Personalized Learning using Multiple Kernel Models

Anthony Kuh* Shuai Huang † Cynthia Chen†
* University of Hawaii, Honolulu, HI
E-mail: kuh@hawaii.eu
† University of Washington, Seattle, WA
E-mail: shuaih@uw.edu & qzchen@uw.edu

***Abstract*—This paper considers a personalized learning system with a large number of users each trying to learn a desired task. The user learning tasks have similarities as each of the users have common attributes. However, the learning tasks are each slightly different as the tasks are personalized to each learning system. There is also the concern that each learning system may not receive enough data to learn its desired task. To account for this a few canonical learners receive all the data from each personalized learning system. Each personalized learning system then takes a weighted sum of the canonical learners to realize their desired task. Here we will consider learning via kernel methods.**

***Index Terms*—personalized learning, kernel methods, canonical models**

I. Introduction

Gaining a better understanding of human behavior can help steer modern societies toward being more sustainable, equitable and healthier. In public health, even with vaccinations with high efficacies, social distancing and wearing face masks remain effective measures against the spread of coronavirus. In transportation, if Americans can use more alternative modes of transportation such as transit and non-motorized modes, ride share, or simply space out times of day for driving, both congestion and emissions would be greatly reduced. In power grids and water systems, if people can conserve energy and water usage, there will also be substantial savings. All these require behavioral changes. But behavioral changes are hard.

The prevalence of mobile devices presents new ways for making behavioral changes on a population scale. This has resulted in vast amount of data, which are distributed across devices. The vast data provides us the opportunity to learn behaviors and the underlying preferences not only in a personalized way but also collectively, seeking commonalities across a population. This paper develops a novel approach of achieving personalized and collective learning at the same time. More specifically, the study proposes the concept of a canonical model structure wherein the idea is to divide a population into L segments and the behaviors and preferences of each population segment is represented by lth canonical model structure, or lth canonical learner. This is collective learning, seeking to identify similarities across individuals in a population. Another important reason for the collective learning is that there may not be enough data for each individual, making the training difficult if learning of a personalized system is only limited to the data of one individual. In contrast the canonical learner receives the aggregated training data from each personalized system and can learn tasks more accurately. Personal preference of an individual is then the weighted sum of those L canonical learners and the weights for each individual are unique. This constitutes personalized learning. This personalized and collective learning approach is represented in Fig. 1; the output of the learning system are N personalized learning systems, each for an individual in a population of N. In estimation, these two learning processes take place iteratively until convergence.

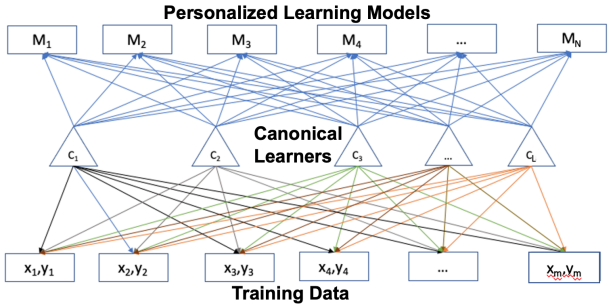

Fig. 1. Canonical learning model with L canonical learning models and N personalized processors

Our study is both different from and related to federated learning in the current literature [1], [2]. Unlike federated learning that is motivated from the fact that data across devices cannot be pulled together for learning either due to limited bandwidth or privacy concerns, the personalized and collective learning approach developed in this study permits access to all data. The two are also different in terms of their respective outputs: in federated learning, the output is a single global model learned by integrating all learning agents, while the outputs from this study are N personalized systems, each of which is a personalized one for an individual. Structurally, our approach is related to federated learning in that the canonical models developed in the study may be viewed as learning agents in the latter and the integration of all canonical models can be viewed as the single global model. How to do this integration is however beyond the scope of this study, which focuses on the development of a personalized and collection

approach to obtain N personalized models.

In previous work regression problems were considered using linear regression models [3] and binary classification problems were considered using logistic regression models [4]. Here we will consider binary classification models where we use kernel methods and optimization techniques based on a mean squared error cost function [5]. The key difference between these earlier papers and this paper is using kernel methods to find the learning model for each of the canonical learning models. Kernel methods allow for solving nonlinear regression and classification problems by using nonlinear transformations and working with kernels in the dual observation space. The problem is solved in the dual space using the representer theorem [6] which states that the learned function can be represented in terms of weighted sums of the evaluated kernels of the support vectors in the dictionary. Using a mean squared cost function and equality constraints this amounts to solving a least squares problem.

This paper gives a framework for personalized learning using kernel methods. Section II formulates the personalized learning models. Section III discusses algorithms for solving the model and Section IV discusses performance issues and variations to the algorithm. Section V discusses this learning framework from a broader societal perspective including a possible transportation application. Finally, Section VI summarizes this paper and discusses further directions. Note that this paper is a preliminary paper with a more complete performance analysis and application simulations to be presented at the conference.

II. Personalized Learning Model

Here we consider N personalized distributed systems at the edge. Each of these systems receives a set of unique data and processes the data to learn a specific task. Here we will assume a supervised learning task where the ith system receives the data $\{(x_i(1), y_i(1)), \dots (x_i(m_i), y_i(m_i))\}$ where $x_i(j) \in \mathcal{R}^n$ is an input vector and $y_i(j) \in \{-1, 1\}$ is the associated desired output. Let $X_i = [x_i(1), \dots, x_i(m_i)]$ and $Y_i = [y_i(1), \dots y_i(m_i)]^T$. The output is given by a mixture model where $\hat{y}_i(x) = \sum_{l=1}^{L} c_{i,l} f_l(x)$ with $c_{i,l}$ being nonnegative real numbers with $\sum_{l=1}^{L} c_{i,l} = 1$ and $c_i = [c_{i,1}, \dots, c_{i,L}]^T$. Here, $c_{i,l}$ represents the degree of affinity of the personalized learner i to the canonical model l. Finally, $\hat{Y}_i = [\hat{y}_i(x_i(1)), \dots, \hat{y}_i(x_i(m_i))]^T$ is the learned output for personalized learner i.

Here we consider learning with kernel functions based on Reproducing Kernel Hilbert Spaces (RKHS), [7]. A key result is that a function that is learned satisfies the Representer Theorem $f_i(x) = \sum_{l \in \mathcal{D}} \alpha_{i,l} k(x, x_l) + \alpha_{i,0}$, k is a real-valued kernel from $\mathcal{X} \times \mathcal{X} \to \mathcal{R}$ on a non-empty set $\mathcal{X}$ with a corresponding RKHS $\mathcal{H}$, $\mathcal{D}$ is a dictionary containing the support vectors, $D = |\mathcal{D}|$, and $\alpha_{i,l} \in \mathcal{R}$.

Let $\alpha_i = [\alpha_{i,0}, \alpha_{i,1}, \dots, \alpha_{i,D}]^T$ be a $D+1$ vector of the kernel weighting coefficients and threshold value with $A = [\alpha_1, \dots, \alpha_L]$ denoting the kernel weight matrix and $C = [c_1, \dots, c_N]$ denoting the canonical weight matrix.

For the binary classification problem, we have that $y_i(x) = \text{sign}((s_D \bullet K(x))Ac_i)$. Here $\bullet$ is the component by component multiplication of two row vectors. The $\bullet$ operation can also represent component by component multiplication of two column vectors or two matrices. Let $x_{\mathcal{D}}$ be the vector of support vectors in the dictionary and $\phi()$ be the transformation from input space to feature space. Then $K(x) = [1, < \phi(x), \phi(x_{\mathcal{D}}) >]$ is a $D+1$ row vector. Similarly $s_D = [1, y_D]$ is also a $D+1$ row vector where y_D is a row vector of all the target outputs of the dictionary of support vectors. We then have that

$$\hat{Y}_i = \text{sign}(((\mathbf{1_{m_i}} s_D) \bullet K_i) A c_i) \quad (1)$$

where $K_i = [K(x_i(1)); \dots; K(x_i(m_i))]$ is an $m_i \times (L+1)$ matrix and $\mathbf{1_{m_i}}$ is an m_i-length vector of 1s. Here we minimize the following cost function

$$J(A, C) = \frac{1}{2m} ||\mathbf{1} - Y(m) \bullet \hat{Y}(m)||^2 + \mathcal{R}(A, C, \mathbf{X}(m)) \quad (2)$$

where $m = \sum_{i=1}^{N} m_i$, $\mathbf{1}$ is a m-length vector of 1s, $Y(m)$ is an m vector that is a concatenation of the Y_is, $\hat{Y}(m)$ is an m vector that is a concatenation of the $\hat{Y}_i$s, and $\mathbf{X}(m)$ is a tensor of all the X_i matrices. We also have $\mathcal{R}$ is a regularization function.

Unlike Least Squares Support Vector Machine (LS-SVM) discussed in [5] which can be solved in primal or dual spaces by finding solutions to least squares problems (unconstrained convex optimization problems) minimizing equation (2) involves solving a nonconvex optimization problem. In the next section we discuss an algorithm that can find a solution to this problem.

Note the solution for this kernel problem is in terms of the dual observation space. When we use kernels as the basis functions to learn, the size of the learning problem grows as the number of observation data. We can limit the size of the support vector machines by adding support vectors if they satisfy a certain criterion that the support vectors sufficiently span the feature space such that linear combinations of support vectors approximate other input data well. The support vectors are added according to criterion used. Two popular criterion are the approximate linear dependence criterion (ALD), [8] and the coherence criterion (CC), [9]. It is shown that the number of support vectors in the dictionary for both criterion converge to a finite value. In this setup since we have multiple canonical learners we could even specify different kernels for each different canonical learner. The solution will then involve using multiple kernels [10]. A popular use of multiple kernels is using Gaussian kernels (Radial Basis function kernels) with different widths.

III. Iterative Solution to Optimization Problem

As mentioned above the overall optimization is not convex as we need to find the weights and threshold values, α_i and the weights associated with the canonical models, c_i. To solve the problem we construct an iterative two step procedure.

Canonical Learning Model (CLM)

1) initialize A and C.
2) fix C and solve for $\alpha_i, 1 \leq i \leq L$.
3) fix A and iteratively update $c_j, 1 \leq j \leq N$.
4) stop when certain criterion are met otherwise go to 2)

A. Finding A with C fixed

Referring to equation (1) we let $b_i = Ac_i$, for $1 \leq i \leq N$. Assuming the regularization term can also be written in terms of a quadratic function of b_i we can first solve N least squares problems to find a solution for the b_is. Then letting $B = [b_1, \ldots, b_N]$ we then solve a least squares problem to find A that minimizes $||B - AC||^2$.

B. Finding C with A fixed

With A fixed, the objective function in Equation 2 is a non-convex optimization problem with non-negative constraint on the parameter C. The algorithm developed in [3] could be adopted here. The basic idea is to derive the Lagrangian function that incorporates the non-negative constraint with the objective function, and further use the first derivative test and the complementary condition to derive an updating rule of the parameter C that will converge to a local optimal solution of C.

IV. Performance of Algorithm

The algorithm to solve for C using updating rule will cost $O(LND)$. The algorithm to solve for A involves solving N least squares problems of size D by solving for the matrix B. Then a least squares problem is solved for A given B and C. This cost is $O(LND^3)$. Together with the complexity for the algorithm for solving for A, the total complexity would be $O(LND^3)$.

Here we will also examine the performance of the learning algorithm in terms of convergence of the algorithm and the probability of error on both trained data and test data. This will depend on the number of support vectors in the dictionary, D and also the hyperparameters of the kernel methods used. The N personalized learners could also be described in terms of a graph structure giving relationships between the different learning models. This could be represented in the regularization function as

$$\mathcal{R}(A, C, \mathbf{X}(\mathbf{m})) = \frac{1}{2} w^T \mathbf{L} w$$

where w is the weight vector associated with each personalized learning model and $\mathbf{L}$ is the Laplacian of the graph structure. The weight vector w depends on the learning model and is a function of A and C.

V. Other Considerations and Transportation Application

Human choices are complex because there are a wide variety of factors that are at play and we, as researchers and analysts, are only aware of a small fraction of them. As an example, in deciding which mode of transportation to choose (e.g., driving vs taking transit), in addition to the usual structural factors such as travel cost and travel time, a wide range of other factors could be at play including, for example, weather, prior experience with transit, one's knowledge about and attitudes toward cars' impacts on the environment, or simply spur of the moment feelings etc. Furthermore, though there are multiple decision making theories explaining how different factors may interact with each other, when they are applied to explain real-world human choices, the amount of variations that can be explained are typically quite low, i.e., less than 30% and it is not uncommon for the variations to be less. This low model fitness is exactly because of the large amount of heterogeneity that are inherent in human choices. Because of the complexity involved in human choices, behavioral changes are challenging to model and predict, creating difficulties for identifying the right interventions that will most likely trigger the behavioral changes. This complexity also makes the use of kernel functions potentially a suitable avenue to model human choices. Unlike conventional machine learning models such as the logistic regression in our previous work for personalized learning [4] that represent subjects as fixed-sized vectors in a real space, kernels bypass the pre-processing step of feature vector generation and rather focus on measuring differences and similarities between subjects. This is done as each kernel evaluation performs an inner product in the feature space (which could be infinite dimensional) and provides relationships between the input entities. Here we use the kernel matrix (sometimes referred to as the Gram matrix) which gives information about the different entries (input data vectors and support vectors in dictionary). In the context of modeling human choices, this means that one no longer needs to first figure out in what form a single factor shall be represented and how factors interact with each other. In summary, using kernels allows us to capture potentially high degrees of interactions between different factors that are likely prevalent in human choices without having to specify them first.

A second advantage of using kernels in the context of canonical learners is that because it works in the dual space, the solution becomes essentially solving a set of least squares problems, which involve finding a set of solutions to a set of linear equations. The transformation of the problem to solving least squares problems is significant in a number of application contexts that were noted in the introduction of this paper, relating to public health, transportation, power and water systems. In those contexts, N is large, representing populations from hundreds of thousands to millions and even more. Let us take the morning commute as a potential application where in a large urban region, millions of commuters are simultaneously making choices such as which mode of transportation to take, when to depart from home and which route to take if one is to drive. If we want to learn the choice behavior of each individual, that means millions of learners are needed and they need to work at the same time. This could pose tremendous amount of pressure on computing and thus an efficient solution to solving a large system becomes very important.

The proposed kernel solutions are also suitable for real-time applications, during which updates need to be frequently

made. This can be done as the kernel methods considered here used a squared error cost function and online learning algorithms can be developed using principles of adaptive filtering [11]. Many real-world applications resemble large-scale games joined by the majority of a population. As an example, the daily commute may be viewed as a large-scale game joined by many. Every commuter has a number of route choices to choose from and the goal of a single commuter is to choose the route with the minimum travel time. And yet, the travel time of each possible route depends on the choices of all commuters. In other words, in such a system, the choice made by one individual at time t affects the state of entire system and consequently others' choices at time $t+1$, and every one is in the game of minimizing his/her own travel time. Because of this game-like dynamics, real-time updates of both the canonical learners (A) and the personalized weights (C) associated with each individual may be needed.

VI. Summary and Further Directions

This paper discusses personalized learning using a shared canonical learning structure. Some features of the personalized learning are that there are substantial similarities between each of the different learners, however there are some personalized differences which is accounted for in the canonical model. Kernel methods using a squared error cost function are used as they allow for the canonical models to efficiently learn nonlinear models while solving least squares problems in the dual observation space. The overall optimization problem is a nonconvex optimization problem involving a two step procedure to alternate between updating the kernel weight matrix A and the canonical weight matrix C.

This is a preliminary paper discussing the personalized kernel learning framework. Further directions include showing convergence of the algorithm and simulating the algorithm on a transportation application. These can then be compared to previous work which uses a logistic regression model [4]. Work will also proceed to make the learning algorithm more online so that real-time learning and decision making can be applied.

References

[1] J. Konečný, B. McMahan, and D. Ramage. Federated optimization:distributed optimization beyond the datacenter, 2015.

[2] P. Kairouz and et. al. Advances and open problems in federated learning, 2021.

[3] Y. Lin, K. Liu, E. Byon, X. Qian, S. Liu, and S. Huang. A collaborative learning framework for estimating many individualized regression models in a heterogeneous population. *IEEE Trans. on Reliability*, 67(1):328–341, Jan. 2018.

[4] J. Feng, X. Zhu, F. Wang, S. Huang, and C. Chen. A learning framework for personalized random utility maximization (rum) modeling of user behavior. *IEEE Trans. on on Automation Science and Engineering*, 17(early access):1–12, dec. 2020.

[5] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least squares support vector machines*. World Scientific Publishing Co., Singapore, 2002.

[6] B Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. computational learning theory. In *International Conference on Computational Learning Theory*, pages 416–426, Amsterdam, July 2001.

[7] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[8] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Trans. on Signal Proc.*, 52(3):2275–2285, Aug. 2004.

[9] C. Richard, J.-C. Bermudez, and P. Honeine. Online prediction of time series data with kernels. *IEEE Trans. on Signal Proc.*, 57(3):1058–1067, Mar. 2009.

[10] M. Yukawa. Multikernel adaptive filtering. *IEEE Trans. on Signal Proc.*, 60(9):4672–4682, Sept. 2012.

[11] S. Haykin. *Adaptive Filter Theory, 5th Ed.* Pearson, 2014.