Web-Scale Multimedia Analysis: Does Content Matter?

Malcolm Slaney
Yahoo! Research and Stanford Center for Computer Research in Music and Acoustics

It pains me to ask this: what good is content analysis? I'm trained as an engineer, a signal processor, and a content person. I tackle problems in a systematic fashion, often bottom-up. When asked a content question, my first instinct is get out my fast Fourier transform (FFT) hammer.

The initial success of Web-image search was based exclusively on the text around an image. Certainly we have progressed since then. But recent research results dramatically beg to differ. For example, if you want to judge the similarity of two different pieces of music, should you look at the musical notes, or should you look at what people say about the music? Similarly, how should you find the best movie to recommend to a friend? Shouldn't the genre of the movie matter? Or when tagging a photo, is it better to look at the pixels, or where the picture lives on the Internet? I want to think that content matters, but in all three cases, metadata about the content proves to be more useful.

It's useful to look at several examples where content has lost out to other forms of data. These examples come from the worlds of music, movies, and images.

Music similarity

We often want to know when two pieces of content are similar, but I suspect this is inherently a personal decision. Soulful songs sung by Billie Holiday and Ella Fitzgerald are like night and day to a jazz lover; yet they are both elevator music to the punk rocker on the streets of London. What does it mean to be similar? Is this an AI-complete question (see http://en.wikipedia.org/wiki/AI-complete)? At a music-information-retrieval conference, I heard one (very dedicated) researcher say that she pondered the question of the similarity of two songs for hours.

In a study I performed a few years ago, we compared two different approaches for judging music similarity.[1] In the classic approach, we use music features that are often used to judge genre. The assumption is that if these features are good for making genre judgments, then they will also tell us something about similarity. This feature is known as a *genregram*.[2] The audio waveform is rich in information—it tells us everything we need to know about the music. In fact, listeners can tell whether they like a radio station within seconds of changing the dial.

The alternative is an item-to-item judgment based on user ratings. The idea considers each song as a point in a multidimensional space defined by a user's rating of the song. On a five-point scale, this is just 2.2 bits of information per user. If a jazz lover, a rock lover, and a classical lover all give two songs the same rating, then the two songs are probably quite similar (see Table 1).

In my study on similarity, I used the ratings by 380,911 listeners of 1,000 different songs. After adjusting for missing data, I formed a vector of all user ratings for each song. Song similarity was defined as the correlation between the user-rating vectors for the two songs.

I tested the two song-similarity approaches by starting with a seed song and forming playlists. In a blind test, users overwhelmingly said

Editor's Note

This is my first column as editor of the Visions and Views column. I hope that with this column I can bring you ideas that are new and interesting. I want to open a dialog about issues that are important to our field, and could change the direction of our work. I don't intend to write most of the columns, and I am soliciting ideas. Please send me your suggestions, either for issues you would like to discuss, or nominate a colleague who has new and interesting ideas that deserve broader attention.

—*Malcolm Slaney*

1070-986X/11/$26.00 © 2011 IEEE

that the songs on the playlist based on rating data were more similar to each other than those based on the genre space, or a random selection of songs. How can this be? Just 2.2 bits beat out a state-of-the-art system on the basis of content.

Movie recommendation

Netflix recently hosted a $1 million competition to find a better recommendation system for their movies. It's not an understatement to say that it captured the entire machine-learning community's interest. Thousands of hours of research, in all different directions, were directed at this problem.

While the identity of the users was unknown, the movie titles were not. Researchers quickly identified each movie and analyzed its content. It only makes sense that Alice, who loves romance movies, will like very different content from Bob, who likes action films. We should be able to use this information to build a better recommendation system.

But alas, content doesn't help. The winning systems included every possible signal.[3] Two features that surprised me were related to the time of the movie's release and the user's rating (See Figure 1). Evidently there is a strong correlation, with older movies getting a higher rating. In the final system, all available signals were combined using a machine-learning technique known as *boosting*. In boosting, various (weak) classifiers are combined to make a prediction (the movie's rating by a new user) if they reduce the error on an unseen test data set. Dozens of different features were included.

Table 1. Measuring similarity. Rating data is a good measure of song similarity. Shown here are the ratings that three different users give to three different songs. No matter what genre songs 1 and 3 come from, a large number of users have the same opinion of them, so the songs are likely to be similar.

User's music type	Song 1	Song 2	Song 3
Jazz	5	0	5
Rock	5	0	5
Classical	0	5	0

Not a single feature was derived from the movie's content. These were well-motivated researchers, with access to the best of the algorithms in the multimedia literature. But we couldn't help them. Arguably, the movie's genre was reflected in the rating data; yet, in the end, the FFT lost to star ratings.

Image tagging

Many multimedia problems are inherently a tagging problem. Is this music blues? Is this a picture of the Golden Gate Bridge? Is Sara

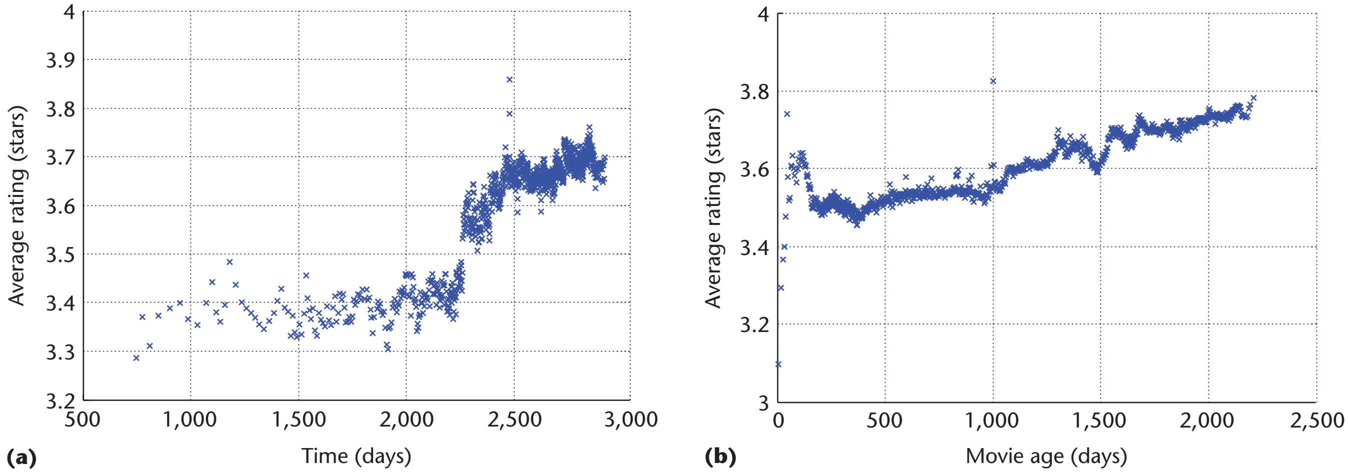
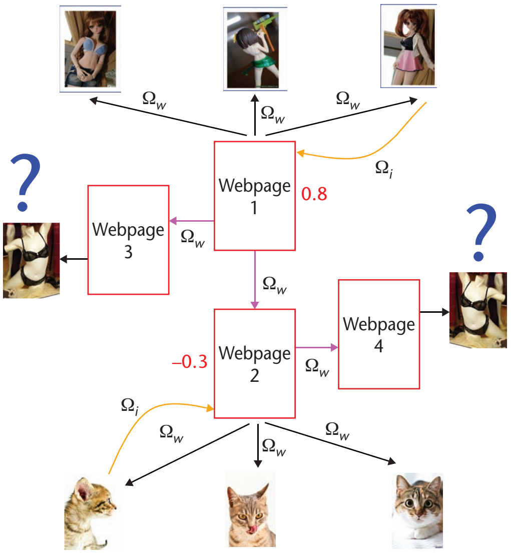


Figure 1. Temporal information in recommendations: the winners of the Netflix competition benefited from all sorts of metadata about the movies. These two images show the average movie rating as a function of (a) the date the rating was made and (b) the age of the movie when it was rated. For reasons that aren't known (perhaps a change in the user interface), the average rating increased by 0.2 stars about 2,300 days into the database. Likewise, the average rating a movie receives increased as the time between when the movie is released and when the movie is rated. Perhaps this is because people watch and rate only the "good" old movies. The effects in both cases are small, but both signals provide robust information to the collaborative filter engine and are included in the final recommendation engine. Conversely, information about the content of the movie was not as useful. (Data courtesy of Yehuda Koren.)

Figure 2. Labeling images based on the Web graph: the Web graph is an important signal to label images more effectively. The three images along the top are adult images; the three on the bottom are clearly not adult. The two images in the middle are unclear, but the Web graph can help make a decision. The Ω_w and Ω_i factors propagate information from the webpage to the image and back. Thus, webpage 1 has a score of 0.8 (adult) while webpage 2 has a score of −0.3 (safe). This information can tip the decisions for webpages 3 and 4. (Images used with permission from Flickr users redoxkun, toel-uru, RobW_, Boaz Arad, Sergiu Bacioiu, and OwenConti.)



in the picture? These are all relatively simple pattern-classification tasks that all come down to a binary decision based on the multimedia content. You would think that the pixels are the most important signal.

Mahajan approached image tagging by extending a successful approach from the world of spam. Spam email is difficult to judge. One person's spam is another person's ham. An important signal is the relationship between the sender and the receiver. First-generation spam detectors looked at the reputation of the sender—bad senders were often sending spam. But the relationship between email providers and spammers is adversarial and email accounts are cheap. Spammers quickly realized that they could create two accounts, send lots of email between them, mark each message as ''not spam,'' and get a good reputation. The newest spam detectors consider the entire network when judging the reputation of a sender.[4] We can't tell anything about Alice and Bob from the email they send each other. But if Charlie only sends email to Alice and Bob and never receives anything in return, then Charlie is suspect. This reputation can then propagate across the network to label all the email senders.

Likewise, the context of an image tells us a lot about what might be in the image. We like to treat multimedia classification as a simple problem: here is an image, does it show a telephone box? But in the real world, every piece of content has a context. At the very least, we know that a real person shot it (or a real person owned the camera). The image was uploaded to a website, and each website has a flavor. Photos on the ESPN website are very different from those at TMZ. Photos uploaded to Flickr are often more artistic than the people shots typical on Facebook. More subtlety, the friends of a person who takes lots of pictures of cats will probably have friends who like and take pictures of cats.

Mahajan took a collection of images from the Web and built a graph from their hyperlinks.[5] He defined an optimization equation that included three terms: a content-detector's decision about the image; a regularization term based on the decision for labeled images; and most importantly, a regularization term based on the decision made for nearby (on the network graph) images. The two regularization terms are important because they help propagate information from one image to another (See Figure 2). Regularization based on the information from labeled images encourages a form of semisupervised learning. Regularization based on the network graph means that the decision at one picture should be the same at other pictures on the same webpage, and similar to images at linked pages. By putting all three terms into an optimization framework, we find the decisions and slack variables that, given the learned hyperparameters, offer the best solution.

All three signals are important. But we were surprised that the single most important term for classifying the images was the network term. It was amazing to us that ignoring the content of the pictures left us with any signal at all. The strength of the signal from an image's neighbors is stronger than we expected. For many classes, including the adult content that Mahajan was detecting, a few pixels is all that separates one class from the other—is that swimsuit really covering all the right bits? It's difficult to find these distinctions in real-world images. Thus, the greater power of context is helping us make the right decision.

Conclusions

It's important to state that one can't prove a negative conjecture. We are asking whether we get more information from the pixels or from the metadata. We humans are exceptionally good at judging the content of a picture. Of course that is a cat. And if it's not something we can tell from a photo (such as whether the cat is male or female), we don't think it's an important problem. In many problems, we are asking our algorithms if they can tell a difference, and we can't say whether our insight into the problem is weak, or if the information is just not present in the signal.

As a content-analysis person, I would never argue that we should ignore the content. Yet there are many ways to solve a problem. We shouldn't overlook the rich metadata that surrounds a multimedia object.

A homework problem might contain only the pixels of an image, but the real world is not this simple. Every object comes with a context, and those who ignore this signal harm science and their chance of success. Representing and manipulating this extra data is difficult. In the image-tagging example, we had to work hard to find a subgraph of the Web that contained both positive and negative examples. Then, we had to further simplify the graph by combining neighboring nodes with no content to keep the optimization small enough to fit on a single CPU. (We could certainly do the computation on our grid, but it would have taken us longer to get our initial results.)

Content analysis is hard, perhaps even AI-complete. The future certainly will give us better feature analyzers and classifiers. Approaches to all three of the examples presented in this article would benefit from better content analysis. Yet, in the end, the signals provided directly by humans—whether they are stars or hyperlinks—tell us more about the content than our FFTs can. This is both depressing and exhilarating. But out with the old and in with the new.

We should all be asking ourselves how we can take advantage of human signals to understand multimedia more effectively. **MM**

References

1. M. Slaney and W. White, ''Similarity Based on Rating Data,'' *Proc. Int'l Soc. Music-Information Retrieval,* Int'l Soc. Music-Information Retrieval, 2007.
2. G. Tzanetakis, G. Essl, and P. Cook, ''Automatic Musical Genre Classification of Audio Signals,'' *Proc. Int'l Symp. Audio Information Retrieval* (ISMIR), Int'l Soc. Music-Information Retrieval, 2001.
3. Y. Koren, *The BellKor Solution to the Netflix Grand Prize,* 2009; http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf.
4. J. Abernethy, O. Chapelle, and C. Castillo, ''Graph Regularization Methods for Web-Spam Detection,'' *Machine Learning J.,* vol. 81, no. 2, 2010, pp. 207-225.
5. D.K. Mahajan and M. Slaney, ''Image Classification Using the Web Graph,'' *Proc. Int'l Conf. Multimedia,* ACM Press, 2010, pp. 991-994.

Contact editor and author Malcolm Slaney at malcolm@ieee.org.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.