Prediction and Change Detection

Abhishek Singh

https://github.com/Abhishek

Department of Mechanical Engineering,

Indian Institute of Technology, Jodhpur

Decisions often require quick response to changes. For example, equity analysts need to quickly identify market changes in order to adjust their investment strategies. Coaches need to track changes in player performance in order to adjust their strategies. When tracking changes, you incur costs if you observe more or less changes than they actually happened. For example, if too conservative criteria are used to detect changes, equity analysts may miss important short-term trends and instead interpret them as random fluctuations. On the other hand, changes are easily recognizable. For example, in basketball, players who form a series of consecutive hoops are often identified as "hot hand" players who feel that their underlying skills have suddenly increased. This can lead to a suboptimal path strategy due to random fluctuations.

In a sequential prediction task, we're looking for ways to explain individual differences .The aim for observers is to predict the next datum in the sequence using stimuli supplied by a basic statistical method. At random points in time, the statistical process latent parameters change discretely. Accurate identification of such changepoints, as well as inference about future outcomes based on the outcomes that preceded the most recent inferred changepoint, are required for success in this endeavour. We adopt a Bayesian approach to the change-point identification problem and develop a simple inference procedure to predict the next datum in a sequence.

Prediction

The sequential prediction task is discussed first, followed by a Bayesian analysis of the prediction problem. The outcomes of a few individuals in this prediction task are then discussed, and then we illustrate how the Bayesian technique may capture individual differences with a single "twitchiness" parameter that reflects how quickly changes are noticed in random sequences.

Two inference approaches are used to break down the prediction problem. To begin, the locations of the changepoints must be determined. Following that, depending on the most recent changepoint locations, predictive inference for the upcoming outcome is made. For change-point situations involving a single or several changepoints, several Bayesian techniques have been developed. While integrating out, we use a Markov Chain Monte Carlo (MCMC) approach to approximate the joint posterior distribution over changepoint assignments χ while integrating out θ. The posterior marginal distribution will be sampled using Gibbs sampling. Following that, the samples can be used to forecast the next outcome in the series. To apply Gibbs sampling, we evaluate the conditional probability of assigning a changepoint at time i, given all other changepoint assignments and the current α value:

$$P\left(x_i \mid x_{-i}, y, \alpha\right) = \int_{\theta} P\left(x_i, \theta, \alpha \mid x_{-i}, y\right)$$

Where x_{-i} represents all switch point assignments except x_i . This can be simplified by considering the location of the most recent changepoint preceding and following time i and the outcomes occurring between these locations. Let n_i^L be the number of time steps from the last changepoint up to and including the current time step i such that $x_{i-n_i^L}=1$ and $x_{i-n_i^L+j}=1$

for $0<j<n_i^L$. Similarly, let n_i^R be the number of time steps that follow time step i up to the next changepoint such that $x_{i+n_i^L-j}=0$ and $x_{i+n_i^R}=1$ for $0<j<n_i^R$.

Let

$$y_i^L = \sum\nolimits_{i-n_i^L<k\le i} y_k \qquad y_i^R = \sum\nolimits_{k<k\le i+n_i^R} y_k$$

The update equation for the changepoint assignment can then be simplified to:

$$P(x_i = m \mid x_{-i}) \propto \begin{cases} (1-\alpha)\displaystyle\prod_{j=1}^{D} \frac{\Gamma\left(1+y_{i,j}^L+y_{i,j}^R\right)\Gamma\left(1+Kn_i^L+Kn_i^R-y_{i,j}^L-y_{i,j}^R\right)}{\Gamma\left(2+Kn_i^L+Kn_i^R\right)} & m=0 \\ \alpha\displaystyle\prod_{j=1}^{D} \frac{\Gamma\left(1+y_{i,j}^L\right)\Gamma\left(1+Kn_i^L-y_{i,j}^L\right)\Gamma\left(1+y_{i,j}^R\right)\Gamma\left(1+Kn_i^R-y_{i,j}^R\right)}{\Gamma\left(2+Kn_i^L\right)\Gamma\left(2+Kn_i^R\right)} & m=1 \end{cases}$$

We initialize the Gibbs sampler by sampling each xt from a Bernoulli(α) distribution. All changepoint assignments are then updated sequentially by the Gibbs sampling equation above. The sampler is run for M iterations after which one set of changepoint assignments is saved. The Gibbs sampler is then restarted multiple times until S samples have been collected.

Predictive Inference

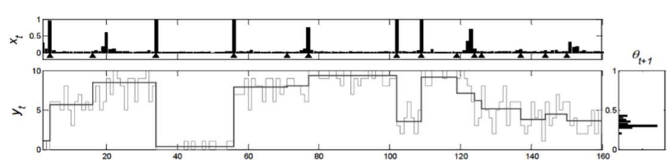
The next latent parameter value θ_{t+1} and outcome y_{t+1} can be predicted on the basis of observed outcomes that occurred after the last inferred changepoint:

$$\theta_{t+1,j} = \sum_{i=t^*+1}^{t} y_{i,j}/K, \qquad y_{t+1,j} = \text{round}\left(\theta_{t+1,j}K\right)$$

where t* is the location of the most recent change point. By considering multiple Gibbs samples, we get a distribution over outcomes y_{t+1}. We base the model predictions on the mean of this distribution.

Illustration of model performance

The performance of the model on a one dimensional sequence (D=1) generated from the changepoint model with T=160, α=0.05, and K=10. The Gibbs sampler was run for M=30 iterations and S=200 samples were collected. The top panel shows the actual changepoints (triangles) and the distribution of changepoint assignments averaged over samples. The bottom panel shows the observed data y (thin lines) as well as the θ values in the generative model (rescaled between 0 and 10). At locations with large changes between observations, the marginal changepoint probability is quite high. At other locations, the true change in the mean is very small, and the model is less likely to put in a changepoint. The lower right panel shows the distribution over predicted θt+1 values.



CONCLUSION

We used an online prediction task to study changepoint detection. Human observers had to predict the next observation in stochastic sequences containing random changepoints. We showed that some observers are too “twitchy”: They perform poorly on the prediction task because they see changes where only random fluctuation exists. Other observers are not twitchy enough, and they perform poorly because they fail to see small changes.