2010 International Conference on Pattern Recognition

Canonical Patterns of Oriented Topologies

Walter C. Mankowski, Ali Shokoufandeh, Dario D. Salvucci
Department of Computer Science
Drexel University
Philadelphia, PA 19104 USA
Email: {walt, ashokouf, salvucci}@cs.drexel.edu

***Abstract*—A common problem in many areas of behavioral research is the analysis of the large volume of data recorded during the execution of the tasks being studied. Recent work has proposed the use of an automated method based on canonical sets to identify the most representative patterns in a large data set, and described an initial experiment in identifying canonical web-browsing patterns. However, there is a significant limitation to the method: it requires the similarity matrix to be symmetric, and thus can only be used for problems that can be modeled as unoriented topologies. In this paper we propose a novel enhancement to the method to support oriented topologies by allowing the similarity matrix to be nonsymmetric. We demonstrate the power of this new technique by applying the new method to find canonical lane changes in a driving simulator experiment.**

***Keywords*-pattern classification; approximation methods; human factors**

I. Introduction

In many areas of human-computer interaction (HCI), human factors, and psychological research, data are often collected in the form of time series known as *behavioral protocols* — sequences of actions performed by the user during the execution of some task under study. Behavioral protocols have been used to study a wide variety of actions, ranging from relatively low-level tasks such as mouse clicks, keystrokes, and eye movements (e.g., [1], [2]), to higher-level tasks such as solving algebra problems (e.g., [3]). While protocols are a rich source of data, they have a significant limitation: often, so much data are recorded that it is impractical to analyze everything by hand. Despite the difficulties, researchers have sometimes laboriously studied individual protocols by hand to identify interesting behaviors (e.g., [4], [5]). Limited work has been done on automated protocol analysis, particularly on techniques that match patterns in observed behaviors to the predictions of a user process model (e.g., [6], [7]). However, this approach also has its limitations: such models are often not available, and when they are not, the complexity of the behaviors sometimes makes specification of process models infeasible.

In our previous work we have introduced the notion of *canonical sets* as a novel way of finding patterns in behavioral protocols [8], [9]. Canonical sets are a small subset of patterns that is most representative of the full data set, providing a reasonable "big picture" view of the data with as few elements as possible. Our method does not require an a priori model of the behavior being studied; all that is needed is a similarity measure between pairs of behaviors. However, there is a significant limitation to the method — it requires the similarities to be symmetric, and therefore it can only be used for problems that can be modeled as undirected graphs. In this paper we propose a novel extension to the method which allows for the similarities to be nonsymmetric. Our new method is somewhat similar to directed graph clustering. This is a well-studied problem; techniques include spectral clustering (e.g., [10]) and simulating stochastic flow (e.g., [11]). Please refer to [12] for a survey of other methods. However, in contrast to our canonical set method, traditional graph clustering algorithms require the graph to be sparse, have no notion of feature stability, and do not identify the most representative elements of the clusters.

To test our proposed canonical set algorithm, we considered the problem of finding canonical lane changes in a driving experiment. While driving in general has been the study of much research, lane changing has received relatively little attention. Previous approaches to the problem have generally focused on detecting lane changes in real-time, using tools such as Hidden Markov Models (e.g., [13], [14]) and steerable filters (e.g., [15]). Our algorithm can be an important complement to these techniques, allowing researchers to build their models based on a few representative samples.

The remainder of this paper is structured as follows. In Section II we review our canonical set algorithm and show the changes we propose to allow for oriented topologies. In Section III we describe an experiment applying our algorithm to find canonical lane changes. Finally in Section IV we summarize our findings and discuss possible future directions of research.

II. Problem Formulation

We first review the canonical set technique of [8], which requires the similarities to be symmetric. We then describe how we have enhanced the algorithm to work with oriented topologies.

Suppose we have a set of features $\mathcal{P} = \{p_1, \ldots, p_n\}$ and a set of corresponding stabilities $\mathcal{T} = \{t_1, \ldots, t_n\}$, and we

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.276

wish to find the set of canonical features $\mathcal{P}^* \subset \mathcal{P}$. Let $\mathcal{W}$ be an $n \times n$ feature similarity matrix, where $\mathcal{W}_{ij}$ represents the similarity between features i and j. For all $1 \leq i, j \leq n$ we assume that $\mathcal{W}_{ij} \geq 0$, and we require that $\mathcal{W}_{ii} = 0$. For now we further assume that $\mathcal{W}_{ij} = \mathcal{W}_{ji}$.

We can think of $\mathcal{W}$ as a complete undirected graph $G = (V, E)$ where the vertices represent the features, the weight of each edge $\{u, v\}$ represents the similarity between features u and v, and the weight of each vertex represents the stability of that feature. The goal is to select a subset of vertices $V' \subset V$ that best represents V. The set of edges E may be divided into three disjoint sets. *Intra edges* are those edges which have both of their endpoints within V'. *Cut edges* are edges which have exactly one endpoint in V'. And finally, *extra edges* are edges which have neither endpoint in V'.

The *canonical set* of features $\mathcal{P}^* \subset \mathcal{P}$ is defined such that the features within $\mathcal{P}^*$ are (1) minimally similar to each other, (2) maximally similar to the features outside of $\mathcal{P}^*$ (i.e., $\mathcal{P} \setminus \mathcal{P}^*$), and (3) maximally stable. Stated in terms of a graph, the goal is to minimize the weights of the intra edges while maximizing the weights of the cut edges and the weights of the vertices in $\mathcal{P}^*$.

This problem may be formulated as a quadratic integer program. We define a set of indicator variables $\mathcal{Y} = \{y_1, \ldots, y_n\}$ where $y_i \in \{-1, 1\}$ is 1 if $p_i \in \mathcal{P}^*$ and -1 otherwise. Using these indicator variables, the above objectives may be stated as

$$\text{Minimize } \frac{1}{4} \sum_{ij} \mathcal{W}_{ij}(1 + y_i)(1 + y_j), \tag{1}$$

$$\text{Maximize } \frac{1}{2} \sum_{ij} \mathcal{W}_{ij}(1 - y_i y_j), \tag{2}$$

$$\text{Maximize } \frac{1}{2} \sum_{i=1}^{n} t_i(1 + y_i), \tag{3}$$

respectively. Instead of maximizing the weight of the cut edges, we can minimize the weights of the non-cut edges, given by the sum

$$\frac{1}{2} \sum_{ij} \mathcal{W}_{ij}(1 + y_i y_j). \tag{4}$$

Similarly, we may minimize the stability of the non-canonical features using the sum

$$\frac{1}{2} \sum_{i=1}^{n} t_i(1 - y_i). \tag{5}$$

If we now consider the problem of finding canonical sets in an oriented topology, then G becomes a complete directed graph and $\mathcal{W}$ can no longer be assumed to be symmetric. We note that there are two types of cut edges in G — those that lead into one of the vertices in the canonical set, and those that lead out from one of the vertices in the canonical set. Let us call these *cut-in* and *cut-out* edges, respectively. Suppose we wish to maximize the weights of the cut-in edges while minimizing the weights of the cut-out edges. Such a formulation would tend to lead to a canonical set of sinks, i.e., vertices where it is easy to arrive, but hard to leave.

Without loss of generality, let us assume that the cut-in edges are those where $y_i = -1$ and $y_j = 1$, and the cut-out edges are those where $y_i = 1$ and $y_j = -1$. Then the weight of the cut-in edges is given by

$$\mathrm{Cut}_{\mathrm{in}}(\mathcal{P}^*) = \frac{1}{4} \sum_{ij} \mathcal{W}_{ij}(1 - y_i y_j)(1 + y_j) \tag{6}$$

$$= \frac{1}{4} \sum_{ij} \mathcal{W}_{ij}(1 - y_i + y_j - y_i y_j), \tag{7}$$

since $y_i^2 = y_j^2 = 1$ for all $1 \leq i, j \leq n$. Similarly,

$$\mathrm{Cut}_{\mathrm{out}}(\mathcal{P}^*) = \frac{1}{4} \sum_{ij} \mathcal{W}_{ij}(1 - y_i y_j)(1 + y_i) \tag{8}$$

$$= \frac{1}{4} \sum_{ij} \mathcal{W}_{ij}(1 + y_i - y_j - y_i y_j). \tag{9}$$

We want to maximize $\mathrm{Cut}_{\mathrm{in}}(\mathcal{P}^*)$, which is the same as minimizing $-\mathrm{Cut}_{\mathrm{in}}(\mathcal{P}^*)$. Combining like terms and simplifying, we find that our goal is to minimize

$$\mathrm{Cut}_{\mathrm{in,out}}(\mathcal{P}^*) = \frac{1}{2} \sum_{ij} \mathcal{W}_{ij}(y_i - y_j). \tag{10}$$

We can therefore restate the original problem for oriented topologies as the following quadratic integer programming problem:

$$\text{Minimize} \quad \lambda_1 \left(\frac{1}{4} \sum_{ij} \mathcal{W}_{ij}(1 + y_i)(1 + y_j) \right) \tag{11}$$

$$+\lambda_2 \left(\frac{1}{2} \sum_{ij} \mathcal{W}_{ij}(y_i - y_j) \right) \tag{12}$$

$$+\lambda_3 \left(\frac{1}{2} \sum_{i=1}^{n} t_i(1 - y_i) \right) \tag{13}$$

$$\text{Subject to} \quad \frac{1}{2} \sum_{i=1}^{n} (1 + y_i) - k_{\min} \geq 0, \tag{14}$$

$$k_{\max} - \frac{1}{2} \sum_{i=1}^{n} (1 + y_i) \geq 0, \tag{15}$$

$$y_i \in \{-1, +1\}, \quad \forall\, 1 \leq i \leq n, \tag{16}$$

where λ_1, λ_2 and λ_3 are nonnegative weighting parameters, $\sum_{i=1}^{3} \lambda_i = 1$, and $k_{\min}$ and $k_{\max}$ are minimum and maximum size of the desired subset, respectively. This optimization is known to be intractable [16], and thus we utilize an approximation algorithm: we relax the quadratic

integer program to a quadratic program, and use Matlab's optimization toolbox to find an approximate solution to the problem [17].

III. Experiment

As an initial experiment in applying our canonical set algorithm to oriented topologies, we looked to the domain of driving, specifically the problem of identifying canonical lane changes. Our data come from a previous experiment examining driving behavior [18]. 11 subjects navigated a simulated straight, flat highway with two lanes in each direction. There were a number of automated vehicles on the road which drove at different speeds. All vehicles stayed in the right lane expect to pass. Each test lasted about 30 minutes, during which time data were logged approximately 15 times per second.

We extracted the lane changes for each subject using annotations in the data. We split each lane change into 0.5 second segments, creating a set of histograms for each lane change. The histograms consisted of two fields from the data. First, we used the lateral velocity, computed as change in lane position over change in time. Second, we used the (x, z) position of the vehicle, normalized so that each lane change began at $z = 0$. To compute the similarity between two lane changes, we compared their sets of corresponding histograms using the well-known oriented Hausdorff distance [19]. We note that Hausdorff distances are nonsymmetric and thus could not be used as a similarity measure with the unoriented formulation of the canonical set algorithm.

To compare a pair of points in the histograms, we felt that the difference in lateral velocity was the most important consideration for close points, but became less important the further away they were. Let d_1 be the absolute difference in lateral velocities between the two points, and let d_2 be the Euclidean distance between the two points. We assumed the distribution of d_2 was linear until some point $x_{\min}$, after which it followed a power law. Using the techniques described in [20], we found the average $x_{\min}$ for our data was approximately 94 meters. For $d_2 \leq 94$, we computed the similarity as $s = p/(1 + d_1) + (1 - p)/(1 + d_2)$, where $p = (x_{\min} - d_2)/x_{\min}$. For $d_2 > 94$, we ignored d_1 and set $s = 1/(1 + d_2)$.

As we discussed in [9], the correct λ_i values depend upon the problem being studied as well as a researcher's individual opinions of what constitute "similar" and "different" behaviors. To study the effect of different λ_i values on lane changes, we assumed each lane change was equally stable, and computed the canonical lane changes for each subject over a range of reasonable λ_1 and λ_2 values.

IV. Discussion

Fig. 1 shows the lane changes our algorithm found for a typical subject with $\lambda_1 = 0.55$, and $\lambda_2 = 0.45$. Results were similar for the other subjects with these settings. The

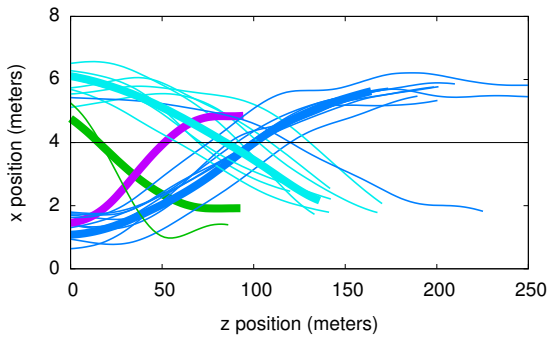

Figure 1. Lane changes for a representative subject. Canonical lane changes are shown in bold, and the other lane changes are drawn in the same color as the canonical lane change they are most similar to.

canonical lane changes are shown in bold, and the other lane changes are drawn in the same color as their most similar canonical lane change. Note that the graph is not shown to scale; the road is only 8 meters wide (two lanes, each 4 meters wide) while many lane changes take several hundred meters to complete. Four canonical lane changes were found for these settings. Decreasing λ_1 produces more canonical lane changes and smaller clusters; conversely, increasing λ_1 results in fewer canonical lane changes and correspondingly larger clusters.

These settings produce good separation between left-to-right and right-to-left lane changes. There is one obvious misclassification: a left-to-right lane change that was found to be most similar to one of the canonical right-to-left lane changes. That lane change is an outlier; it is 50 meters longer than the next longest left-to-right lane change, and it is nearly 100 meters longer than the longest canonical left-to-right lane change. It is not very similar to any of the canonical lane changes, but over the last 50 meters, when lateral velocity does not contribute much to similarity, it is much closer to the longest right-to-left canonical path (recall that the dimensions along the y axis are exaggerated) than to the longest left-to-right path.

As with the rest of the subjects, the canonical lane changes tend to be shorter and smoother than the noncanonical lane changes. This is due to our choice to maximize the cut-in edges while minimizing the cut-out edges. This favors paths which are similar to many other paths, but which do not have many paths similar to them. Such cases occur naturally when using Hausdorff distance since, given two parallel paths of different lengths, the distance from the shorter to the longer path is always less than the distance from the longer to the shorter path. Similarly, when comparing lateral velocities, the smoother path will always be closer to the curvier path than vice versa. Of course, different results can be obtained by changing the distance measure and the assumptions of the algorithm. For example, if we instead tried to maximize the cut-out edges and minimize the cut-in edges, we would

favor the selection of longer, curvier canonical lane changes.

We have presented a novel enhancement to our canonical set algorithm which allows the method to be applied in oriented topologies. As an initial experiment, we have shown that the method can be used to find canonical lane changes. This initial experiment suggests several promising directions for future research. For example, we have not taken into account behavioral stability in this initial experiment. An experiment might be run which considers smooth lane changes to be more stable than those where the driver veers back and forth. Also, many more driving parameters were recorded in the experiment than we used here. It might be interesting, to incorporate eye tracking data into the distance measure. This might require to the development of novel distance measures, as well as tools to assist researchers in exploring the space of λ_i values for their experiment.

ACKNOWLEDGMENT

This work was funded in part by Office of Naval Research (ONR) grant N00014-09-1-0096. The work of second author was funded by ONR grant N00014-04-1-0363. The authors would like to thank Trip Denton for the use of his canonical set code.

REFERENCES

[1] M. D. Byrne, J. R. Anderson, S. Douglass, and M. Matessa, "Eye tracking the visual search of click-down menus," in *Proc. CHI 1999*. ACM Press, 1999, pp. 402–409.

[2] S. K. Card, A. Newell, and T. P. Moran, *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1983.

[3] R. Milson, M. W. Lewis, and J. R. Anderson, "Artificial intelligence and the future of testing," in *The Teacher's Apprentice Project: Building an Algebra Tutor*, R. Freedle, Ed. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1990, pp. 53–71.

[4] K. A. Ericsson and H. A. Simon, *Protocol Analysis: Verbal Reports as Data*, revised ed. Cambridge: MIT Press, 1993.

[5] D. D. Salvucci and J. R. Anderson, "Automated eye-movement protocol analysis," *Human-Computer Interaction*, vol. 16, no. 1, pp. 39–86, 2001.

[6] F. E. Ritter and J. H. Larkin, "Developing process models as summaries of HCI action sequences," *Human-Computer Interaction*, vol. 9, no. 3, pp. 345–383, 1994.

[7] J. B. Smith, D. K. Smith, and E. Kupstas, "Automated protocol analysis," *Human-Computer Interaction*, vol. 8, no. 2, pp. 101–145, 1993.

[8] T. Denton, A. Shokoufandeh, J. Novatnack, and K. Nishino, "Canonical subsets of image features," *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 55–66, 2008.

[9] W. C. Mankowski, P. Bogunovich, A. Shokoufandeh, and D. D. Salvucci, "Finding canonical behaviors in user protocols," in *Proc. CHI 2009*. ACM Press, 2009, pp. 1323–1326.

[10] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, "Detecting communities in large networks," *Physica A: Statistical Mechanics and its Applications*, vol. 352, no. 2-4, pp. 669–676, 2005.

[11] S. van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, University of Utrecht, The Netherlands, 2000.

[12] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[13] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural Computation*, vol. 11, pp. 229–242, 1999.

[14] D. D. Salvucci, H. M. Mandalia, N. Kuge, and T. Yamamura, "Lane-change detection using a computational driver model," *Human Factors*, vol. 49, no. 3, pp. 532–542, 2007.

[15] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, March 2006.

[16] M. R. Garey and D. S. Johnson, *Computers and Intractibility: A Guide to the Theory of NP-Completeness*. San Francisco: W.H. Freeman and Co., 1979.

[17] V. V. Vazirani, *Approximation Algorithms*, 2nd ed. Springer-Verlag, 2003.

[18] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," *Human Factors*, vol. 48, no. 2, pp. 362–380, 2006.

[19] G. A. Edgar, *Measure, Topology, and Fractal Geometry*. Springer, 1995.

[20] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.