected complete data joint distribution:

$$\begin{aligned} R(\boldsymbol{w}, \boldsymbol{w}_t) &= \left\langle \log p(\boldsymbol{w}, Z, \mathcal{D}) \right\rangle_{Z|\mathcal{D}, \boldsymbol{w}_t} \\ &= \log p(\boldsymbol{w}) + \sum_{i=1}^{N} \left\langle \log p(z_i|\boldsymbol{w}, \boldsymbol{x}_i) \right\rangle_{Z|\mathcal{D}, \boldsymbol{w}_t} + \text{const} \\ &= -\frac{1}{2}\boldsymbol{w}^T \alpha \mathbf{I} \boldsymbol{w} - \sum_{i=1}^{N} \left\langle \frac{(z_i - \boldsymbol{w}^T \phi(\boldsymbol{x}_i))^2}{2\beta^{-1}} \right\rangle_{Z|\mathcal{D}, \boldsymbol{w}_t} + \text{const} \\ &= -\frac{1}{2}\boldsymbol{w}^T \alpha \mathbf{I} \boldsymbol{w} - \sum_{i=1}^{N} \frac{\langle z_i^2 \rangle - 2\langle z_i \rangle \boldsymbol{w}^T \phi(\boldsymbol{x}_i) + (\boldsymbol{w}^T \phi(\boldsymbol{x}_i))^2}{2\beta^{-1}} + \text{const} \end{aligned} \tag{10}$$

Let us take the derivative of $R(\boldsymbol{w}, \boldsymbol{w}_t)$ with respect to $\boldsymbol{w}$:

$$\begin{aligned} \frac{\partial R(\boldsymbol{w}, \boldsymbol{w}_t)}{\partial \boldsymbol{w}} &= -\alpha \mathbf{I} \boldsymbol{w} + \frac{\beta}{2} \sum_{i=1}^{N} 2\langle z_i \rangle \phi(\boldsymbol{x}_i) - 2(\boldsymbol{w}^T \phi(\boldsymbol{x}_i))\phi(\boldsymbol{x}_i) \\ &= -\alpha \mathbf{I} \boldsymbol{w} + \beta \sum_{i=1}^{N} \langle z_i \rangle \phi(\boldsymbol{x}_i) - \beta \left(\sum_{i=1}^{N} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T \right) \boldsymbol{w} \\ &= -\left(\beta \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T + \alpha \mathbf{I} \right) \boldsymbol{w} + \beta \sum_{i=1}^{N} \langle z_i \rangle \phi(\boldsymbol{x}_i) \end{aligned} \tag{11}$$

Maximizing for $\boldsymbol{w}$:

$$\frac{\partial R(\boldsymbol{w}, \boldsymbol{w}_t)}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} = \left(\beta \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T + \alpha \mathbf{I} \right)^{-1} \beta \sum_{i=1}^{N} \langle z_i \rangle \phi(\boldsymbol{x}_i) \tag{12}$$

3 Evaluation

We estimate MAP parameters for polynomials with degrees $d = 1, \ldots, 5$ in Table 1. In our implementation of the EM algorithm we use 30 iterations and 10 random restarts. We plot the MAP curves using the estimated parameters in Figure 2. For comparison, we also plot the MAP curves for the standard regression for the model $y_i = \boldsymbol{w}^T \phi(\boldsymbol{x}_i) + \eta_i$. We use the analytic formula to find the parameters (omitting the derivation):

$$\boldsymbol{w} = \left(\beta \sum_{i=1}^{N} \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^T + \alpha \mathbf{I} \right)^{-1} \beta \sum_{i=1}^{N} y_i \phi(\boldsymbol{x}_i) \tag{13}$$

Comparing two curves, one can see how the censored regression takes into account the censored data that is ignored by the standard regression. To verify the convergence of our algorithm we also plot incomplete data log-posterior as a function of iteration (Figure 3). Indeed, log-posterior increases after each M-step and converges quickly.

Degree	**Estimated MAP parameters**
1	[5.2797; 1.2552]
2	[5.7734; 1.0746; -0.4908]
3	[5.8030; 1.2196; -0.5033; -0.0582]
4	[5.7066; 1.2963; -0.1504; -0.0794; -0.1048]
5	[5.7278; 1.4491; -0.1922; -0.2818; -0.0958; 0.0468]

Table 1: Estimated MAP parameters for censored regression

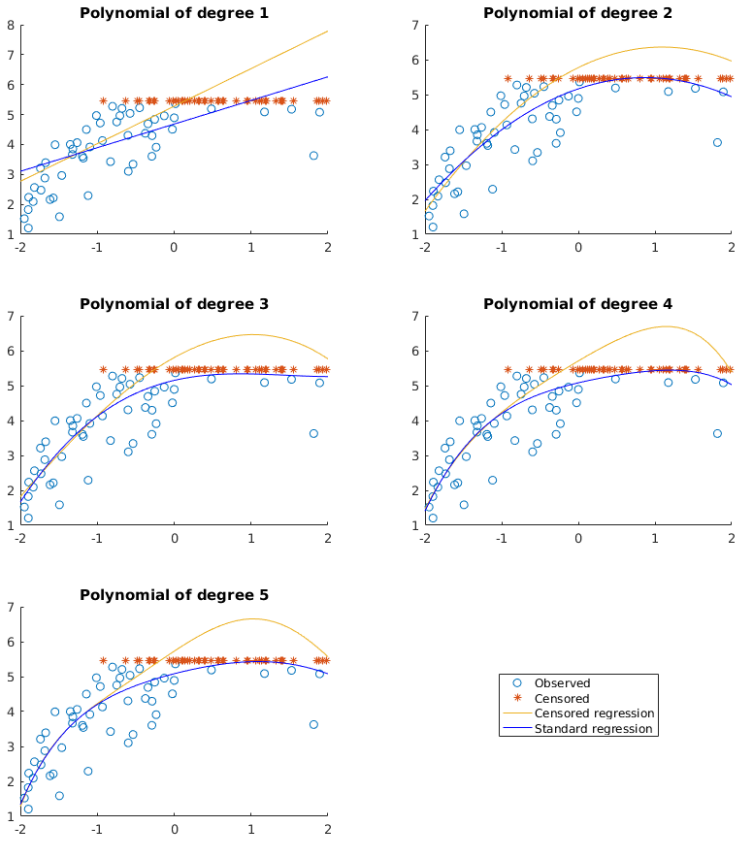

Figure 2: MAP curves for censored and standard regression

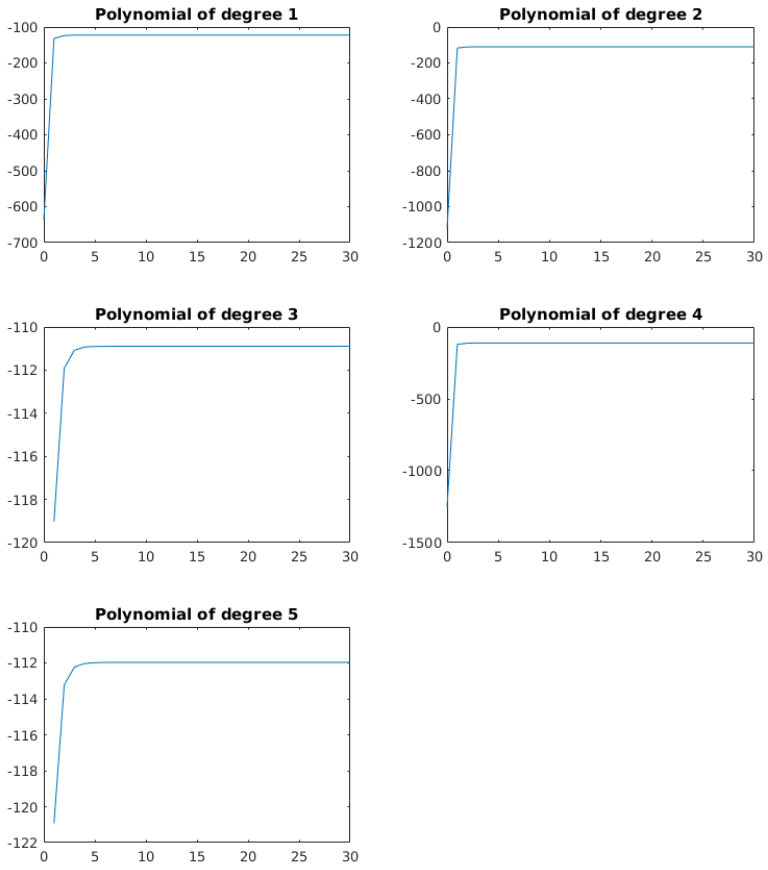

Figure 3: Incomplete data log-posterior as a function of iteration

4 Model Selection

To select a model we compute Bayesian information criterion (BIC) for each d (Figure 4). We choose the model with the lowest BIC - polynomial of degree $d = 2$. Looking at the MAP curves from Figure 2 again, we can say that our choice is consitent and reasonable.

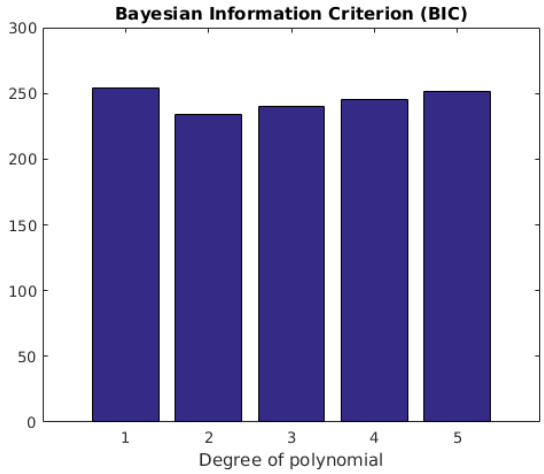

Figure 4: Bayesian information criterion (BIC)

5 Implementation

Code for computing the log-posterior function for the incomplete data:

```
function [ logpst ] = logPosterior( phi_x, y, c, w, alpha, beta )
%LOGPOSTERIOR Compute incomplete data log-posterior
    d = length(w);
    log_pw = -d/2*log(2*pi*1/alpha) - 1/2*(w'*alpha*eye(d)*w);
    sum_log_pz = 0;
    mean_z = phi_x*w;
    for i = 1:length(y)
        if c(i)
            % y_i = c_i
            sum_log_pz = sum_log_pz + ...
                log(1 - normcdf(y(i), mean_z(i), beta^(-1/2)));
        else
            % y_i < c_i
            sum_log_pz = sum_log_pz + ...
                log(normpdf(y(i), mean_z(i), beta^(-1/2)));
        end
    end
    logpst = log_pw + sum_log_pz;
end
```

Implementation of the EM algorithm:

```
for d = 1:max_degree

    best_log_post = -Inf;
    best_w = -Inf*ones(d+1, 1); % MAP estimate for parameters
    phi_x = polyBasis(x, d);
    S = inv(beta*(phi_x'*phi_x) + alpha*eye(d+1));
    E_z = zeros(size(y));

```

```
    for s = 1:n_starts
        % Initialize weights
        w = mvnrnd(zeros(1,d+1), alpha^(-1/2)*eye(d+1))';
        % incomplete data log-posterior
        log_posts = zeros(n_iter + 1, 1);
        % log-posterior before first iteration
        log_posts(1) = logPosterior(phi_x, y, c, w, alpha, beta);

        for i = 1:n_iter
            % E-step
            mean_z = phi_x*w;
            for j = 1:n_obs
                if c(j)
                    alpha2 = (y(j) - mean_z(j))/beta^(-1/2);
                    E_z(j) = mean_z(j) + beta^(-1/2)*H_function(
alpha2);
                else
                    E_z(j) = y(j);
                end
            end
            % M-step
            w = beta*S*phi_x'*E_z;

            % Compute log-posterior
            log_posts(i+1) = logPosterior(phi_x, y, c, w, alpha,
beta);
        end
        if log_posts(n_iter+1) > best_log_post
            best_log_post =  log_posts(n_iter+1);
            best_w = w;
            log_postconvergences = log_posts;
        end

    end

    % Compute BIC
    bics(d) = bic(best_log_post, d, n_obs);

    % Store MAP estimate
    ws{d} = best_w;
end
```

Code for computing polynomial basis functions:

```
function phi_x = polyBasis(x, d)
%POLYBASIS Polynomial basis function
    phi_x = bsxfun(@power, x, 0:d);
end
```

Code for computing Bayesian information criterion:

```
function [ crt ] = bic( log_post, d, n_obs )
%BIC Bayesian information criterion
    crt = (d+1)*log(n_obs) - 2*log_post;
end
```

6 Acknowledgements

The section Project Description is taken from the original description for the course written by Pekka Marttinen, Pekka Parviainen, Pedram Daee, and Sami Remes.