Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

Kernel regression for travel time estimation via convex optimization

Sébastien Blandin , Laurent El Ghaoui and Alexandre Bayen

Abstract—**We develop an algorithm aimed at estimating travel time on segments of a road network using a convex optimization framework. Sampled travel time from probe vehicles are assumed to be known and serve as a training set for a machine learning algorithm to provide an optimal estimate of the travel time for all vehicles. A kernel method is introduced to allow for a non-linear relation between the known entry times and the travel times that we want to estimate. To improve the quality of the estimate we minimize the estimation error over a convex combination of known kernels. This problem is shown to be a semi-definite program. A rank-one decomposition is used to convert it to a linear program which can be solved efficiently.**

I. Introduction

Travel time estimation on transportation networks is a valuable traffic metric. It is readily understood by practitioners, and can be used as a performance measure [3] for traffic monitoring applications. Note however that the travel time estimation problem is easier to address on highways than on arterial roads. This can be intuitively interpreted by the fact that properties of highways can be considered to be 'more spatially invariant' than the ones of arterial roads. Indeed the latter present complex features such as intersections and signalization forcing to stop resulting in spatially discontinuous properties. In this article, we propose a new method to estimate travel time on road segments without any elaborated model assumption. This method is shown to belong to a specific class of convex optimization problems and provides a non-linear estimate of the travel time. The kernel regression method introduced allows for estimation improvement through the online extension of the set of kernels used. In particular, we assess the performance of this technique through a rank one kernel decomposition.

Highway traffic modeling is a mature field. Macroscopic models date back to [10], [16], [22] and usually fall under the theory of scalar conservation laws [9]. Microscopic models take into account vehicle driving behaviors and can be derived from the *car-following* model [17]. Flow models and driving behavior models on arterials are still the focus of significant ongoing research.

S. Blandin is a Ph.D. student, Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720-1710 USA (e-mail: blandin@berkeley.edu). Corresponding author.

L. El Ghaoui is a Professor, Department of Electrical Engineering and Computer Science, University of California Berkeley, CA 94720-1710 USA (e-mail: elghaoui@berkeley.edu).

A. Bayen is an Assistant Professor, Systems Engineering, Department of Civil and Environmental Engineering, University of California Berkeley, CA 94720-1710 USA (e-mail: bayen@berkeley.edu).

Sensors such as loop detectors are widely available on highways. They provide accurate measurements of density or speed [13], but are extremely sparse on arterials. Thus hardly any information of arterial traffic is available in real-time. It is only recently that the growth of mobile sensors has been shown to offer reliable information for traffic monitoring [12], [29]. One can hope to use this source of information to provide accurate estimate of travel time on arterials.

Travel time estimation on highways has been investigated with different tools. Efforts have been made from the modeling side, assuming local knowledge of density or speed, and producing an estimate given by a deterministic or stochastic model [5], [14], [20]. This problem has also been addressed using data analysis and machine learning techniques with various types of learning methods [18], [19], [21], [28], [30].

Arterial travel time estimation is more complex because the continuum approximation of the road might not apply at intersections, where the dynamics is not easily modeled [1]. Information about the state of traffic on arterials is also limited because of the sparsity of sensors. Some attempts have been made to estimate travel time on arterials, but in practice it is not always possible to know the traffic lights cycles or to obtain a dedicated fleet of probe vehicles, often needed for estimation [24], [25]. However the ubiquity of GPS now enables one to realistically assume the knowledge of sampled travel times, an assumption for example verified in sections of Northern California with the Mobile Millennium system [11].

We propose to focus on arterial travel time estimation using machine learning techniques and convex optimization. We use kernel methods [23] to provide a non-linear estimate of travel time on an arterial road segment. We assume the knowledge of the travel times of a subset of vehicles and estimate the travel time of all vehicles. We use convex optimization [2] to improve the performance of the non-linear estimate through kernel regression. The regression is done on a set of kernels chosen according to their usually good performances, or physical criteria. The kernel framework [6] enables the addition of features to the set of covariates in the regression problem. The kernel regression gives the possibility to select the most relevant features via optimization.

This article is organized as follows. In section II, we describe the optimization problem, introducing the regularization parameter and the kernel in a learning setting. In section III, we pose the kernel regression problem and show that it can be written as a convex optimization problem, transform it into a linear program, which can be solved

978-1-4244-3872-3/09/$25.00 ©2009 IEEE

efficiently and we describe the general learning algorithm used. In section IV, we present the simulation dataset used for validating the method and the results obtained. In particular, we discuss the theoretical results stating that kernel regression enables to obtain a better estimate on the validation set. Finally, based on these results, we enumerate in section V ongoing extensions to this work.

II. Problem Statement

A. Travel time estimation

We investigate travel time estimation for a given road segment. Assuming a set of entry times and travel times on the section, we apply machine learning techniques to use the knowledge of a subset of the pairs entry time, travel time, in order to produce an estimate of travel time for every entry time. The dataset used for validation is described later in section IV-A. We assume the knowledge of a dataset of size N which reads $\mathcal{S} = \{(x_i, y_i) \in \mathbb{R}^+ \times \mathbb{R}^+ | i = 1 \cdots N\}$ where for each value of the index $i = 1 \cdots N$, x_i is an entry time on the road section and y_i is the realized travel time (known as the *a-posteriori travel time* in the transportation community) for entry time x_i. We would like to learn a function $h : \mathbb{R}^+ \to \mathbb{R}^+$ which given $\mathcal{S}$, would provide an estimate of the travel time y for any $x \in \mathbb{R}^+$. This is a typical regression problem as described in [6]. The well-known unconstrained least-squares method can be formulated as an optimization problem:

$$\min_\theta \quad \|y - x^T \theta\|_2^2 \tag{1}$$

where $y \in \mathbb{R}^{N \times 1}$ is the vector of realized travel time or output, $x^T \in \mathbb{R}^{N \times 1}$ is the vector of entry time or input. One must note that here x is a row vector and y is a column vector so θ is a scalar. The well-known solution of this problem can be computed as:

$$\theta_{\text{opt}} = (x\, x^T)^\dagger\, x\, y \tag{2}$$

where the notation $(x\, x^T)^\dagger$ denotes the pseudo-inverse of $(x\, x^T)$ and the optimal estimate is given by $\hat{y} = (x\, x^T)^\dagger\, x^T\, x\, y$. This estimate does not have bias, i.e. the mean of the output y equals the mean of the estimate $\hat{y}$.

B. Regularization

The regression problem defined in (1) is often ill-posed in the sense that the solution does not depend continuously on the data (the case of multiple solutions falls into that denomination). Formulation (1) could also lead to over-fitting in the case of non-linear regression since there is no penalization for high values of the solution θ_{opt}. In order to prevent these two possible flaws, it is a common practice to add to the objective function a quadratic term called *Tikhonov regularization* [26] which has the form $\rho^2 |\theta|^2$ in the scalar case. Then the optimal estimate becomes:

$$\hat{y} = (x\, x^T + \rho^2\, \mathrm{I})^{-1}\, x^T\, x\, y. \tag{3}$$

For ρ large enough, the problem is well-posed and over-fitting with respect to θ is prevented [8].

C. Kernel methods

The regression method described in section II-A in the linear case can be extended to the non-linear case through the use of a kernel. One can consider a mapping function $\phi(\cdot)$ and consider the linear regression problem between the mapped covariates $\phi(x_i)$ and the outputs y_i. This is the main principle of kernel methods, which consist in using a feature space, in which the dataset is represented, and to consider linear relations between objects in this feature space, and between these features and the outputs. Given a positive semi-definite matrix $K = (k_{ij})$, we define the kernel function by $K_{\text{f}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $K_{\text{f}}(x_i, x_j) = k_{ij}$. This implicitly defines a feature mapping $\phi(\cdot)$ between the input set $\mathcal{X}$ and a Hilbert space $\mathcal{H}$ by $\phi(\cdot) : \mathcal{X} \to \mathcal{H}$ such that $\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} = K_{\text{f}}(x_i, x_j)$. In the following we note X_{map} a matrix representation of the mapping $\phi(\cdot)$ (thus the i-th column of X_{map} is $\phi(x_i)$). When $\phi(\cdot)$ has scalar values, X_{map} is a row vector.

Remark 1: One does not have to define a mapping function $\phi(\cdot)$ to define a kernel matrix, but can simply consider a positive semi-definite matrix and use it as a kernel. It is also possible to define a kernel matrix from a mapping $\phi(\cdot)$ and one of its matrix representation X_{map} as $K = X_{\text{map}}^T\, X_{\text{map}}$.

The inner product in $\mathcal{H}$ naturally appears to be given by the Gram matrix K, called the kernel. Kernel techniques [6], [23] have several benefits:

- They enable to work with any types of features of the initial data-set, which has a priori no particular structure, in a Hilbert space.
- They guarantee a reasonable computational cost for the algorithm by allowing a complexity related to the number of points represented and not the number of features used (this is known as the kernel trick and is described in Remark 4).

Thus, kernel methods provide several extensions to usual regression methods, and can be easily written in a machine learning framework.

D. Learning setting

We assume the knowledge of a training set $\mathcal{S}_{\text{tr}} = \{(x_i, y_i) | i = 1 \cdots n_{\text{tr}}\}$ and we look for the best estimate of the elements of the test set, $\mathcal{S}_{\text{t}} = \{x_i | i = n_{\text{tr}} + 1 \cdots n_{\text{tr}} + n_{\text{t}}\}$. In order to match the structure of this problem, we define the kernel matrix in block form as:

$$K = \begin{pmatrix} K_{\text{tr}} & K_{\text{trt}} \\ K_{\text{trt}}^T & K_{\text{t}} \end{pmatrix} \tag{4}$$

where $k_{ij} = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$ for $i, j = 1 \cdots n_{\text{tr}}, n_{\text{tr}}+1 ... n_{\text{tr}}+ n_{\text{t}}$. The Gram matrix K_{tr} is the result of an optimization problem over the training set, and we learn the cross-term K_{trt}, which expresses the inner-product in the feature space

between the elements of the test-set and the elements of the training set. K_{t} is the inner product between the elements of the test set.

III. Analysis

A. Convex formulation

Expressing the linear least-squares (1) for the mapped input X_{map} with the regularization term described in section II-B yields:

$$p^* = \min_\theta \quad \|y - X_{\mathrm{map}}^T \theta\|_2^2 + \rho^2 \|\theta\|_2^2 \tag{5}$$

where we note p^* the optimal value of this problem. Using the change of variable $z = X_{\mathrm{map}}^T \theta - y$ yields the equivalent formulation:

$$p^* = \quad \min_{\theta,z} \quad \|z\|_2^2 + \rho^2 \|\theta\|_2^2 \tag{6}$$

$$\text{subject to} \quad z = y - X_{\mathrm{map}}^T \theta \tag{7}$$

The lagrangian dual of this problem reads:

$$d^* = \max_\alpha \quad -2\,\alpha^T y - \alpha^T \left(I + \frac{X_{\mathrm{map}}^T X_{\mathrm{map}}}{\rho^2} \right) \alpha. \tag{8}$$

In this equation, we see the expression of the kernel matrix:

$$K = X_{\mathrm{map}}^T X_{\mathrm{map}}. \tag{9}$$

If we denote $K_\rho = I + \frac{X_{\mathrm{map}}^T X_{\mathrm{map}}}{\rho^2}$ the regularized kernel, the dual optimal point and dual optimal value of problem (8) can be expressed as:

$$\alpha^* = K_\rho^{-1} y \quad \text{and} \quad d^* = y^T K_\rho^{-1} y. \tag{10}$$

Remark 2: Since the primal (6)-(7) and dual (8) are convex and strictly feasible, strong duality holds and primal optimal value p^* and dual optimal value d^* are equal. We note that expression (8) shows that the dual optimal value is a maximum over a set of linear functions of K_ρ, so the optimal value is a convex function of the regularized kernel matrix K_ρ. Since the choice of the kernel is crucial for the optimal value it is interesting to minimize the optimal value d^* with respect to the kernel.

Remark 3: Optimizing the kernel matrix physically means looking for the best mapping function $\phi(\cdot)$ such that there is a linear relation between the features of the inputs $\phi(x_i)$ and the outputs y_i. If one takes $\phi(\cdot)$ as the identity mapping, then the optimal value of (8) becomes:

$$y^T K_\rho^{-1} y = y^T \left(I + \frac{x^T x}{\rho^2} \right)^{-1} y \tag{11}$$

which may not be optimal for non-linear systems.

Remark 4: One must note that the kernel matrix (9) is a square matrix which has the dimension of $x^T x$, and thus its size does not depend on the number of features represented in X_{map} but only on the number of covariates x_i. The dimension of the image space of $\phi(\cdot)$ which is the dimension of the feature space, does not appear in the kernel matrix. This is the kernel trick mentioned in section II-C.

B. Cross-validation

The optimal value of (5) as expressed in (10) depends on the kernel matrix (9) and on the regularization parameter ρ. The parameter ρ is tuned through a re-sampling procedure [7], the *k-fold cross-validation* (here k does not denote the kernel matrix but the number of folds used in the cross-validation method). This technique consists in dividing the dataset into k parts of approximately equal size, and using a subset for training while the remainder is used for testing [27]. For instance if the different parts are $\{P_i | i = 1 \cdots k\}$ then given $n \in \{1 \cdots k\}$ one would use P_n as a training set and $\bigcup_{i=1\cdots k,\ \ i\neq n} P_i$ as a test set. This is useful to make extensive use of the dataset while avoiding bias on the training set. Here we use this method on the training set to pick the optimal value of the regularization parameter ρ and on the whole set to have a meaningful estimation error.

C. Kernel regression

As stated in Remark 2, the optimal value of the regularized regression problem (5) is a convex function of the regularized kernel matrix K_ρ and can be optimized over the kernel. The kernel optimization problem, which consists in minimizing the value d^* defined in (10) with respect to the regularized kernel K_ρ reads:

$$\min_{K_\rho} \quad y^T K_\rho^{-1} y \tag{12}$$

$$\text{subject to} \quad K_\rho \geq 0 \tag{13}$$

where the constraint on the kernel matrix enforces that the regularized kernel K_ρ must be a Gram matrix. This problem is convex according to Remark 2. In order to prevent overfitting with respect to K_ρ, we follow [15] and constrain K_ρ to be a convex combination of given kernels, i.e. we define a set of kernels $\{K_1 \cdots K_k\}$ and consider the problem:

$$\min_\lambda \quad y^T K_\rho^{-1} y \tag{14}$$

$$\text{subject to} \quad \lambda_i \geq 0 \quad \sum_{i=1}^k \lambda_i = 1 \tag{15}$$

$$K_\rho = \sum_{i=1}^k \lambda_i K_i. \tag{16}$$

In a learning setting, the optimization problem (14) is defined only on the training set but the expression of the kernel matrix as a linear combination of known kernels must be satisfied on the whole set. Using the notation introduced in section II-D we write $K_\rho = \begin{pmatrix} K_{\mathrm{tr}} & K_{\mathrm{trt}} \\ K_{\mathrm{trt}}^T & K_{\mathrm{t}} \end{pmatrix}$ and under this form the problem reads:

$$\min_\lambda \quad y^T K_{\mathrm{tr}}^{-1} y \tag{17}$$

$$\text{subject to} \quad \lambda_i \geq 0 \quad \sum_{i=1}^k \lambda_i = 1 \tag{18}$$

$$K_\rho = \sum_{i=1}^k \lambda_i K_i \tag{19}$$

which can be written in a semi-definite program form using an epigraph property and the Schur complement:

$$\min_{\lambda,t} \quad t \tag{20}$$

$$\text{subject to} \quad \lambda_i \geq 0 \quad \sum_{i=1}^{k} \lambda_i = 1 \tag{21}$$

$$K_\rho = \sum_{i=1}^{k} \lambda_i K_i \;\text{ and }\; \begin{pmatrix} t & y^T \\ y & I + \frac{K_{\text{tr}}}{\rho^2} \end{pmatrix} \geq 0. \tag{22}$$

The solution of this optimization problem is the parameter λ^* giving the optimal convex combination of the set of kernels $\{K_1 \cdots K_k\}$ which minimizes d^* from (10).

D. Rank-one kernel optimization

The kernel optimization problem in the form of (20)-(21)-(22) is not tractable and cannot be efficiently solved by standard optimization software. In this section we use the rank-one decomposition of kernels to find an equivalent formulation in a linear program form. This is done through the introduction of several intermediate problems. We assume that we can write the regularized kernel as a convex combination of dyads: $K_\rho = \sum_{i=1}^{p} \nu_i \, l_i \, l_i^T$ where l_i are row vectors and ν_i are positive scalars such that $\sum_{i=1}^{p} \nu_i = 1$. Since by definition $K_\rho = I + \frac{K}{\rho^2}$, the decomposition of K_ρ into a sum of dyads is possible whenever the kernel K itself can be written as a sum of dyads. In practice the kernel is a positive semi-definite matrix so it can be diagonalized in an orthonormal basis and this property is satisfied. Thus we can write an equivalent formulation of problem (14)-(15)-(16) as:

$$\Psi = \quad \min_{\nu} \quad y^T \, K_\rho^{-1} \, y \tag{23}$$

$$\text{subject to} \quad \nu_i \geq 0 \quad \sum_{i=1}^{p} \nu_i = 1 \tag{24}$$

$$K_\rho = \sum_{i=1}^{p} \nu_i \, l_i \, l_i^T \tag{25}$$

where the vectors l_i are the eigenvectors of the matrices K_j from equation (16). Introducing the change of variable $\kappa = K_\rho^{-1}(\nu)$ and doing some computations enables one to rewrite problem (23)-(24)-(25) as:

$$\Psi = \max_{\kappa} \quad \left(2 \, y^T \, \kappa - \max_{1 \leq i \leq p} (l_i^T \, \kappa)^2 \right) \tag{26}$$

and the optimal κ is related to the optimal ν by the relation

$$\kappa^* = K_\rho^{-1}(\nu^*). \tag{27}$$

One can note that solving problem (26) for the vector variable κ is the same as solving the problem:

$$\Psi = \min_{\gamma,\beta} \quad \left(2 \, y^T \, \gamma \, \beta - \max_{1 \leq i \leq p} (l_i^T \, \gamma \, \beta)^2 \right) \tag{28}$$

for the variables γ and β. This is simply obtained by writing $\kappa = \gamma \, \beta$ in problem (26), with γ scalar and β vector. The optimal point (γ^*, β^*) of problem (28) satisfies:

$$\Psi^{1/2} \, \beta^* = \gamma^* \, \beta^* = \kappa^*. \tag{29}$$

If we minimize over γ in (28) we obtain the following optimization problem:

$$\Psi^{1/2} = \quad \max_{\beta} \quad y^T \, \beta \tag{30}$$

$$\text{subject to} \quad |l_i^T \, \beta| \leq 1 \quad i = 1...p. \tag{31}$$

The lagrangian of this problem can be written as:

$$\mathcal{L}(\beta, u) = y^T \, \beta + \sum_{i=1}^{p} \left(|u_i| - u_i \, \left(l_i^T \, \beta \right) \right) \tag{32}$$

and taking the lagrangian dual of problem (30)-(31) yields:

$$\Psi^{1/2} = \quad \min_{u} \quad \|u\|_1 \tag{33}$$

$$\text{subject to} \quad y = \sum_{i=1}^{p} u_i \, l_i \tag{34}$$

using the strict feasibility of the primal and convexity of the primal and the dual. Problem (34) is a linear program. The optimal ν^* can be retrieved from the optimal u^* from the relation:

$$\nu_i^* = \frac{|u_i|^*}{\Psi^{1/2}}. \tag{35}$$

Indeed one can check that with this value of the vector ν equations (27)-(29) yields $\Psi^{1/2} \, K_\rho(\nu^*) \, \beta^* = y$ and on the other hand we can write $\Psi^{1/2} \, K_\rho(\nu^*) \, \beta^* = \sum_{i=1}^{p} |u_i^*| \, (l_i^T \, \beta) \, l_i$ which is equal to $\sum_{i=1}^{p} u_i \, l_i$ using the optimality condition in the lagrangian (32). This proves that if u^* is optimal for (33)-(34) then ν^* given by (35) is optimal for (23)-(24)-(25) and vice-versa.

E. Choice of kernels

Several types of kernels are commonly used in machine learning [23]. Here, we propose to combine several classical kernels with a kernel motivated by the known physical properties of the phenomenon we want to estimate.

1) Classical kernels: We consider a Gaussian kernel K^σ defined by $k_{ij}^\sigma = \exp(-\frac{|x_i - x_j|^2}{\sigma^2})$. We also use a linear kernel K^{lin} defined by $k_{ij}^{\text{lin}} = x_i \, x_j$. Since we use a rank-one decomposition of each kernel, the regression problem with the linear kernel K^{lin} is expected to produce a slightly better estimate than the regular linear least-squares, because in the kernel method the weight of the eigenvectors can be different.

2) Physics of the phenomenon: Since we are interested in estimating the travel time across a traffic light intersection, we consider the physical properties of this phenomenon as described in [1]. According to the authors, a reasonable model is the following: the travel time across a traffic light intersection increases suddenly at the beginning of the red light and decreases linearly from there until the next beginning of the red light. This motivates us to use a piecewise

linear function $\phi(\cdot)$ as a mapping. In order to do so we assume the traffic cycle length to be constant of value c. The slope of the linear function is left free and is chosen to be an optimization parameter. These considerations lead us to define the mapping function for the third kernel as $\phi^{\text{phy}}(x) = x \ mod \ c$ where c is the traffic cycle length. It is motivated by the fact that according to the model the phenomenon is periodic of period c, and on a period the relation between the entry time and the travel time is linear. We assume that $c = 60$ seconds.

IV. Simulation results

A. Dataset description

We use a dataset generated by a traffic micro simulator, Paramics [4]. It is based on the car-following theory and driving behavior modeling and has been the subject of extensive research funded by Caltrans. It is assumed to accurately reproduce the macroscopic properties of traffic as well as inconsistent driving patterns observed in real life. Thus it provides a challenging dataset to estimate the performance of our algorithm. The dataset consists of 1055 pairs (x_i, y_i) where x_i is an entry time and y_i is the travel time of a vehicle entering the road section at time x_i. This dataset has been generated for a road segment in Berkeley, California. It consists of an arterial link of length 1207 feet and the simulation has been run for half an hour between $3:30$ PM and $4:00$ PM on a week day.

B. Analysis method

Given an entry time x_i, we would like to provide a travel time estimate $\hat{y}_i$. We are interested in the quadratic error between this estimate $\hat{y}_i$ and the effective travel time y_i. In order to evaluate the performance of the techniques described in section III, we follow the method described below. The error metric used is a L_2 relative norm.

- We consider a training set whose size is one half of the size the whole dataset. This can be considered to model the fact that we know one half of the travel times of vehicles flowing on the road section, and we want to estimate the travel time of the other vehicles.
- In order to define the optimal regularization parameter for the training set, we define a 5-fold on this set. We use cross-validation as defined in section III-B on this 5-fold. Namely we use one of the five subsets as a training set, and the remainder serves as the test-set. We solve the optimization problem described in section III on each of the five training sets, and for several values of ρ, and we pick the one which minimizes the error metric.
- Having defined a regularization parameter at the previous step, we compute the optimal weight vector from the training set and evaluate the error on the test-set.
- We iterate this method for different training sets being in size one-half of the whole dataset, and we average the errors obtained. The results are given in Table I for different convex combinations of kernels.

These computations are executed with Matlab, and the optimization problems are solved by CVX, which is a disciplined convex programing tool, using the program SDPT3.

C. Results and discussion

Kernel	Error on training set	Error on test set
K^{lin}	1.09	1.10
K^{phy}	1.07	1.09
K^{σ}	0.79	0.76
$K^{\text{lin}} + K^{\text{phy}}$	1.07	1.10
$K^{\text{lin}} + K^{\sigma}$	0.79	0.76
$K^{\text{phy}} + K^{\sigma}$	0.77	0.74

Table I
VALUES OF THE L_2 RELATIVE ERROR ON THE TRAINING SET AND ON THE TEST SET FOR DIFFERENT COMBINATIONS OF KERNELS, FOR A TRAINING SET OF SIZE 50 % OF THE SIZE OF THE WHOLE DATASET. WE NOTE K^{σ} THE GAUSSIAN KERNEL AND USE $\sigma = 100$, K^{LIN} THE LINEAR KERNEL, AND K^{PHY} THE PHYSICAL KERNEL.

The results shown in Table I yield several observations. First, the high value of the relative errors must be compared to usual techniques, such as the conventional linear least-squares. For this dataset, the linear least-squares optimal estimate gives a L^2 relative error of 1 on the training set, because it produces an estimate without bias. The error on the test set is as close to 1 as the distribution of the training set is close to the distribution of the test set. When using only one kernel, the estimate performs at least almost as well as the linear least-squares estimate. The performance of the combination of kernels is at least as good at the performance of the best kernels, and is better when the two kernels have different features. In the case of a Gaussian kernel and a physical kernel, the result is improved because the physical kernel uses a feature (the operator *mod*) which does not exist in the Gaussian kernel. In the case of the linear kernel and the Gaussian kernel, there is no added value compared to the Gaussian kernel alone. The benefit of a combination of kernels is that if one does not know a-priori which kernel performs better, the optimization algorithm picks an optimal combination and yields a mixed non-linear estimate which, as illustrated in figure 1, is able to locally capture trends of the phenomenon. This property of the estimate does not appear in the comparison proposed in Table I but is really useful for traffic applications. Here the linear least-squares estimate has a constant value at the mean value of the dataset, whereas the estimate given by a combination of the kernels oscillates. This is the subject on ongoing research which uses the same formalism with other norms such as the H^1 norm instead of a L^2 norm in the objective function of (5).

V. Conclusions and future work

The result presented in previous sections show that even if the accuracy obtained by the kernel regression technique is not spectacular, an added value is the fact that the estimate is able to follow the trend of the travel time.

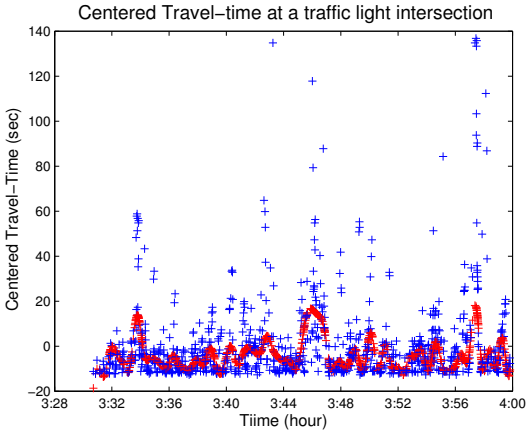

Figure 1. Observed travel times (blue sparse points) and estimated travel times (red smooth curve) for a linear combination of a Gaussian kernel (with $\sigma = 100$) and a linear kernel.

The kernel regression technique enables to add kernels to the set used in order to provide a richer signal providing better accuracy. Thus extensions to this work include the use of different kernels offering other features to improve the results obtained. In particular, it would be satisfying to reach sufficiently good estimation accuracy with kernels only based on the physical properties of the road and some varying parameters (weather, time of day). Applying results from the support vector machines theory allowing to bound the error in classifiers would significantly improve the quality of our travel time estimate. The estimated travel time on a road segment may be as important for practitioners as its range of variations. This is related to the discussion in section IV-C on the ongoing research focusing on the use of other norms in the regression problem (5), and how to find a tractable and efficient way to solve the problem in these cases.

REFERENCES

[1] X. BAN, R. HERRING, P. HAO, and A. BAYEN. Delay pattern estimation for signalized intersections using sampled travel times. In *Transportation Research Board Annual meeting*, Washington, D.C., January 10-14 2008.

[2] S. BOYD and L. VANDENBERGHE. *Convex optimization*. Cambridge university press, 2004.

[3] T. CHOE, A. SKABARDONIS, and P. Varaiya. Freeway performance measurement system: an operational analysis tool. *Transportation Research Record*, 1811(-1):67–75, 2002.

[4] L. CHU, H. LIU, and W. RECKER. Using microscopic simulation to evaluate potential intelligent transportation system strategies under nonrecurrent congestion. *Transportation Research Record*, 1886(-1):76–84, 2004.

[5] B. COIFMAN. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. *Transportation Research Part A*, 36(4):351–364, 2002.

[6] N. CRISTIANINI and J. SHAWE-TAYLOR. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[7] B. EFRON and G. GONG. A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician*, pages 36–48, 1983.

[8] H. ENGL, K. KUNISCH, and A. NEUBAUER. Convergence rates for tikhonov regularization of nonlinear ill-posed problems. *Inverse Problems*, 5(4):523–540, 1989.

[9] M. GARAVELLO and B. PICCOLI. *Traffic flow on networks*. American Institute of Mathematical Sciences, Springfield, USA, 2006.

[10] B. GREENSHIELDS. A study of traffic capacity. *Proceedings of the Highway Research Board*, 14(1):448–477, 1935.

[11] R. HERRING, A. HOFLEITNER, S. AMIN, T. ABOU NASR, A. ABDEL KHALEK, P. ABBEEL, and A. BAYEN. Using mobile phones to forecast arterial traffic through statistical learning. *Submitted to Transportation Research Board*, 2009.

[12] B. HOH, M. GRUTESER, R. HERRING, J. BAN, D. WORK, J.-C. HERRERA, A. BAYEN, and Q. JACOBSON. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *6th International Conference on Mobile Systems, Applications, and Services*, pages 15–28, Breckenridge, CO, June 17-18 2008.

[13] Z. JIA, C. CHEN, B. COIFMAN, and P. VARAIYA. The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 536–541, 2001.

[14] P. KACHROO, K. OZBAY, and A. HOBEIKA. Real-time travel time estimation using macroscopic traffic flowmodels. *2001 Proceedings of IEEE Intelligent Transportation Systems*, pages 132–137, 2001.

[15] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[16] M. LIGHTHILL and G. WHITHAM. On kinematic waves II a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London*, 229(1178):317–345, 1956.

[17] G. NEWELL. A simplified car-following theory: a lower order model. *Transportation Research Part B*, 36(3):195–205, 2002.

[18] D. NIKOVSKI, N. NISHIUMA, Y. GOTO, and H. KUMAZAWA. Univariate short-term prediction of road travel times. *2005 IEEE Intelligent Transportation Systems, 2005. Proceedings*, pages 1074–1079, 2005.

[19] T. PARK and S. LEE. A bayesian approach for estimating link travel time on urban arterial road network. *Lecture notes in computer science*, pages 1017–1025, 2004.

[20] K. PETTY, P. BICKEL, M. OSTLAND, J. RICE, F. SCHOENBERG, J. JIANG, and Y. RITOV. Accurate estimation of travel times from single-loop detectors. *Transportation Research Part A*, 32(1):1–17, 1998.

[21] J. RICE and E. VAN ZWET. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004.

[22] P. RICHARDS. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

[23] B. SCHOLKOPF and A. SMOLA. *Learning with kernels*. MIT press, 2002.

[24] A. SKABARDONIS and N. GEROLIMINIS. Real-time estimation of travel times on signalized arterials. In *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, 2005.

[25] K. SRINIVASAN and P. JOVANIS. Determination of number of probe vehicles required for reliable travel time measurement in urban network. *Transportation Research Record*, 1537(-1):15–22, 1996.

[26] A. TIKHONOV. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl*, volume 4, pages 1035–1038, 1963.

[27] P. TURNEY. A theory of cross-validation error. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:361–361, 1994.

[28] J. VAN LINT and H. Van Zuylen. Monitoring and predicting freeway travel time reliability: Using width and skew of day-to-day travel time distribution. *Transportation Research Record*, 1917:54–62, 2005.

[29] D. WORK, O-P. TOSSAVAINEN, S. BLANDIN, A. BAYEN, T. IWUCHUKWU, and K. TRACTON. An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In *47th IEEE Conference on Decision and Controls, 2008 Cancun, Mexico*, pages 5062–5068, 2008.

[30] J. YEON, L. ELEFTERIADOU, and S. LAWPHONGPANICH. Travel time estimation on a freeway using discrete time markov chains. *Transportation Research Part B*, 42(4):325–338, 2008.