Reducing Vehicle Emissions via Machine Learning for Traffic Signal Program Selection (Extended Abstract)

Shumeet Baluja
Google Research, Google, Inc.

1 Motivation and Background

Real-time optimization of traffic flow addresses important practical problems: reducing a driver's wasted time, improving city-wide efficiency, reducing gas emissions, and improving air quality. Many current implementations and research studies that address traffic signal control construct a light controller's program (whether adaptive or static) by segmenting the day into divisions in which distinct traffic patterns are expected, *e.g.* rush hours, weekends, nights, etc.

We consider the problem of automatically adapting a set of traffic lights to changing conditions based upon the distribution of observed traffic-density in surrounding areas. Unlike previous techniques which *a priori* specify the number of unique flow patterns, we assume an over-complete set of traffic patterns. A combination of machine learning approaches are used to create a diverse set of traffic-light programs that can be instantiated when traffic flow patterns are recognized. Using this Match-Based Program Selection, we have observed significant reduction in expected emissions and delays, while being agnostic to the number of underlying distinct patterns in traffic.

2 Data and Simulations

In this study, we utilize a 9-light grid-based world within the SUMO environment (Krajzewicz et al., 2012)[1], see Figure 1. In the simulations, cars enter and exit on any of the 12 edge segments. The speed limit for each road segment was chosen independently and randomly. Five types of car were instantiated with differing acceleration, deceleration, following distance, length, etc.

In our previous experiments with real-user data gathered from Mountain View, California and Chicago Illinois, we determined that modeling traffic density through anonymized location tracks collected from opted-in Google/Android cell phone users (Barth, 2009) provided a robust and reliable source of flow information — an alternative to historic induction loop sensor data. For this study, we simulated the collection of similar data: traffic density measurements of all the road segments to be controlled were recorded during the entire length of the simulations.

[1]SUMO was chosen because it is available to all researchers (open-source), extensible, and allows for massively parallel scenario testing, which was crucial for the $O(10^5)$ simulations required for the experimental results.

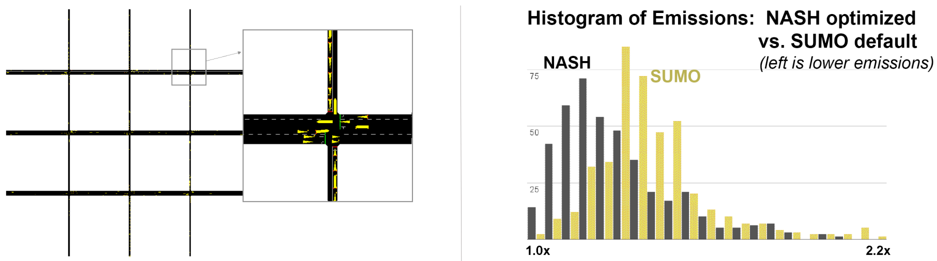


Figure 1: LEFT: Roadway visualized in SUMO. 1-3 lanes in each direction. Traffic-lights at 9 intersections. RIGHT: Histogram of relative emissions for NASH-optimized lights (black) vs. SUMO default (yellow) on 500 scenarios. Note left shift (*less emissions*) with NASH, and large variability across scenarios (220%).

To test the limits of this approach, traffic flows were constructed with significant variability. This rendered any single light setting (with or without induction-loops) suboptimal. To generate the needed variability, 100 independent probability distributions, specifying the probability of external edges being the origin and/or the destination, were created. We simulated a total of 500 unique scenarios (each 45 minutes). For each, one of the 100 distributions was selected and 4,000 cars were instantiated by randomly drawing from the associated origin-destination distribution. Even among sessions drawn from the same underlying distribution, tremendous variability in the backups and delays was created by the randomized sampling order.

Before the measurements were recorded, the static programs of the 9-lights were optimized. We used a straightforward stochastic optimization technique (similar to Genetic Algorithms), *NASH: Next Ascent Stochastic Hillclimbing* (Covell et al., 2015). NASH was allowed to modify each light's phase and offset timings independently through a standard generate-and-test methodology. Recall that with real data, it would be impossible to ascertain the exact number of underlying distributions that generated the scenarios (*if* we had this information, we would ideally select scenarios from each distribution for training). Instead, to reflect real usage, the training scenarios for NASH were randomly selected from the set of 500. Once the light programs were optimized, all 500 scenarios were simulated and the density of flows on each road segment were recorded (as specified previously). Optimization of the programs yielded significant reduction in emissions and travel times over the SUMO default (Figure 1). These optimized lights provide a realistic and strong baseline of performance to which improvements will be measured in the rest of this study.

3 A Clustering Approach to Traffic-Light Program Selection

How do we utilize all the density information collected from the 500 scenarios? Recall that with real data, neither the underlying probability distributions nor even their number are known. Next, we present a clustering method to automatically discover the salient distinctions in the data.

1. From the 500 simulated scenarios, the density of traffic on each directional-road-segment is calculated for 9 non-overlapping 100-second periods (15 min.). This characterizes the traffic conditions in any 15-minute interval by a time-series of traffic densities on each road segment.

With 9 entries for each of the 48 road segments in the map the result is a 432 dimensional vector. Intuitively, when two traffic conditions are similar in their traffic distributions, they should be close in this feature space as well. From the 432 dimensions, we subselect those that exhibit high variance across the 500 scenarios and discard the rest (*feature subset selection*).

2. Using the selected measurements in Step (1), the full 500×500 matrix of correlations between all scenarios is calculated. Intuitively, this reveals, for each pair of scenarios, how similar the density of traffic on the road segments is, when measured over 15 minutes.

3. The scenarios are clustered into C clusters; the similarity between clusters is specified by the correlation matrix. Numerous clustering procedures were explored, a procedure similar to Learning Vector Quantization (LVQ) was finally employed (Kohonen, 1995). Briefly, in LVQ, C points are randomly placed in the high-dimensional feature space. Each scenario then finds its closest point (from C). Then, simultaneously, all of the C points are moved closer to the centroid of their matched scenarios. The procedure is repeated until the points no longer move. Note that with real data, we will not know the correct number of clusters. Overestimating C will not degrade performance since information will be be present, but duplicated. As the effects of underestimating the number are less certain, we study the performance with $C = 10, 50, 100$.

4. For each cluster, c where $c \in C$, we find the scenario, S, where $S \in c$, that is closest to the c's centroid. With S, the light programs for all 9 lights are trained from scratch again to reduce emissions (or reduce delays, etc) just on the particular scenario, S. This yields a specialized light setting, L_c. Exactly as before, the system of lights is trained using *NASH* using the full 45 minutes of scenario, S. More scenarios from c can be used in training L_c, at the expense of extra computation. In the end, C specialized lights settings (each for 9 lights) are created.

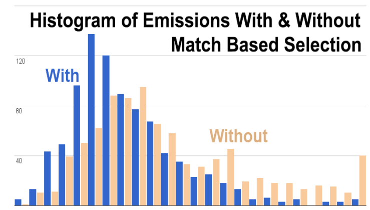
5. Each L_c is tested on *all* 500 scenarios, not just those in it cluster c. Although it is tempting to assume that scenarios in cluster c will perform best with light setting L_c, due to the stochastic nature of training, it is possible that local minima may have been encountered in training and that other light settings have superior performance. Further, some scenarios may have traffic densities that are not well represented by their assigned cluster. After the tests ($500 \times C$), each scenario, S, is assigned the single light setting (L_s where $s \in C$) that best reduced emissions.

4 Experiments and Discussion

Upon completion of the procedure outlined in the previous section, we have a repository of 500 scenarios and the traffic-light programs that work well with each. In deployment, every 15 minutes, 432 traffic-density measurements across all the road segments are recorded. This is accomplished by examining patterns in Android cell phone usage (alternatively, more standard sensors can be used). From these measurements, data extraction proceeds as before: the same high-variance dimensions found in Step (1) are used to find the most similar N scenarios out of the

Table 1: Relative reduction in emissions and travel-time on 1000 scenarios using Match-Based Program-Selection. Consistent 10-12% improvement (**over previously *optimized* timings**). Right Figure: Trials with and w/o Match-Based Selection. Note histogram's shift to the left (lower emissions) with matching.

	Optimized Baseline	C=100	C=50	C=10
Emissions (CO_2, mg)	100%	90.6%	91.9%	91.7%
Travel Time (sec.)	100%	87.7%	89.4%	89.2%
# Scenarios Improved (Tied, Worse)	n/a	900 (18,82)	884 (23,93)	925 (31,44)



the 500 collected (*nearest neighbor search on the 432-dimensional vector using an L_2-Euclidean distance metric*). From these N matches, the associated best light settings are examined. The settings with the most votes (weighted by the scenario's closeness) are deployed to the 9-lights. (In our experiments, $1 \leq N \leq 10$, revealed similar performance).

To test our system, 1,000 *new* scenarios were created. As before, the scenarios were generated by stochastically sampling one of the 100 distributions of origin/destination pairs. Table 1 compares the performance of the baseline NASH-optimized lights with those that used the match-based light switching to instantiate a new program after 15 minutes of observation. The reduction in CO_2 emissions and travel times, compared to the NASH-optimized baseline are given (reductions in CO are similar). Despite the large variability in the distributions of flows, even $C = 10$ clusters modeled enough of the underlying variability to yield improvement (Table 1, last column).

Improved performance in both emissions and travel time is consistently obtained over the previously optimized lights by matching observed traffic densities to similar scenarios and then instantiating the known good light setting for that scenario. Intuitively, each scenario serves as a fingerprint to which observed densities are matched. Because we matched new traffic flows to individual scenarios and not just to clusters, we effectively expanded the granularity of the basis set and were able to find closer matches. The most similar related work comes from Case-Based Reasoning, see Kofod-Petersen et al. (2014).

This paper incorporated machine learning into light-program selection. There are two immediate next steps. The first is scaling and testing this system on city traffic data. The large number of underlying distributions that were successfully modeled is promising for the variations expected in real data. Second, employing this system with lights triggered by external (*e.g.* induction-loop/camera) sensors is conceptually possible, and is an interesting direction for exploration.

References

D. Barth. The bright side of sitting in traffic: Crowdsourcing road congestion data. http://googleblog.blogspot.com/-2009/08/bright-side-of-sitting-in-traffic.html, 2009.

M. Covell, S. Baluja, and R. Sukthankar. Micro-auction based traffic light control. In *ITSC-2015*, 2015.

A. Kofod-Petersen, O. J. Andersen, and A. Aamodt. Case-based reasoning for improving traffic flow in urban intersections. In *Case-Based Reasoning Res. and Dev.*, pages 215–229. Springer, 2014.

T. Kohonen. Learning vector quantization. In *Handbook of Brain Theory and Neural Networks*, 1995.

D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - simulation of urban mobility. *Int. J. On Advances in Systems and Measurements*, 5, 2012.