



IN THE NEWS

Features Editor: **Brian Brannon**, bbrannon@computer.org

AI Heralds a New Musical Age

Mark Ingebretsen

In an era when people carry thousands of songs in their MP3 players or role-play as rock stars via video games, AI stands poised to transform how we interact with music in even more dramatic ways. Recently developed software applications that rely on AI tools such as machine learning and Markovian analysis are providing new ways to analyze music, arrange

playlists, and even devise customized accompaniments to user-created melodies.

"A lot of the projects that we do involve simply taking aspects of what humans are able to do and trying to develop algorithms that can mimic that ability, so a human listener or player-improviser can mentally organize the music in certain ways so as to better understand or to manipulate it," says Elaine Chew, a concert pianist and engineer who heads the Music Computation and Cognition Laboratory (www-rcf.usc.edu/~mucoaco) at the University of Southern California's Viterbi School of Engineering.

Personalized Radio

Indeed, the marriage of AI and music has already enjoyed widespread popularity. For proof, just ask some of the more than 20 million users of Pandora.com. The eight-year-old Oakland, California-based service lets listeners create personalized online radio stations. Select the artist or song you like, and the company's proprietary algorithms scour its library to build a playlist with a similar style. Click the thumbs-up or thumbs-down icons as each new song begins playing, and you can continually refine the kinds of songs you hear.

Pandora's elegantly simple interface belies the complexity of the system driving it. As Tim Westergren, the service's founder, explains, Pandora is built atop what he terms the music genome, an extensive set of tags assigned to each song. To create the genome, human evaluators, who like Westergren are musicians themselves, review each piece in Pandora's library on the basis of 400 musical attributes.

Once the attributes are cataloged, the system's AI components take over. "Each of the 400 attributes has a weight" assigned to it, Westergren says. For example, the amount of vibrato a lead vocalist uses might receive a lower weight than the song's tempo. Over time, "that weighting vector will get altered based on the feedback of a particular listener and the aggregate feedback of everyone who has listened to the song," he says.

Thus, as Pandora's system dutifully catalogs every piece of user input, employing a process known as contextual collaborative filtering, the system can constantly improve its ability to match musical tastes with selections from the service's library. "We know what the thumbs-up, thumbs-down is for every station on which a particular song plays," Westergren continues. If listeners who have created rap music stations overwhelmingly give a thumbs-down to a particular song, the algorithm will begin playing that song less frequently on all rap stations. Likewise, songs that receive many thumbs-up responses will play more often. However, individual listeners can always block or hot-list particular songs on their own stations.

See the Music

Pandora's trained music analysts can spend up to 30 minutes applying musical tags to one song. But competing applications employ AI tools that perform comparable evaluations automatically. One example is Visualizing Music, an application that

Also Featured

**Multiagent Designs
Could Safeguard Networks
across the Web**

How to Reach Us

Writers

For detailed information on submitting articles, write for our Editorial Guidelines (isystems@computer.org) or access www.computer.org/intelligent/author.htm.

Letters to the Editor

Send letters to

Brian Brannon, Lead Editor
IEEE Intelligent Systems
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
bbrannon@computer.org

Please provide an email address or daytime phone number with your letter.

On the Web

Access www.computer.org/intelligent for information about IEEE Intelligent Systems.

Subscription Change of Address

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify IEEE Intelligent Systems.

Membership Change of Address

Send change-of-address requests for the membership directory to directory.updates@computer.org.

Missing or Damaged Copies

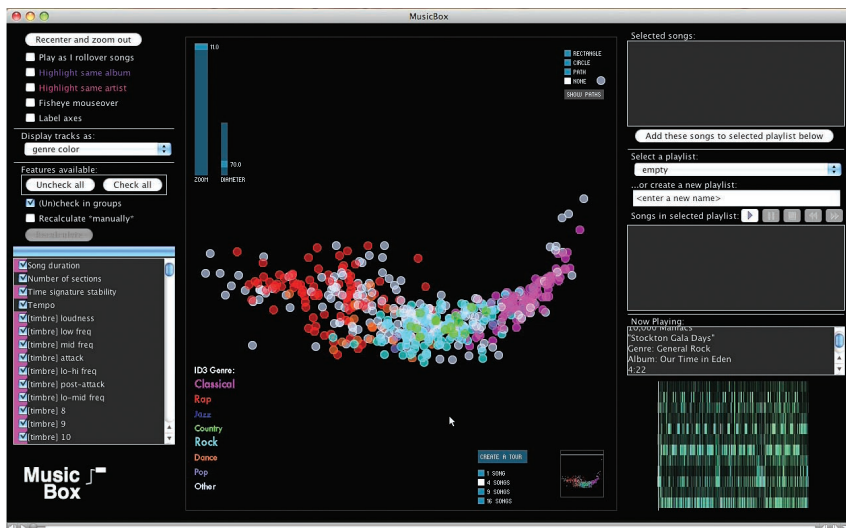
If you are missing an issue or you received a damaged copy, contact membership@computer.org.

Reprints of Articles

For price information or to order reprints, email isystems@computer.org or fax +1 714 821 4010.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at copyrights@ieee.org.



Anita Lillie's MusicBox program. Anita Lillie created MusicBox while attending MIT's Media Lab. The application analyzes a user's library of songs, then automatically catalogs and displays those songs according to their musical genre.

Anita Lillie completed while pursuing her master's at MIT's Media Lab.

Lillie's software scans each song and categorizes it according to 300 attributes, a process that takes roughly 20 seconds. As she explains, "The analysis consists of things that people wouldn't be very good at, such as the frequency spectrum of a particular piece of music or a quantitative description of the rhythm."

Chew notes that this kind of analysis parallels what linguistic researchers have long employed with literature. Works of Shakespeare, for instance, exhibit particular patterns when scrutinized by AI tools. Once an algorithm understands the patterns of Shakespeare's writing, it can calculate the probability that a newly introduced work was also written by the Bard.

"The calculation of musical similarity often employs data-driven machine intelligence," says Chew. Similarly, "in music, the data could reveal the spectrum of tonal patterns that a composer tends to use." A trained algorithm can then tell you the likelihood an unknown work was from the same composer.

Lillie's Visualizing Music application depicts a song's attributes in a patchwork diagram resembling a genetic sequence (www.flyingpudding.com/projects/viz_music). An observer well versed in Lillie's application might be able to tell whether a particular pattern represents a heavy metal or new age composition. Chew notes that visual representations of music can supplement centu-

ries-old methods of analyzing and writing music, revealing nuances impossible to convey via notes on a page or in a scale.

Welcome to My Library

Lillie's gene-sequence-like diagrams can also help anyone with a sizeable music library. MusicBox, another application that Lillie developed as part of her master's thesis project, aggregates the Visualizing Music-assigned metadata from each composition and ferrets out similarities among groups of songs in a library (see the figure). Just as Pandora's evaluators assign a weight to individual features of the music, Lillie's system, which runs on a stand-alone PC, lets users emphasize one attribute over another.

These user-supplied rules prompt her software to depict an entire music library as a series of dots. Dots close to one another represent songs that share musical attributes, while those at opposite ends of the array denote vastly different genres, such as techno and classical. Users can create playlists of similar music simply by highlighting a collection of dots from one segment of the array. "You can define which components contribute to the categorization of your music by turning them on and off. If you are interested only in tempo and bass, then you can view your music library organized simply on the basis of those components," Lillie says.

Lillie hopes to adapt her software for MP3 players so that listeners can create playlists away from their PCs. Another goal would be to embed the system into a much

larger library such as those found on music vending sites. That could mean developing other visualization methods such as clouds or different colors and shapes, she says.

Make Me a Song

Arranging musical collections and playlists is just one example of how AI promises to transform music. Chew and one of her students at USC, Ching-Hua Chuan, have developed an application that designs a customized chord accompaniment to a melody input by a user. The chord progressions Chuan and Chew's system outputs can also mimic a particular musician's style.

Chuan, now an assistant professor of computer science at Barry University, got the idea for the program when she played guitar in an all-female rock band in Taiwan. Band members had a hard time using only words to describe the style of music they wanted their cohorts to play. But simply playing a recording of a song in that style intuitively conveyed the message without a problem.

The experience inspired Chuan to devise a system to identify important features of a reference composition. Then, when provided with a new melody, her system generated an accompaniment in the style of the reference piece.

The Chuan-Chew system relies on neo-

Riemannian transformations, a method of graphically representing the relationship between notes in a composition. Next, a machine-learning-based decision tree determines which notes in a user-supplied melody to include in the chords forming the accompaniment. Then, the system employs Markovian analysis to calculate the probability that a certain chord progression will create a pleasing accompaniment. In a final step, the system picks out the progression with the highest probability.

The underlying training process differs significantly from how AI algorithms typically learn, because the fewer examples the Chuan-Chew program receives, the better the accompaniment emulates a desired style. In contrast, Chuan says, "If you put too many songs into the system, then you just generate a regular piece without character." The neo-Riemannian transformations work best with rock music, because rock chords are fairly standardized in their relationship to one another. The extended chords used in, say, a jazz composition might create a less-pleasing accompaniment and require more complex representations.

Searching for Boundaries

Chuan successfully tested the software with students, who weren't able to identify

whether an accompaniment was created by the computer or the UK rock band Radiohead, which served as a reference (www.scf.usc.edu/~ise575/c/projects/chuan).

For now, Chuan and Chew's software is limited to simple MIDI instrument sound output. However, a little tweaking might allow someone with limited musical training to overlay more complex audio accompaniment to create a work embellished with voices and wind or string instruments—and all from a simple melody. Skilled musicians might even create variations on the works of famed composers.

Purists might cringe at the idea of tampering with music's greatest works. But Chew, a formally trained musician herself, believes the marriage between music and AI is a healthy one. "The training of musicians, particularly that of performers, often aims to preserve tradition," she says. "Many are taught to emulate the masters, ideas, and styles that are already out there. A scientific approach, where we are constantly trying to find the boundaries and create new ones, can imbue this august tradition with a spirit of discovery and pave new paths for the future."

Multiagent Designs Could Safeguard Networks across the Web

Mark Ingebreitson

As hackers get smarter and computer networks become more intertwined, some researchers believe multiagent systems offer the best way to provide universal protection against security threats.

Designs by network researchers are still largely in prototype form. Nevertheless, they suggest that multiagent systems could harness AI tools such as neural networks to analyze traffic and help determine whether requests are normal or dangerous anomalies. Some designers envision giving agents the ability to access libraries of known computer viruses and attack methods. Armed

with that knowledge, the agents could then employ Markov decision processes or other means to determine whether an attack is in progress.

Individual agents might also transmit vital information about the method of attack in real time to other agents throughout the network. Eventually, agents operating autonomously—either solo or as a group—

might opt to take defensive actions such as tightening security protocols, isolating infected computers, or even acting in concert to quash an attack.

When deployed, such multiagent designs would be in stark contrast to today's network security systems that aim primarily to protect a single network. Currently, widespread sharing of information about attacks among systems operators is typically ad hoc and often occurs only after a serious invasion has taken place. By that time, the virus could have already spread to an enormous number of computers.

A Computer Security Interpol

Exactly what might a multiagent network look like, and how would it be organized? José M. Vidal, associate professor at the University of South Carolina's Computer Science and Engineering Department, believes the first stage would be an "open system where agents act like sensors in a dis-

**IEEE Computer Society
Publications Office**

10662 Los Vaqueros Circle, PO Box 3014
Los Alamitos, CA 90720-1314

Staff

Lead Editor

Brian Brannon

bbrannon@computer.org

Senior Editorial Services Manager

Crystal R. Shif

Magazine Editorial Manager

Steve Woods

Staff Editors

**Dale Strok, Dennis Taylor,
and Linda World**

Assoc. Peer Review Manager

Hilda Carman

Publications Coordinator

Alkenia Winston

Production Editor

Jennie Zhu

Technical Illustrations

Alex Torres

Director of Products & Services

Evan Butterfield

Digital Library Marketing Manager

Georgann Carter

Senior Business Development Manager

Sandra Brown

Senior Advertising Coordinator

Marian Anderson

Submissions: For detailed instructions and formatting, see the author guidelines at www.computer.org/intelligent/author.htm or log onto *IEEE Intelligent Systems'* author center at Manuscript Central (www.computer.org/mc/intelligent/author.htm). Visit www.computer.org/intelligent for editorial guidelines.

Editorial: Unless otherwise stated, bylined articles as well as products and services reflect the author's or firm's opinion; inclusion does not necessarily constitute endorsement by the IEEE Computer Society or the IEEE.

tributed sensor network." The agents would be empowered to publish or relay important information to each other.

After the stage-one system has proven reliable, "we can start thinking about having the agents act on that information: shutting down rogue PCs, dropping packets, filtering certain protocols, applying patches, etc.," Vidal explains.

To succeed, a multiagent system would need to be widely propagated among cooperating organizations, researchers say. But owing to security concerns, these researchers agree that access shouldn't be as easy as downloading the latest browser or security patch.

"Similar to any type of software systems, it is important that their propagation is controlled and appropriate measures are in place to avoid propagation of malicious software agents," says Haralambos Mouratidis, a principal lecturer at the University of East London.

Vidal sees a model for how a multiagent security system might be organized in the way the Internet is currently administered. "The best way for multiagent security to be effective is by having one worldwide multiagent security protocol," he says. Organizations could freely opt to use the system and determine their level of participation.

For example, the simplest form of participation might be for Company X, let's say, to offer a REST (representational state transfer, http://en.wikipedia.org/wiki/Representational_State_Transfer) page. The page would provide data on the security status of the organization's internal network. "Data on the REST page would be used by agents on each machine, whether local or remote, to detect and stop security threats," Vidal says.

To satisfy privacy concerns, organizations closely tied to Company X might have access to all the information on the page, while other organizations—those that simply had opted into the multiagent protocol—might have access only to some of that information. That way, information about Company X's own security measures would be less vulnerable to hackers.

"Each organization must decide what information to make public, how to use information from others, and how to handle outside requests," Vidal says. Eventually, developers could expand the REST interface to permit other organizations to make reports or requests. For example, he says,

"an outside agent might ask another one to shut down a particular connection coming from its domain because it believes it to be a [denial-of-service] attack."

Multiagent Evolution

To be sure, much work must be done before a system that network operators everywhere trust can take hold. "Most of the research so far has been focused on issues such as making agents more secure [and] understanding how agents can be employed in various security-related scenarios," says Mouratidis, who is coauthor of the forthcoming book *Safety and Security in Multiagent Systems: The Early Years* (Springer). "More work is needed before we can say for sure that agents themselves are well protected."

Mouratidis lists several possible threats, starting with false alarms. Here, agents might misidentify a threat and cause a slow-down or halt in network traffic. Consider a banking network, for instance. If an agent mistakenly identified a threat and subsequently blocked all incoming messages, "then all legitimate incoming messages, possibly related to updating accounts, would also be blocked," he says. Worse yet, suppose hackers could gain control of the agents themselves and turn them against the systems they were supposed to protect.

In a paper published in the *International Journal of Software Engineering and Knowledge Engineering*, Mouratidis and his colleague, Paolo Giorgini, an assistant professor at the Department of Information and Communication Technology, University of Trento, write that developers can minimize these risks by building in security measures from the ground up when they're designing the multiagent systems (www.dit.unitn.it/~pgiorgio/papers/IJSEKE06-1.pdf). Today, in contrast, designers might add security mechanisms only after coding the agents' main functions.

Vidal believes additional protections would result from the system's distributed nature. Organizations could opt out of the system whenever they chose, and at whatever operational level they chose. But like Mouratidis, Vidal believes system designers can incorporate some of the best protection. "In open multiagent systems we strive to distribute power," he explains, "that is, to minimize the power of the most powerful agent in the system. In this way we also minimize the possibility of a catastrophe, either planned or accidental." ■