Entropy Studies of Software Market Evolution

Liguo Yu, Department of Computer and Information Sciences, Indiana University South Bend, IN 46634, U.S.A., ligyu@iusb.edu
S. Ramaswamy, Department of Computer Science, University of Arkansas at Little Rock, Little Rock, AR 72204, U.S.A., srini@acm.org
R.B. Lenin, Department of Computer Science, University of Arkansas at Little Rock, Little Rock, AR 72204, U.S.A., rblenin@ualr.edu

Abstract

Entropy is a concept commonly used in various disciplines. In thermodynamics, it is used to measure the randomness of particles (atoms, molecules, etc.). In information theory, it is used to measure the amount of information in a program unit (variable, expression, function call, etc.) In this paper, entropy is used to measure the diversity of a marketplace. The formulas used in thermodynamics and information theory to calculate entropy is adapted for market analysis. Based on the theory and observations of thermodynamics, we hypothesize that the diversity/entropy of certain market tends to increase over time. The case studies of the evolution of programming languages, operating systems, web browsers, and web servers verify the hypothesis.

Keywords: Entropy, market entropy, market share, programming languages, operating systems, web servers, web browsers

1. Introduction

The concept of entropy originates in thermodynamics [1]. It is used to measure the disorder of particles present in a system. The particles could be atoms, molecules, plasma, etc. In information theory, entropy is used to measure the amount of information contained in a program unit, such as variable, expression, function, and query [2]. On the one hand, because the behavior of marketplace is similar to a thermodynamic system, entropy could also be utilized in market analysis [3]. For example, Sandroni [4] used market entropy to represent the accuracy of agents beliefs and found belief accuracy is the critical factor in determining survival of trading. McCauley [5] used the empirical market distribution to represent an asset's entropy and found that financial markets are unstable and accordingly do not behave thermodynamically. On the other hand, because information plays a key role in financial market, Shannons entropy theory is also utilized in financial market analysis. For example, Theil [6] applied information theory to problems in economics, such as the measurement of income inequality, industrial concentration, and concentration in international trade. Maasoumi and Racine [7] applied entropy metrics to examine the predictability of stock market returns and found that the entropy metric is capable of detecting nonlinear dependency within the returns and nonlinear affinity between the returns. Chen [8, 9] showed that most empirical evidences about investor performance and market behavior can be explained by an entropy-based information theory.

In the software industry, market share refers to the proportion of the total available market that is being serviced by a software product. It is represented as the percentage of a software product unit sales volume over the total volume of units of the same type of products in the marketplace. Market share analysis is a critical part of a company’s planning activities, which is useful to help a company in deciding strategies on inventory, purchase, work force and facility expansion/contraction, and so on. [10]. The market share of a software product is dependent on many factors, such as the quality and the price of the product, the service and the marketing strategies of the company, and the symbiotic relations between software products ([11-14]).

The objective of this study is to adapt the concept of entropy that is widely used in thermodynamics, information theory, and financial market to software market analysis. The remaining of the paper is organized as follows. Section 2 discusses the concept and the metric of market share entropy. Section 3 describes the case studies. Conclusions are presented in Section 4.

2. Market Share Entropy

Let a marketplace, \( M \), has \( n \) products, \( m_1, m_2, \ldots, m_n \), each with market share \( p_1, p_2, \ldots, p_n \), respectively. The entropy of marketplace \( M \) is defined as

\[
E(M) = -\sum_{i=1}^{n} p_i \log_2 p_i.
\]  (1)

Equation (1), known as the entropy function, is first introduced by Boltzmann in the 1870s, is also the general formula for information [2]. The entropy (E) in formula (1) represents the diversity of a marketplace. For example, if a marketplace is dominated by a small number of products, i.e., a few products account the major market share, the entropy (E) value of this market place is low, which means the diversity of this marketplace is low; if a marketplace is evenly shared by large number of different products, the entropy (E) value of this market place is high, which means the diversity of this marketplace is high.
If all the $n$ products of the market $M$ have equal share, then

$$p_i = \frac{1}{n}, \quad i = 1, 2, 3, \ldots, n. \quad (2)$$

For this case, we have from (1),

$$E(M) = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = -n \log_2 \frac{1}{n} = \log_2 n.$$

Equation (3) gives the upper bound for the entropy function.

Consider a marketplace with four products, Product 1, Product 2, Product 3, and Product 4. Let us examine four different distributions of market share as shown in Table 1. Distribution 1 has the least diverse market; the market share is unevenly occupied by Product 1 and Product 2. Distribution 2 is more diverse than Distribution 1 in that the market share is evenly occupied by Product 1 and Product 2. Distribution 3 is even more diverse than Distribution 2 in that the market is shared by three products, Product 1, Product 2, and Product 3. Distribution 4 is most diverse: the market is evenly shared by all four products. The entropy of each distribution is calculated and it shows that market share entropy increases as the diversity of the marketplace increases. Therefore, we can use market share entropy ($E$) to represent and measure market diversity.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Products</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75% 25% 0 0</td>
<td>0.81</td>
</tr>
<tr>
<td>2</td>
<td>50% 50% 0 0</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>50% 25% 25% 0</td>
<td>1.50</td>
</tr>
<tr>
<td>4</td>
<td>25% 25% 25% 25%</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Table 1: The diversity of market place and the market share entropy

3. The Evolution of Software Market Share Entropy

In this paper, we study the market share of various programming languages, programming language is not an application software product, it can be considered as a software tool. Therefore, in this paper, programming languages are also considered as software. The market share data of programming language is collected from [15]. The market share data of operating systems and web browsers is collected from Net Applications [16]. The market share data of web servers is collected from Web Server Survey [17].

3.1. General results

Figure 1 through Figure 6 show the evolution of the market share of major programming languages, major operating systems, web browsers, and web servers, respectively, in which, the data of operating systems is represented in two figures, Figure 2 and Figure 3. It should be noted that the market share is measured as the percentage of an observed market that is being serviced by one product. The accumulation of all the products shown in one figure represents one observed marketplace.

For programming languages, Java, C, C++ have a general decreasing trend; Delphi, Javascript, Python, and C# have a slightly increasing trend; while PHP, VB (Visual Basic), and Perl are relatively stable. For operating systems, Windows in general has decreasing trends, except Vista; operating systems have an increasing trend. For web browsers, IE’s (Internet Explorer) market share is decreasing while others are increasing. For web servers, Apache is decreasing while IIS (Internet Information Services) is increasing.

From Figures 1 to 5, we can see that in general the market share of a certain software marketplace is becoming more and more evenly distributed among different products. To quantitatively measure this trend, the evolution of market share entropy of each marketplace (programming languages, operating systems, web browsers, and web servers) is studied. The results are shown in Figures 7 to 10. The entropy results are normalized by dividing the values by the upper bound (3).

From Figures 1 to 5 it is clear that market share of programming languages, web browsers, and web servers are more diversified with time. As a result, their respective entropies in Figures 6 to 8, increase with time. The domination of the market by one or few products is more and more difficult. For operating systems, since the release of Windows Vista in 2007, the Windows market share is no longer dominated by and the entire market share is becoming more and more diverse.

Figure 1: The evolution of the market share of major programming languages
Figure 2: The evolution of the market share of major operating systems (Part 1)

Figure 3: The evolution of the market share of major operating systems (Part 2)

Figure 4: The evolution of market share of web browsers

Figure 5: The evolution of market share of web servers

Figure 6: The evolution of the market share entropy of major programming languages

Figure 7: The evolution of the market share entropy of major operating systems
3. Conclusions

In this paper, we adapted the entropy concept from thermodynamics and information theory and applied in software market share analysis. Our study found that entropy in market share analysis can be used to represent the market diversity. The case studies on the market share of programming languages, operating systems, web browsers, and web servers showed that in general, these software markets are becoming more and more diverse.

This paper is an attempt to demonstrate the applicability of entropy studies in software evolution. Software companies that have the ability to collect more fine grain data would be able to use this work, adapting it to analyze their respective products.

Acknowledgments

This work is based in part, upon research supported by the National Science Foundation (under Grant Nos. CNS-0619069, EPS-0701890 and OISE 0729792), NASA EPSCoR Arkansas Space Grant Consortium (#UALR 16804) and Acxiom Corporation (#281539). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

The authors would like to thank the anonymous reviewers for their detailed comments.

References
