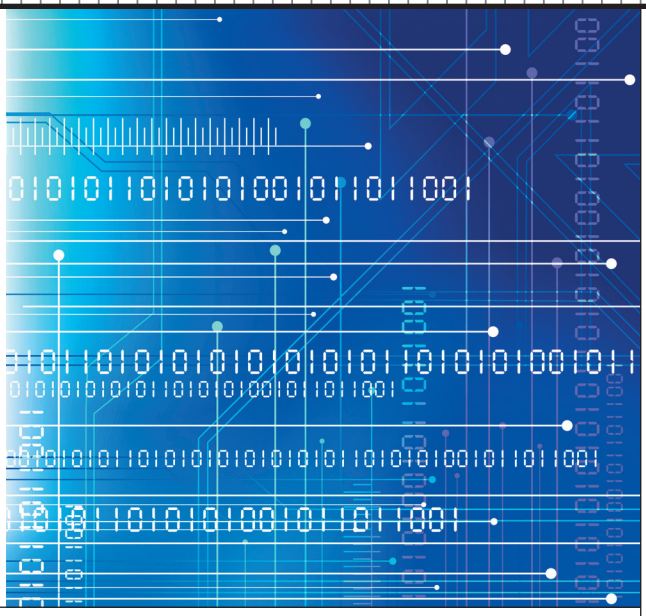
Using Data Mining to Help Design Sustainable Products



Manish Marwah, Amip Shah, Cullen Bash, and Chandrakant Patel, *HP Labs*
Naren Ramakrishnan, *Virginia Tech*

Data mining techniques make it possible to automate product life-cycle assessment with reasonable accuracy, even in cases of low-quality inventory data.

When making purchases, customers increasingly consider products' environmental impact as well as traditional criteria such as cost and features. In 2008, market research firm Gartner estimated that about 75 percent of enterprises would include some type of life-cycle environmental assessment in their purchasing decisions about future IT systems (D. Plummer et al., "Gartner's Top Predictions for IT Organizations and Users, 2008 and Beyond," report G00154035, 8 Jan. 2008). The recent surge in ecolabels and green stickers also indicates growing public sentiment that consumer offerings should meet minimum sustainability requirements or limit harm to the environment.

A product's environmental footprint is typically estimated through *life-cycle assessment* (LCA), which takes a comprehensive view of multiple environmental impacts such as greenhouse gas emissions, toxicity, and carcinogenicity (A. Shah et al., "Assessing ICT's Environmental Impact," *Computer*, July 2009, pp. 91-93). Researchers could use LCA to answer questions such as: How do Apple iPad, Samsung Galaxy Tab, or HP TouchPad compare in terms of their carbon footprint? Is an e-reader more environmentally friendly than a paper book? (D. Goleman and G. Norris, "How Green Is My iPad?" *The New York Times*, 4 Apr. 2010).

However, LCA can be a manual and laborious process. Accurately estimating the environmental impact factors associated with a server, for example, may involve creating a detailed inventory of all its components, usually down to parts such as integrated circuits (ICs), resistors, fans, heat sinks, and even screws, paint, and labels; estimating their mass or volume; and, finally, mapping each component to representative entries in an environmental database.

We treat LCA as a data mining problem and propose an automated LCA (auto-LCA) approach that lets a user simply input on existing product inventory and obtain an approximate environmental footprint of all its components as output.

AUTO-LCA

Given the large-scale manual processing of semistructured data associated with performing LCA, we redefine LCA as a data mining problem and integrate data mining solutions from different contexts to obtain an auto-LCA methodology.

We consider a product inventory, such as a bill of materials (BOM), to be a compositional containment hierarchy and represent it as a tree. For example, a desktop computer contains a printed circuit board (PCB), which contains ICs, capacitors, and resistors; these components in turn contain silicon and other metals.

An auto-LCA system requires two databases: a product database, which includes each product's BOM as well as information such as part number and description; and an environmental database of generic information about the environmental impact of various components—for example, the Ecoinvent database (www.ecoinvent.ch).

As Figure 1 shows, the path from BOM to environmental footprint

0018-9162/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

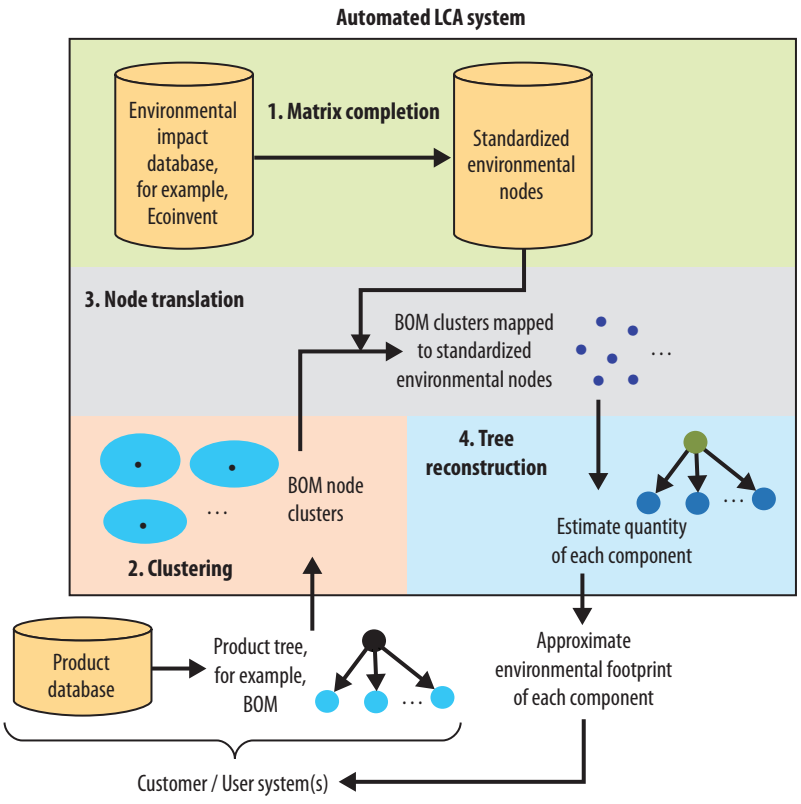


Figure 1. Automated life-cycle analysis (auto-LCA) methodology. Obtaining a product's approximate environmental footprint from its bill of materials (BOM) entails four main steps: matrix completion, clustering, node translation, and tree reconstruction.

entails four main steps: matrix completion, clustering, node translation, and tree reconstruction.

Matrix completion

We first check the environmental database for incomplete or invalid information. For example, many such databases only have a few impact factors, such as energy and ecotoxicity, listed for certain nodes, and lack relevant information such as carbon footprint data or quantified impacts on human health. The environmental database can be viewed as a matrix with components as rows and impact factors as columns. Estimating missing values thus constitutes a matrix completion task. This problem is similar to that encountered in recommender systems, in which the goal is to estimate missing ratings for unseen items such as movies and books.

Clustering

Next, we perform cluster analysis on the potentially hundreds of items listed in the BOM with the objective of grouping similar nodes—those likely to have comparable environmental impact—together. The clustering algorithm requires a distance metric computed from the node attributes to posit groupings. Simply using part numbers isn't sufficient, as many BOM components could be quite similar from the standpoint of environmental impact but very different in terms of how they're identified in the product tree. For example, two identical stainless steel screws that reside in different parts of the system might have distinct part numbers.

To compute the distance between node descriptions, we use approximate string-matching techniques such as longest common subsequence (LCS), longest common prefix (LCP), Levenshtein distance (LD), or a combination of these. Once we obtain a distance metric, we employ clustering algorithms such as *k*-medoids to group similar BOM nodes together. The resulting clusters reduce the number of parts to be evaluated from up to several thousand to a smaller, more manageable number—typically, at least an order of magnitude lower.

Node translation

We then assign each of these clusters a representative node similar to its medoid from the environmental database. In this way, we "translate" BOMs associated with distinct products that come from various suppliers and have different naming schemes into a standard terminology derived from the environmental database, thereby yielding insight into the environmental impact related to each cluster. Ideally, such translation would be learned based on some training data or by comparing BOM and environmental node descriptions, but we currently perform this translation manually. It's worth noting that clustering allows such manual translation, as it reduces the number of required translations by more than an order of magnitude.

Tree reconstruction

A challenge for translation is that the units specified in the BOM nodes and the environmental database nodes can differ. For example, most product BOMs specify the number of repeating instances for a particular part, while the environmental nodes could be specified by mass (kg). To rectify this, we use the property that for any environmental impact, the sum of the child node impact values approximately equals that of the parent (root), forming a linear system of simultaneous equations with the coefficients of the child nodes being the unknowns. To fully reconstruct the BOM tree comprising environmental nodes, we must estimate these coefficients.

We perform a least squares regression fit to best estimate the coefficients. Because the coefficients must be positive, the goal is to obtain a non-negative least squares (NNLS) fit (C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems*, Society for Industrial and Applied Mathematics, 1995). Knowing a single node's weight (the root or one of the child nodes) lets us compute the environmental contribution of each child node to the total (parent) impact.

CASE STUDY: SERVER PCB

With these building blocks in place, it's possible to estimate the environmental footprint of an arbitrary product tree or BOM. We illustrate the approach by analyzing a real PCB from an enterprise server. This PCB BOM contains about 560 components, including a mix of resistors, capacitors, ASICs, and logic devices.

We cluster the BOM nodes using *k*-medoids to identify 22 unique clusters. Figure 2 shows two examples. It's relatively easy to translate these clusters (actually their medoids) into a list of nodes from the environmental database. For the resulting environmental tree, we can utilize the impact factors available from the environmental database and successfully solve for the coefficients of the child nodes. Figure 2 includes the resulting coefficients of select nodes in the environmental tree, which enables readily computing each child node's environmental impact. The median error between the sum of the child node impact factors and the parent, for about 200 impact factors, is only 12.8 percent, a highly satisfactory result.

After obtaining each child node's environmental footprint, we perform an environmental "hotspot" analysis. This essentially involves generating a Pareto list of the largest environmental contributors to the overall PCB footprint so that a designer or LCA practitioner can focus on those areas requiring further effort.

As Figure 3 shows, due to their upstream manufacturing, ICs are the biggest contributor to the PCB's overall carbon footprint, followed by the use of copper in the connectors and capacitors. These automated results match those obtained from a manual LCA, and are easy to under-

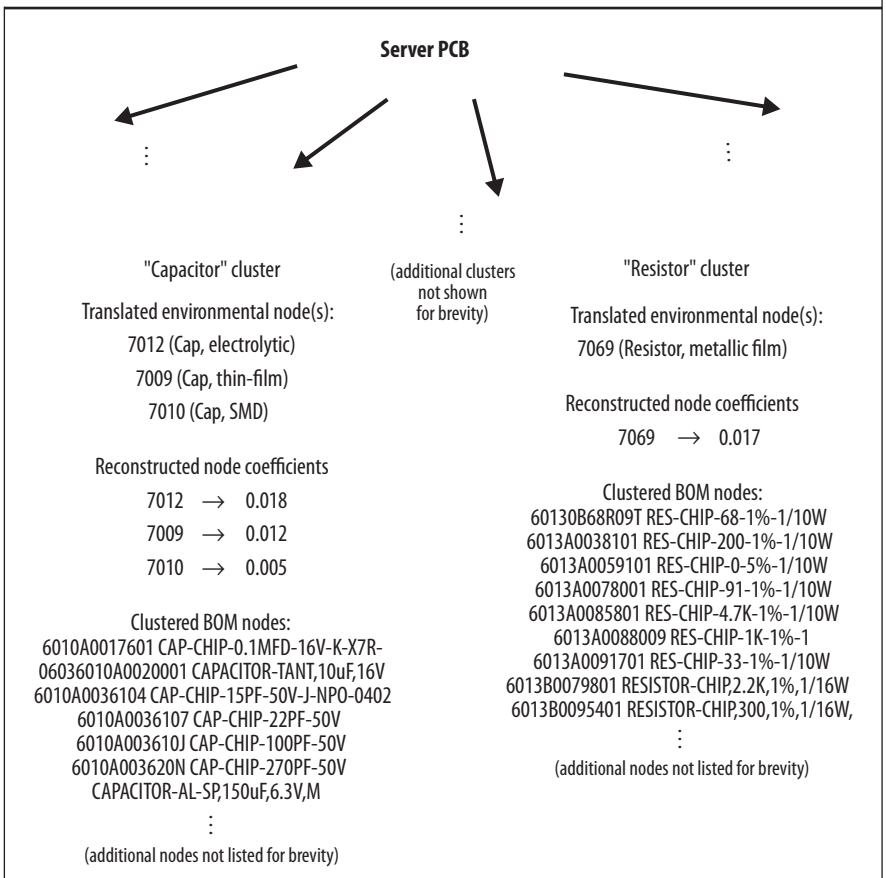


Figure 2. Server printed circuit board (PCB) clusters, node translations, and coefficients.

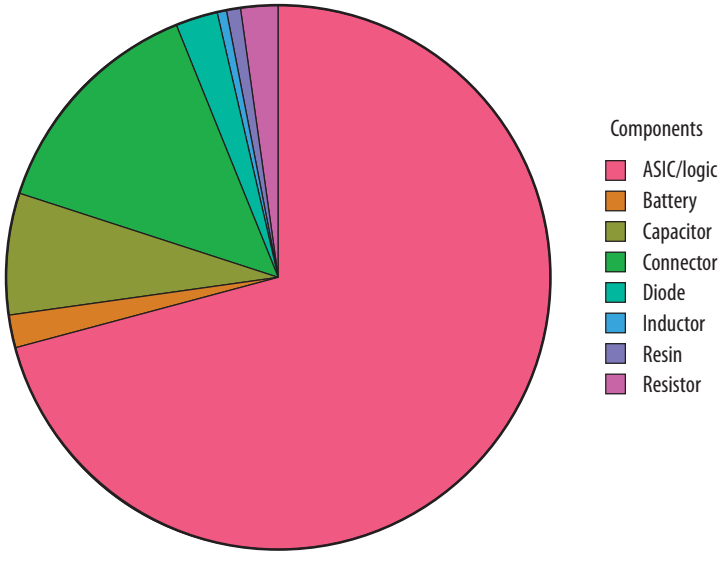


Figure 3. Results of environmental "hotspot" analysis. ICs are the biggest contributor to the PCB's overall carbon footprint.

stand even for those unfamiliar with LCA or environmental impact analysis.

PCB designers can use our auto-LCA approach to assess their design's sustainability against that of other designs as well. We anticipate eventually creating a tool that automatically scans similar ICs preloaded into the environmental database to aid in this assessment.

Consumers as well as enterprises today demand more information about products' environmental impact. Data mining techniques such as matrix completion, clustering, node translation, and tree reconstruction make it possible to automate LCA with reasonable accuracy and within fairly broad constraints, even in cases of low-quality inventory data. In the future, we plan to further evaluate these algorithms' scalability and test auto-LCA on a wider variety of systems.

Manish Marwah *is a senior research scientist at HP Labs, Palo Alto, California. Contact him at manish.marwah@hp.com.*

Amip Shah *is a senior research scientist at HP Labs, Palo Alto, California. Contact him at amip.shah@hp.com.*

Cullen Bash *is a distinguished technologist at HP Labs, Palo Alto, California. Contact him at cullen.bash@hp.com.*

Chandrakant Patel *is a senior fellow at HP Labs, Palo Alto, California. Contact him at chandrakant.patel@hp.com.*

Naren Ramakrishnan, *the Discovery Analytics column editor, is a professor of computer science at Virginia Tech. Contact him at naren@cs.vt.edu.*