Frequentist and Bayesian Learning Approaches to Artificial Intelligence

Sunghae Jun
Department of Statistics, Cheongju University, Cheongju, Korea

Abstract
Artificial intelligence (AI) is making computer systems intelligent to do right thing. The AI is used today in a variety of fields, such as journalism, medical, industry as well as entertainment. The impact of AI is becoming larger day after day. In general, the AI system has to lead the optimal decision under uncertainty. But it is difficult for the AI system can derive the best conclusion. In addition, we have a trouble to represent the intelligent capacity of AI in numeric values. Statistics has the ability to quantify the uncertainty by two approaches of frequentist and Bayesian. So in this paper, we propose a methodology of the connection between statistics and AI efficiently. We compute a fixed value for estimating the population parameter using the frequentist learning. Also we find a probability distribution to estimate the parameter of conceptual population using Bayesian learning. To show how our proposed research could be applied to practical domain, we collect the patent big data related to Apple company, and we make the AI more intelligent to understand Apple’s technology.

Keywords: Statistics, Artificial intelligence, Bayesian inference, Frequentist, Learning from data, Apple technology

1. Introduction
By the remarkable developments of learning algorithms, the technologies for artificial intelligence (AI) are constantly evolving. Recently the AI of Google DeepMind, AlphaGo defeated the human champion at the game of Go [1]. The AlphaGo learned from diverse data with deep neural networks and tree search [2]. Also the IBM Watson won in the popular quiz show, Jeopardy [3, 4]. They were all developed by diverse learning algorithms. Statistics and machine learning are typical approaches to AI learning [5]. In this paper, we focus on the statistics for making AI more intelligent. Ross [6] defined statistics as follow; “Statistics is the art of learning from data.” This means that statistics has learning ability to get intelligence for optimal decision. Human is more and more intelligent through learning. Machine also gets its intelligence by learning from data. Machine learning (ML) is to study the way how machine (computer) learns from experience (data) [7]. The ML is an important area of AI [5]. Using ML investigations, we can improve the AI performance based on data. AI wants to automatically learn from complex examples for making intelligent decisions optimally. Both statistics and ML are data learning tools for intelligent decisions. But the difference between the two is to apply the probabilistic assumption for error term. Statistics represents the structure of input (X) and output (Y) variables as follow [8].

\[ Y = f(X) + \epsilon, \]  
(1)
where the error variable $\epsilon$ has zero means, $E(\epsilon) = 0$, also it is uncorrelated to the input variable $X$ as follow [9].

\[ \text{Cov}(\epsilon, X) = 0. \]  \hspace{1cm} (2)

That is, the covariance value between $\epsilon$ and $X$ is 0. This assumption is made the basis of statistics. In comparison, there is not any assumption of error in the ML. The error in statistics controls the uncertainty of complex environments by probabilistic approach. In this paper, we propose a connection between statistics and AI to improve the performance of AI technology. The remainder of this paper is made up as follows. Section 2 introduces the research background related to our research. We propose the connection between statistics and AI using the causal inference and learning approaches to construct the efficient and effective AI in Section 3. A case study for verifying the performance of our research is shown in Section 4. In Section 5, we present our conclusions.

2. Research Background

So many studies on making AI were published in diverse research fields. In addition, most of them were based on the learning methodologies such as fuzzy logic or statistics [10-13]. In this paper, we study on the learning methods based on statistics for making AI system. Statistics has two approaches, frequentist and Bayesian to data analysis [14]. The frequentist is interesting to compute the probability of result ($x$) given cause ($\theta$). In comparison, the Bayesian is focused on the probability of cause ($\theta$) when the result ($x$) is occurred. Figure 1 shows two approaches to learning process.

In Figure 1, the frequentist builds the probability model $P(X|\theta)$ from cause to result. The model based on the Bayesian is represented by $P(\theta|X)$ from result to cause. Most researches for statistical learning process were based on Bayesian statistics [15, 16]. Bayesian statistics is based on Bayesian inference. This is the fitting process of probability model to data and result by probability distribution for predicting new observation, and has learning process as follow [15].

\[ \text{Prior} \times \text{Likelihood} \propto \text{Posterior}. \]  \hspace{1cm} (3)

3. Connection between Statistics and Artificial Intelligence

AI is the intelligence based on machine or computer systems, that is, the AI is to make computers do the right things [5]. In addition it can be existed in agent or software [5]. In diverse fields, the AI researches have been performed [2, 17]. Most of them were depended on searching methods for problemsolving. To solve the given problems, the AI searches the similar cases from knowledge databases. This provides good performances when the AI has complete knowledge databases for given problem. But the AI is having trouble not preparing the perfect examples. To overcome this problem of AI, we propose a statistical intelligence based on learning from data. Figure 2 shows the statistical intelligence for AI.

We meet small and big data sets in the work of learning from data for constructing AI systems. From the small and big data, we get the intelligence for the optimal decision in AI using statistics as a learning tool. In addition, the Bayesian statistics has its strong learning power. Figure 3 represents the Bayesian learning process.

We set prior and posterior of Bayesian statistics to current and new intelligences respectively. Also the likelihood of Bayesian
Where $X$ is explanatory or independent variable, $Y$ is response or dependent variable. Also $\theta$ is the parameter vector of probability model. General method of this probability model is regression. This model can be used for the AI systems do the right thing. Using the observed data including $X$ and $Y$, we make a probability model for AI. When given new $X$ in an environment, the AI system can do optimal decisions for $Y$. In the frequentist statistics, $\theta$ is updated by new experience of $X$ and $Y$ as follow:

$$P(Y \mid \theta_{\text{new}}, X) \rightarrow P(Y \mid \theta_{\text{current}}, X) + P(Y \mid \theta_{\text{learning}}, X).$$  \hspace{1cm} (6)

Here $P(\cdot)$ represents a probability model. In addition, $P(Y \mid \theta_{\text{current}}, X)$ is the previous belief about $\theta$, and $P(Y \mid \theta_{\text{learning}}, X)$ represents the belief learned by new given data. These are combined to compute the updated belief $P(Y \mid \theta_{\text{new}}, X)$. So the AI systems can do the right thing every single time using the $P(Y \mid \theta_{\text{new}}, X)$ in the frequentist statistics. Next in the Bayesian statistics for the AI systems, we consider Bayes’ rule firstly as follow:

$$P(\text{Cause} \mid \text{Result}) = \frac{P(\text{Result} \mid \text{Cause})P(\text{Cause})}{P(\text{Result})}. \hspace{1cm} (7)$$

In Bayesian statistics, the probability of cause given result is computed by the probability of cause and the conditional probability of result given cause. In comparison, the frequentist is interesting the conditional probability of result given cause as follow, $P(\text{Result} \mid \text{Cause})$. So we build Bayesian regression model using Bayes' rule as follow:

$$P(\theta \mid X, Y) \propto P(\theta) P(Y \mid X, \theta). \hspace{1cm} (8)$$

In Bayesian statistics for AI systems, we get the updated value of $\theta$ for constructing Bayesian regression model for optimal decision of $Y$ given $X$. That is, given the data of $X$ and $Y$, we get the probability distribution of the parameter vector ($\theta$). This means that the parameter is considered as random variable in Bayesian statistics. Using the result of the behavior of $\theta$, the AI systems can do the right thing. But the frequentist assumes that the parameter is fixed (not random variable) and the data are random [15]. In the process of AI decision, the prior $P(\theta)$ represents the current intelligent capacity of AI system, and we can quantify the intelligence of AI to a probability distribution. In the Bayesian regression model for AI systems, the regression parameters show the intelligence power of AI. Also the AI system learns from data which are represented by the likelihood function $P(Y \mid X, \theta)$. By combining the current intelligence of AI ($P(\theta)$) with the learning result of AI ($P(Y \mid X, \theta)$), the AI system is updated to the improved intelligence represented by $P(\theta \mid X, Y)$. When the AI system has a problem to be solved, it use the simulation result from the probability distribution of

![Figure 4. Frequentist learning process.](Image)
4. Experimental Results

In our experiments, we carried out two approaches to connect statistics to AI. First we performed the frequentist statistics for learning from data. To illustrate how our proposed approach could be applied in real problem, we used patent documents applied by Apple. Apple is a leading company in technological innovation, and many researches about analysis Apple’s technology were proposed [18, 19]. In this paper, we extracted keywords from the retrieved patent documents of Apple using text mining techniques as follow: application, circuit, code, component, computer, connector, device, display, graphics, image, information, interface, location, media, memory, mobile, portable, power, receiving, remote, request, screen, set, storage, system, user, video, voltage, window, wireless [18-21]. In the selected keywords, we determined ‘system’ as a response variable, and the rest as explanatory variables. Our learning approach is based on the linear model as follow:

\[ \text{system} = \theta_0 + \theta_1 \text{application} + \theta_2 \text{circuit} + \cdots + \theta_{29} \text{wireless}. \]  

(9)

In this paper, we make the AI to predict the optimal value of ‘system’ given new input values of ‘application,’ ‘circuit,’ ‘code,’ ‘component,’ ‘computer,’ ‘connector,’ ‘device,’ ‘display,’ ‘graphics,’ ‘image,’ ‘information,’ ‘interface,’ ‘location,’ ‘media,’ ‘memory,’ ‘mobile,’ ‘portable,’ ‘power,’ ‘receiving,’ ‘remote,’ ‘request,’ ‘screen,’ ‘set,’ ‘storage,’ ‘user,’ ‘video,’ ‘voltage,’ ‘window,’ and ‘wireless.’ In this experiment, we used total 8,119 patents for the AI learning. We sampled 2,000 patents without replacement from the all patents by four times for the AI learning as shown in Figure 5.

Figure 5. Dividing total data into four learning data sets.

We considered all collected patents as the conceptual population, and sampled four patent groups without replacement. Using the groups we performed step by step learning for making the AI. Firstly we carried out the AI learning based on frequentist statistics. The parameter is updated by the following method.

\[ \theta_{\text{new}} = \frac{\theta_{\text{current}} + \theta_{\text{learning}}}{2}. \]  

(10)

The learning result by frequentist regression based on four patent groups is shown Table 1.

Table 1 represents the estimated values of parameters of all input keywords by learning four steps. The first column is the population parameter value, and the other columns are the estimated values for population parameter by the learning steps. We knew the estimated values are approximated to the parameter of conceptual population as the learning progressed. To find the difference between the estimated value and parameter in each keyword, we computed the parameter difference between frequentist learning step and conceptual population in Table 2.

In Table 2, the last row shows the absolute mean values of each learning step, and we knew the values decreased according to repeated learning. That is, the absolute mean of the difference of 4th learning is the smallest among them. So we found the frequentist approach to make AI has its learning validity. Next we performed Bayesian based learning for building AI in same model to frequentist approach. Table 3 shows the Bayesian learning regression for AI decision.

In this experiment, we used the keywords same to the frequentist learning. The first column of Table 3 shows the parameter values of keywords in conceptual population like the frequentist. The second and third columns are the estimated values by Bayes’ rule and Markov chain Monte Carlo (MCMC) method respectively. In addition, the fourth and fifth columns represent the lower and upper of confidence interval for the parameter of conceptual population. To compare the performances of Bayes and MCMC, we computed the parameter difference between Bayesian learning step and conceptual population in Table 4.

The last row shows the absolute mean values of Bayes and MCMC learning methods, and we knew the value of MCMC is smaller than Bayes. So we concluded that the Bayesian learning based on MCMC is an efficient approach to make AI decision. Figure 6 shows the posterior functions of four input keywords, ‘device,’ ‘interface,’ ‘mobile,’ and ‘user’ using the MCMC simulation.

We perform the MCMC simulation using these probability distributions for optimal decision by AI. In general, the frequentist learning provides a value for making computer intelligent, but the Bayesian learning gives a probability distribution (posterior function) to have computer do the right things.
Table 1. Frequentist learning regression for artificial intelligence (AI) decision

| Keyword      | Conceptual population | 1st learning | 2nd learning | 3rd learning | 4th learning |
|--------------|-----------------------|--------------|--------------|--------------|--------------|
| application  | 0.0541                | 0.1240       | 0.0939       | 0.0711       | 0.0745       |
| circuit      | -0.0608               | -0.0139      | -0.0708      | -0.0833      | -0.0744      |
| code         | 0.0763                | 0.0445       | 0.0515       | 0.0520       | 0.0366       |
| component    | 0.0598                | 0.0785       | 0.0830       | 0.0865       | 0.0727       |
| computer     | 0.2912                | 0.2896       | 0.3157       | 0.2935       | 0.2873       |
| connector    | 0.1018                | 0.2087       | 0.1596       | 0.1076       | 0.0934       |
| device       | -0.0574               | -0.0491      | -0.0462      | -0.0473      | -0.0534      |
| display      | 0.0464                | 0.0577       | 0.0615       | 0.0595       | 0.0449       |
| graphics     | 0.1479                | 0.0371       | 0.0283       | 0.0525       | 0.0478       |
| image        | 0.0363                | 0.0802       | 0.0157       | 0.0239       | 0.0202       |
| information  | 0.1600                | 0.1929       | 0.1183       | 0.0804       | 0.0754       |
| interface    | 0.0507                | 0.1613       | 0.1154       | 0.1271       | 0.0929       |
| location     | 0.1103                | -0.0478      | 0.0476       | 0.0907       | 0.1290       |
| media        | 0.0265                | 0.0262       | 0.0029       | 0.0025       | 0.0040       |
| memory       | 0.1033                | 0.0607       | 0.1325       | 0.1837       | 0.1683       |
| mobile       | -0.1012               | -0.1876      | -0.1522      | -0.1153      | -0.1167      |
| portable     | -0.0634               | -0.0768      | -0.0924      | -0.0779      | -0.0781      |
| power        | 0.1264                | 0.2064       | 0.1759       | 0.1609       | 0.1403       |
| receiving    | 0.2806                | 0.1515       | 0.2032       | 0.2446       | 0.2609       |
| remote       | 0.2695                | 0.2171       | 0.1877       | 0.2097       | 0.2611       |
| request      | 0.4248                | 0.5991       | 0.5308       | 0.5233       | 0.5191       |
| screen       | -0.1304               | -0.1687      | -0.1636      | -0.1580      | -0.1605      |
| set          | 0.0611                | 0.0394       | 0.0379       | 0.0401       | 0.0487       |
| storage      | 0.1423                | 0.2278       | 0.1920       | 0.2069       | 0.1993       |
| user         | -0.0478               | -0.0711      | -0.0722      | -0.0786      | -0.0737      |
| video        | 0.0563                | 0.0848       | 0.0787       | 0.0661       | 0.0700       |
| voltage      | -0.0582               | -0.0493      | -0.1157      | -0.1377      | -0.1201      |
| window       | 0.0812                | 0.0577       | 0.0672       | 0.0547       | 0.0459       |
| wireless     | 0.0877                | 0.3532       | 0.2320       | 0.1818       | 0.1579       |

5. Conclusions

In this paper, we proposed frequentist and Bayesian approaches to make computer intelligent. Both frequentist and Bayesian are statistical approaches to learning from data. But there is a broad distinction between the frequentist and Bayesian. The frequentist learning is only depended on the given data, while the Bayesian learning is performed by the prior belief as well as the given data. Both ways of the frequentist and Bayesian are all important to AI decision. So we performed two approaches to make computer more intelligent. To verify the performance of our methodology, we made experiments using all patents applied by Apple. To make the AI do the right things about Apple technologies, we considered the frequentist and Bayesian learning methods. In the frequentist learning, we combined the current values for estimating parameter with the learning value based on new given data to get updated new parameter. Also we found that the MCMC simulation has better performance than the Bayes’ rule. Also we can get the posterior distribution to support AI do optimal decision using the Bayesian learning based on MCMC. In addition, we knew that the frequentist learning provides a fixed value to predict the parameter for...
Table 2. Parameter difference between frequentist learning step and conceptual population

| Keyword   | Diff. 1 | Diff. 2 | Diff. 3 | Diff. 4 |
|-----------|---------|---------|---------|---------|
| application | 0.0699  | 0.0397  | 0.0170  | 0.0204  |
| circuit    | 0.0468  | 0.0100  | 0.0225  | 0.0136  |
| code       | 0.0318  | 0.0248  | 0.0243  | 0.0397  |
| component  | 0.0187  | 0.0232  | 0.0267  | 0.0129  |
| computer   | 0.0016  | 0.0245  | 0.0022  | 0.0040  |
| connector  | 0.1069  | 0.0578  | 0.0058  | 0.0084  |
| device     | 0.0083  | 0.0112  | 0.0102  | 0.0041  |
| display    | 0.0113  | 0.0150  | 0.0131  | 0.0016  |
| graphics   | 0.1108  | 0.1196  | 0.0954  | 0.1001  |
| image      | 0.0440  | 0.0205  | 0.0124  | 0.0161  |
| information | 0.0328 | 0.0417  | 0.0796  | 0.0846  |
| interface  | 0.1106  | 0.0647  | 0.0765  | 0.0422  |
| location   | 0.1581  | 0.0627  | 0.0196  | 0.0187  |
| media      | 0.0003  | 0.0236  | 0.0240  | 0.0225  |
| memory     | 0.0425  | 0.0292  | 0.0804  | 0.0650  |
| mobile     | 0.0863  | 0.0509  | 0.0141  | 0.0154  |
| portable   | 0.0134  | 0.0290  | 0.0145  | 0.0147  |
| power      | 0.0800  | 0.0495  | 0.0345  | 0.0139  |
| receiving  | 0.1291  | 0.0774  | 0.0359  | 0.0197  |
| remote     | 0.0523  | 0.0817  | 0.0598  | 0.0083  |
| request    | 0.1743  | 0.1060  | 0.0986  | 0.0943  |
| screen     | 0.0383  | 0.0332  | 0.0275  | 0.0300  |
| set        | 0.0217  | 0.0232  | 0.0210  | 0.0124  |
| storage    | 0.0855  | 0.0497  | 0.0646  | 0.0570  |
| user       | 0.0234  | 0.0245  | 0.0308  | 0.0259  |
| video      | 0.0285  | 0.0224  | 0.0099  | 0.0137  |
| voltage    | 0.0090  | 0.0575  | 0.0795  | 0.0618  |
| window     | 0.0235  | 0.0140  | 0.0265  | 0.0353  |
| wireless   | 0.2655  | 0.1443  | 0.0941  | 0.0702  |
| Absolute mean | 0.0629 | 0.0459  | 0.0386  | 0.0319  |

making AI more intelligent. On the other hand, the Bayesian learning can estimate the parameter by not a value but a probability distribution called by posterior function. Therefore we can select the frequentist and Bayesian learning approaches to AI by the target jobs.

In our research, we applied the proposed methodology to linear model, but we can expand our study to other analytical domains such as classification and clustering. This is our future research works. In addition, more advanced statistical methods are needed to make AI more intelligent, this also our homework on the future work.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

[1] Google DeepMind, “AlphaGo: the first computer program to ever beat a professional player at the game of go,” Available https://deepmind.com/alpha-go.html
[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, et al., “Mastering the game of Go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484-489, 2016. http://dx.doi.org/10.1038/nature16961
[3] IBM Watson, Available http://www.ibm.com/smarterplanet/us/en/ibmwatson
[4] J. Jackson, “IBM Watson vanquishes human jeopardy foes,” Available http://www.pcworld.com/article/219893
[5] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach, 3rd ed. Essex: Pearson, 2014.
[6] S. M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 4th ed. (Transl. K. S. Lee). New York: Academic Press, 2012.
[7] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Waltham, MA: Morgan Kaufmann, 2012.
Table 3. Bayesian learning regression for AI decision

| Keyword    | Conceptual population | Bayes   | MCMC   | CI (2.5%) | CI (97.5%) |
|------------|-----------------------|---------|--------|-----------|------------|
| application| -0.0608               | 0.0541  | -0.0608| 0.0208    | 0.0871     |
| circuit    | 0.0763                | 0.0598  | -0.0605| -0.0977   | -0.0229    |
| code       | 0.2912                | 0.1018  | 0.2914 | 0.2630    | 0.3203     |
| component  | 0.0598                | 0.2912  | 0.0599 | 0.0079    | 0.1119     |
| computer   | 0.1018                | -0.0574 | 0.1017 | 0.0594    | 0.1441     |
| device     | -0.0574               | 0.0464  | -0.0576| -0.0759   | -0.0392    |
| display    | 0.0464                | 0.1479  | 0.0464 | 0.0222    | 0.0707     |
| graphics   | 0.1479                | 0.0363  | 0.1482 | 0.0963    | 0.2019     |
| image      | 0.0363                | 0.1600  | 0.0362 | 0.0133    | 0.0590     |
| information| 0.1600                | 0.0507  | 0.1603 | 0.1297    | 0.1907     |
| interface  | 0.0507                | 0.1103  | 0.0502 | 0.0135    | 0.0881     |
| location   | 0.1103                | 0.0265  | 0.1102 | 0.0599    | 0.1620     |
| media      | 0.0265                | 0.1033  | 0.0265 | 0.0107    | 0.0420     |
| memory     | 0.1033                | -0.1012 | 0.1033 | 0.0787    | 0.1277     |
| mobile     | -0.1012               | -0.0634 | -0.1012| -0.1501   | -0.0519    |
| portable   | -0.0634               | 0.1264  | -0.0634| -0.1011   | -0.0260    |
| power      | 0.1264                | 0.2806  | 0.1263 | 0.1027    | 0.1505     |
| receiving  | 0.2806                | 0.2695  | 0.2807 | 0.2085    | 0.3540     |
| remote     | 0.2695                | 0.4248  | 0.2692 | 0.2100    | 0.3285     |
| request    | 0.4248                | -0.1304 | 0.4243 | 0.3609    | 0.4886     |
| screen     | -0.1304               | 0.0611  | -0.1301| -0.1919   | -0.0684    |
| set        | 0.0611                | 0.1423  | 0.0610 | 0.0209    | 0.1008     |
| storage    | 0.1423                | -0.0478 | 0.1426 | 0.0935    | 0.1915     |
| user       | -0.0478               | 0.0563  | -0.0477| -0.0717   | -0.0234    |
| video      | 0.0563                | -0.0582 | 0.0562 | 0.0290    | 0.0833     |
| voltage    | -0.0582               | 0.0812  | -0.0581| -0.1039   | -0.0117    |
| window     | 0.0812                | 0.0877  | 0.0813 | 0.0372    | 0.1253     |
| wireless   | 0.0877                | 0.0877  | 0.0881 | 0.0415    | 0.1346     |

AI, artificial intelligence; MCMC, Markov chain Monte Carlo; CI, confidence interval.

[8] M. Berthold and D. J. Hand, *Intelligent Data Analysis: An Introduction*, Berlin: Springer, 1999. [http://dx.doi.org/10.1007/978-3-662-03969-4](http://dx.doi.org/10.1007/978-3-662-03969-4)

[9] M. Akritas, *Probability and Statistics with R for Engineers and Scientists*, Boston, MA: Pearson, 2015.

[10] A. B. Khalifa and H. Frigui, “Multiple instance Mamdani fuzzy inference,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 15, no. 4, pp. 217-231, 2015. [http://dx.doi.org/10.5391/IJFIS.2015.15.4.217](http://dx.doi.org/10.5391/IJFIS.2015.15.4.217)

[11] J. S. Kim and J. S. Jeong, “Pattern recognition of ship navigational data using support vector machine,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 15, no. 4, pp. 268-276, 2015. [http://dx.doi.org/10.5391/IJFIS.2015.15.4.268](http://dx.doi.org/10.5391/IJFIS.2015.15.4.268)

[12] M. Kim, “Online selective-sample learning of hidden Markov models for sequence classification,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 15, no. 3, pp. 145-152, 2015. [http://dx.doi.org/10.5391/IJFIS.2015.15.3.145](http://dx.doi.org/10.5391/IJFIS.2015.15.3.145)
### Table 4. Parameter difference between Bayesian learning step and conceptual population

| Keyword   | Diff. Bayes | Diff. MCMC |
|-----------|-------------|------------|
| application | 0.1149      | 0.0001     |
| circuit    | 0.1371      | 0.0002     |
| code       | 0.0165      | 0.0006     |
| component  | 0.2314      | 0.0001     |
| computer   | 0.1894      | 0.0001     |
| connector  | 0.1592      | 0.0001     |
| device     | 0.1039      | 0.0001     |
| display    | 0.1014      | 0.0001     |
| graphics   | 0.1116      | 0.0004     |
| image      | 0.1238      | 0.0000     |
| information | 0.1094     | 0.0003     |
| interface  | 0.0596      | 0.0004     |
| location   | 0.0838      | 0.0001     |
| media      | 0.0767      | 0.0001     |
| memory     | 0.2045      | 0.0000     |
| mobile     | 0.0378      | 0.0000     |
| portable   | 0.1898      | 0.0000     |
| power      | 0.1542      | 0.0001     |
| receiving  | 0.0111      | 0.0002     |
| remote     | 0.1553      | 0.0003     |
| request    | 0.5552      | 0.0005     |
| screen     | 0.1915      | 0.0003     |
| set        | 0.0811      | 0.0001     |
| storage    | 0.1900      | 0.0003     |
| user       | 0.1041      | 0.0001     |
| video      | 0.1145      | 0.0001     |
| voltage    | 0.1394      | 0.0001     |
| window     | 0.0065      | 0.0001     |
| wireless   | 0.0000      | 0.0004     |
| Absolute mean | 0.1294  | 0.0002     |

MCMC, Markov chain Monte Carlo.

---

Sunghae Jun is professor in the Department of Statistics, Cheongju University, Chungbuk, Korea. He received B.S., M.S., and Ph.D. degrees from Department of Statistics, Inha University, Incheon, Korea, in 1993, 1996, and 2001, respectively. He also received Ph.D. degree from the Department of Computer Science, Sogang University, Seoul, Korea, in 2007, and Ph.D. from Information Management Engineering, Korea University, Seoul, Korea, in 2013. He was visiting scholar in the Department of Statistics, Oklahoma State University, Oklahoma, USA, during 2009-2010. His current research interests include big data learning and artificial intelligence (AI).

Phone : +82-43-229-8205  
Fax : +82-43-229-8432  
E-mail : shjun@cju.ac.kr

---

[13] R. Zhao, D. W. Lee, and H. K. Lee, “Fuzzy logic based navigation for multiple mobile robots in indoor environments,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 15, no. 4, pp. 305-314, 2015. [http://dx.doi.org/10.5391/IJFIS.2015.15.4.305](http://dx.doi.org/10.5391/IJFIS.2015.15.4.305)

[14] S. B. McGrayne, *The Theory That Would Not Die: How Bayes’ Rule Cracked the Enigma Code, Hunted Down Russian Submarines & Emerged Triumphant from two Centuries of Controversy*, New Haven, CT: Yale University Press, 2011.

[15] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL: CRC Press, 2013.

[16] J. K. Kruschke, *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*, 2nd ed. London: Elsevier, 2015.

[17] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*, 2nd ed. Boca Raton, FL: CRC Press, 2011. [http://dx.doi.org/10.1201/b10391](http://dx.doi.org/10.1201/b10391)

[18] S. Jun and S. S. Park, “Examining technological innovation of apple using patent analysis,” *Industrial Management & Data Systems*, vol. 113, no. 6, pp. 890-907, 2013. [http://dx.doi.org/10.1108/IMDS-01-2013-0032](http://dx.doi.org/10.1108/IMDS-01-2013-0032)

[19] J. M. Kim and S. Jun, “Graphical causal inference and copula regression model for apple keywords by text mining,” *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 918-929, 2015. [http://dx.doi.org/10.1016/j.aei.2015.10.001](http://dx.doi.org/10.1016/j.aei.2015.10.001)

[20] I. Feinerer, K. Hornik, and D. Meyer, “Text mining infrastructure in R,” *Journal of Statistical Software*, vol. 25, no. 5, pp. 1-54, 2008. [http://dx.doi.org/10.18637/jss.v025.i05](http://dx.doi.org/10.18637/jss.v025.i05)

[21] I. Feinerer and K. Hornik, “Package ‘tm’: Text Mining Package ver. 0.6-2," Available [https://cran.r-project.org/web/packages/tm/tm.pdf](https://cran.r-project.org/web/packages/tm/tm.pdf)