Reducing the Environmental Impact of Aviation: A Data Mining Approach to Instantaneous Estimation of Fuel Consumption

Nikunj C. Oza, Ph.D.,
NASA Ames

Kamalika Das, Ph.D.,
UARC, NASA Ames

I. Purpose

The environmental impact of aviation is enormous given the fact that in the US alone there are nearly 6 million flights per year of commercial aircraft. This situation has driven numerous policy and procedural measures to help develop environmentally friendly technologies which are safe and affordable and reduce the environmental impact of aviation. However, many of these technologies require significant initial investment in newer aircraft fleets and modifications to existing regulations which are both long and costly enterprises. Additionally, there is a need for detailed characterization of the situations under which fuel consumption anomalies are more likely so that we can target our limited resources toward remedies that will have the highest impact. As a step toward this latter need, we demonstrate a comprehensive anomaly detection method based on Virtual Sensors (VS) to help detect overconsumption of fuel in aircraft. VS are algorithms that perform statistical estimation of one sensor measurement given other potentially non-linearly correlated sensor measurements. Our method relies only on the data recorded during flight of existing commercial aircraft and related operational information. In this paper, we describe and study the results of our approach to analyzing and detecting overconsumption where actual fuel consumption is compared to the statistically expected consumption.

II. Background

Commercial aircraft use a significant amount of fuel which account for a large percentage of the total operating costs of a commercial airline—some reports estimate as high as 30%—and means that the overall "carbon footprint" of the aviation system is substantial. Improvement in the fuel efficiency of these aircraft can have significant environmental and economic benefits to airline operators, airframe and engine manufacturers, and the public at large. Because fuel consumption represents such a large part of the operating cost for an airline, it is monitored and controlled very carefully through several procedures. These procedures essentially compare the total fuel consumed on a particular leg of a trip against other trips which have the same origin, destination, take-off weight, make-and-model of aircraft (and engine), flight time and durations of the phases of flight (such as take-off, cruise, descent, and landing), and other contextual factors. If a particular aircraft uses more fuel than expected, it may be targeted for maintenance and further investigation. We have seen in earlier work that there can be a wide degree of variability in the total fuel consumption for a particular aircraft on a given trip even after taking the previously identified factors into account. This variability poses a challenge and results in a classic signal-to-noise issue: is the observed high fuel usage within a specific flight truly high or just due to noise?

Excessive fuel consumption in jet engines not only leads to excessive expense, but also lead to increased carbon emissions. A typical Boeing 747 can carry nearly 184,000 liters of fuel. With a conservative assumption that 90% of the fuel is converted into carbon dioxide, and assuming a fuel density of 0.81 kg/l, we can estimate that a single Boeing 747 can emit as much as 134,000 kg of carbon into the high atmosphere during a long range flight. It is clear that when such emissions are multiplied by the number of aircraft in world-wide operation (estimated to be around 15,000 aircraft according to the Intergovernmental Panel on Climate Change (IPCC) [1], the environmental impact is substantial, especially due to the emissions being deposited at such high altitude [2].

III. Approach

We approach this problem using a regression models, which are designed to predict the value of one continuous variable given a set of other discrete and continuous variables. These models are extremely powerful and are used in numerous applications including finance, the social sciences, medicine, engineering systems, and related areas. The predictions are based on a model that is learned from training data that can then be applied to a test data set. Using a regression model, we predict the instantaneous fuel consumption of an aircraft given a vector of continuous and discrete variables that are measured concurrently. Although this paper presents the use of a concurrent state vector, the approach presented here generalizes to state vectors formed with information from the current time as well as past times. This input vector represents the instantaneous state of the aircraft. The intuition behind this model is that the fuel consumption may be a complex function of the state that can be learned from the data given an appropriate regression model. An issue that arises in this procedure is that we assume that the FOQA data used for training the models are from aircraft operating in nominal conditions. However, it could be that some subset of the training data could be from aircraft operating in off-nominal conditions. Throughout this text, we refer to nominal conditions as those that are according to plan or design, in consonance with the language used in aerospace engineering. We address this problem by requiring that we have a large number of flights for the same make and model of aircraft for a given city pair (origin and destination airport). The assumption that we make is that while a subset of the data may have off nominal characteristics, the vast majority operate in a nominal condition. We believe this assumption is valid because of the high degree of emphasis on tracking fuel consumption in modern fleets. Thus, any anomalies detected would be present in a small minority of flights.

We pose the problem as a regression problem where the input variables are the ones that affect fuel flow (such as airspeed, altitude, mass etc) and target variable is the instantaneous fuel flow. The overall approach described here is called Virtual Sensors, because we are developing an estimator of one sensor measurement (fuel consumption) given other potentially nonlinearly correlated sensor measurements. In the next few sections we describe our proposed distributed Support Vector Regression algorithm ParitoSVR, which is a Parallel Iterated Optimizer for Support Vector Regression in the Primal, that has been designed specifically for scaling the VS-based fuel consumption modeling to extremely large data sets, that was a bottleneck in the past study.

A. ADMM and SVR

Our ParitoSVR algorithm uses as a building block two components: (1) SVR and (2) Alternating Direction Method of Multipliers (ADMM). In this section, we discuss these two topics.

SVR: Support vector machine [3] is a powerful tool for a wide variety of regression and classification tasks, yielding good predictive performance on many datasets. In this section, we present a brief introduction to support vector regression.

Give m data tuples (training set) $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^n$ is the input and $y_i \in \mathbb{R}$ is the corresponding output or target, SVR solves the following optimization problem:

$$\min_{\mathbf{w},b} \left[\lambda||\mathbf{w}||^2 + \sum_{i=1}^{m} \ell_\varepsilon(\mathbf{w}\cdot\mathbf{x}_i + b - y_i) \right] \tag{1}$$

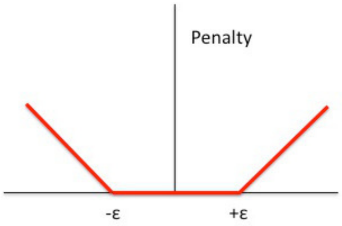


Fig. 1. *ε-insensitive loss function*

where λ is a constant and ℓ_ε is the *ε-insensitive loss function* defined as, $\ell_\varepsilon(r) = \max(|r| - \varepsilon, 0)$. In this formulation our goal is to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time is as fast as possible. In other words, we do not care about errors as long as they are less than ε, but will not accept any deviation larger than this, where ε is set to trade off between minimizing error (for which lower values are preferred) and reducing overfitting (for which higher values are preferred). Figure 1 shows the nature of this function. As can be seen, this is a convex optimization problem which can be solved using convex optimization solvers such as CVX[1].

ADMM: ADMM [4] is a decomposition algorithm for solving separable convex optimization problems of the form:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{y}} \quad G_1(\mathbf{x}) + G_2(\mathbf{y}) \\ &\text{subject to} \quad A\mathbf{x} - \mathbf{y} = 0, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{y} \in \mathbb{R}^m \end{aligned} \tag{2}$$

where $A \in \mathbb{R}^{m\times n}$ and G_1 and G_2 are convex functions. ADMM is an iterative technique and the update equations are:

$$\begin{aligned} \mathbf{x}^{t+1} &= \min_{\mathbf{x}} \left\{ G_1(\mathbf{x}) + \rho/2 \left\| A\mathbf{x} - \mathbf{y}^t + \mathbf{p}^t \right\|_2^2 \right\} \\ \mathbf{y}^{t+1} &= \min_{\mathbf{y}} \left\{ G_2(\mathbf{y}) + \rho/2 \left\| A\mathbf{x}^{t+1} - \mathbf{y} + \mathbf{p}^t \right\|_2^2 \right\} \\ \mathbf{p}^{t+1} &= \mathbf{p}^t + A\mathbf{x}^{t+1} - \mathbf{y}^{t+1} \end{aligned}$$

where $\mathbf{p} = (1/\rho)\mathbf{z}$. ADMM effectively decouples the $\mathbf{x}$ and $\mathbf{y}$ updates such that parallel execution becomes possible. In a distributed computing framework, this becomes even more interesting since each computing node can now solve a (smaller) subproblem in $\mathbf{x}$ independently, and then, these solutions can be efficiently gathered to compute the consensus variable $\mathbf{y}$ and the dual variable $\mathbf{p}$. ADMM converges within a few iterations when moderate precision is required. This can be particularly useful for many large scale problems, similar to what we consider here.

In the next section we show how to build SVR models for very large datasets using distributed computing via the ADMM technique.

IV. ParitoSVR formulation

For the linear ParitoSVR algorithm setup, we assume that the training data is distributed among N client

[1] http://cvxr.com/cvx/

processors (nodes) $P_1, \ldots, P_N$ with a central machine P_0 acting as the server or collector. The dataset at machine P_j, denoted by D_j, consists of m_j data points *i.e.* $D_j = \left\{\mathbf{x}_i^{(j)}, y_i^{(j)}\right\}_{i=1}^{m_j}$. It is assumed that the datasets are disjoint: $D_i \bigcap D_j = \emptyset$ and $\bigcup_{j=1}^{N} D_j = D$, where D is the total (global) data set. The goal is to learn a linear support vector regression model on D without exchanging all of the data among all the nodes.

Given Eqn. 1, the optimization problem is now:

$$\min_{\mathbf{w}} \left[\sum_{i=1}^{m} \ell_\varepsilon(\mathbf{w} \cdot \mathbf{x}_i - y_i) + \lambda ||\mathbf{w}||^2 \right]$$

$$\Leftrightarrow \min_{\mathbf{w}} \left[\sum_{j=1}^{N} \sum_{i=1}^{m_j} \ell_\varepsilon \left(\mathbf{w} \cdot \mathbf{x}_i^{(j)} - y_i^{(j)} \right) + \lambda \left\| \mathbf{w} \right\|^2 \right]$$

The inner sum can be computed by each node independently (assuming that $\mathbf{w}$ is known). We next write it in a form such that it is decoupled across the nodes:

$$\min_{\mathbf{w}_1, \ldots, \mathbf{w}_N, \mathbf{z}} \left[\sum_{j=1}^{N} \sum_{i=1}^{m_j} \ell_\varepsilon \left(\mathbf{w}_j \cdot \mathbf{x}_i^{(j)} - y_i^{(j)} \right) + \lambda \left\| \mathbf{z} \right\|^2 \right] \quad (3)$$
$$\text{subject to} \quad \mathbf{w}_j = \mathbf{z}$$

In the ADMM decomposition, each node can solve its local problem using its own data and optimization variable and then coordinate the results across the nodes to drive them into consensus. The nodes update the consensus variable $\mathbf{z}$ iteratively, based on their local data and scatter-gather operations on $\mathbf{z}$ until they converge to the same result.

Theorem 4.1: The ADMM update rules for the linear support vector regression primal optimization are:

$$\begin{aligned}
\mathbf{w}_j^{t+1} &= \min_{\mathbf{w}_j} \left\{ \sum_{i=1}^{m_j} \ell_\varepsilon \left(\mathbf{w}_j \cdot \mathbf{x}_i^{(j)} - y_i^{(j)} \right) + \frac{\rho}{2} \left\| \mathbf{w}_j - \mathbf{z}^t - \mathbf{u}_j^t \right\|_2^2 \right\} \\
\mathbf{z}^{t+1} &= \min_{\mathbf{z}} \left\{ \lambda \left\| \mathbf{z} \right\|_2^2 + \frac{N\rho}{2} \left\| \mathbf{z} - \overline{\mathbf{w}}^{t+1} - \overline{\mathbf{u}}^t \right\|_2^2 \right\} \\
\mathbf{u}_j^{t+1} &= \mathbf{u}_j^t + \mathbf{w}_j^{t+1} - \mathbf{z}^{t+1}
\end{aligned}$$

where $\mathbf{u} \in \mathbb{R}^n$ is the (scaled) dual variable and $\overline{\mathbf{w}}^{t+1}$ and $\overline{\mathbf{u}}^{t+1}$ are the averages of the variables over all the nodes.

The $\mathbf{w}$ update can be executed in parallel for each machine. It involves solving a convex optimization problem in $n+1$ variables at each node. This solution depends only on the data available at that partition. The $\mathbf{z}$ update step involves computing the average of the $\mathbf{w}$ and $\mathbf{u}$ vectors in order to combine the results from the different partitions. Critical to the working of ADMM is the convergence criteria. The primal and dual residuals can be written as:

$r_p^t = \left\| \mathbf{w}^t - \mathbf{z}^t \right\|_2^2$ and
$r_d^t = \left\| \rho(\mathbf{z}^t - \mathbf{z}^{t-1}) \right\|$

Also, given the thresholds ϵ_{pri} and ϵ_{dual}, the primal and dual thresholds can be written as,

$\epsilon_{pri} = \epsilon_{abs} \sqrt{m} + \epsilon_{rel} \max(\left\| \mathbf{w} \right\|, \left\| \mathbf{z} \right\|)$ and
$\epsilon_{dual} = \epsilon_{abs} \sqrt{m} + \rho \epsilon_{rel} \left\| \mathbf{u} \right\|$.

The iterations terminate when $r_p^t < \epsilon_{pri}$ and $r_d^t < \epsilon_{dual}$.

The pseudo code of the ParitoSVR for the linear case is presented in Alg. 1, Alg. 2, and Alg. 3. Alg. 1 is the driver which calls the **ADMM_Linear_SVR** routine to split the data into N chunks. It then calls the **RunDistributedJob** function in parallel for these N subproblems to find the $\mathbf{w}$ minimizer. Then it aggregates the results and updates the $\mathbf{z}$ and $\mathbf{u}$ variables. This process is repeated until the primal and dual residuals fall below the thresholds or the total number of iterations exceed MAXITER.

```
Input: $D$, $\varepsilon$, $\lambda$, $\rho$, $N$, $MaxIter$
Output: $\mathbf{w}$ of the SVR model
Initialization: Initialize $\mathbf{z}^0$, $\mathbf{u}^0$

Split $D$ into $D_1, \ldots, D_N$;
Call ADMM_Linear_SVR($D_1, \ldots, D_N$, $\varepsilon$, $\lambda$, $\rho$, $N$, $MaxIter$)
```

Algorithm 1: Linear ParitoSVR

```
Procedure ADMM_Linear_SVR($D_1, \ldots, D_N$, $\varepsilon$, $\lambda$, $\rho$, $N$, $MaxIter$)
forall the $t$=1 to $MaxIter$ do
    forall the $j$=1 to $N$ do
        $\mathbf{w}_j^t$ = RunDistributedJob($D_j$, $\varepsilon$, $\lambda$, $\rho$, $\mathbf{u}^{t-1}$, $\mathbf{z}^{t-1}$);
    end
    $\mathbf{z}^t = \frac{N\rho}{2\lambda + N\rho} \left( \overline{\mathbf{w}^t} + \overline{\mathbf{u}^t} \right)$;
    forall the $j$=1 to $N$ do
        $\mathbf{u}_j^t = \mathbf{u}_j^{t-1} + \mathbf{w}_j^t - \mathbf{z}^t$;
    end
    if $r_p^t < \epsilon_{pri}$ and $r_d^t < \epsilon_{dual}$ then
        break;
    end
end
Return $\overline{\mathbf{w}}$
```

Algorithm 2: Procedure ADMM_Linear_SVR

```
Procedure RunDistributedJob($\mathcal{D}$, $\varepsilon$, $\lambda$, $\rho$, $\mathbf{u}^{t-1}$, $\mathbf{z}^{t-1}$)
$\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{m_j}$
$\mathbf{w} = \min_{\mathbf{w}} \left\{ \sum_{i=1}^{m_j} \ell_\varepsilon (\mathbf{w} \cdot \mathbf{x}_i - y_i) + \frac{\rho}{2} \left\| \mathbf{w} - \mathbf{z}^{t-1} + \mathbf{u}^{t-1} \right\|_2^2 \right\}$
Return $\mathbf{w}$
```

Algorithm 3: Procedure RunDistributedJob

A. ParitoSVR performance

ParitoSVR has been implemented in MATLAB 2011b. The experiments have been executed in a 64-bit Linux cluster consisting of 16 slave nodes where each node is a dual processor 1-U server containing two quad-core Intel Xeon 2.66GHz processors totaling 128 cores and 128GB Ram (1Gb/Core). For solving the convex problems at each iteration, we have used the convex optimization toolbox CVX for Matlab[2]. For evaluating the accuracy of prediction of our algorithm, we have measured the RMSE

[2] http://cvxr.com/cvx/

metric on the test dataset defined as:

$$\frac{1}{\max\limits_{i=1:m} y_i} \sqrt{\frac{1}{m} \sum_{i=1}^{m} (f(\mathbf{x}_i) - y_i)}$$

Fig. 2 shows the sample dataset generated from a linear model following $y = \mathbf{w} \times \mathbf{x} + noise$, where $\mathbf{w}$ is the weight of the regression model. We have used 2 nodes in this experiment and, for each node, chosen a different $\mathbf{w}$ vector so that each node sees a different data distribution. The data of the two nodes are shown in two different colors (circle and plus markers). Also shown in the figure are the models (straight lines) formed by node 1 at different iterations of ParitoSVR algorithm. As seen, the algorithm updates the model as iterations proceed. For comparison, we have also plotted the centralized model formed by the union of all the data. As clearly shown, the ParitoSVR model in the final iteration is very close to the model formed by the centralized model. Note that, at convergence, all nodes have the same model. Fig. 2(b) shows the primal and dual residuals (r_p and r_d) in solid blue lines and the primal and dual thresholds ϵ_{pri} and ϵ_{dual} in red dotted lines. Fig. 2(c) shows the objective values for different iterations of the algorithm. For this experiment, we have used the following values of the parameters: $\varepsilon = 10, \lambda = 1, \epsilon_{pri} = 10^{-4}, \epsilon_{dual} = 10^{-2}$. The values of these parameters are chosen to optimize the running time (convergence) as well as accuracy of results based on trial and error.

V. Results

We use ParitoSVR algorithm to detect anomalous fuel consumption in a fleet of commercial aircraft from an European carrier that we call Carrier X. We model the average fuel flow as a function of 29 different parameters that are listed below:

- left and right ailerons
- left and right elevators
- rudder
- stabilizer
- roll, pitch, and side-slip angles
- angle-of-attack
- mach number
- left and right engine thrust
- lateral, vertical, and longitudinal velocities
- lateral, vertical, and longitudinal accelerations
- gross weight
- altitude
- wind speed and direction
- aileron differential
- roll and pitch rate
- roll and pitch acceleration
- dynamic pressure

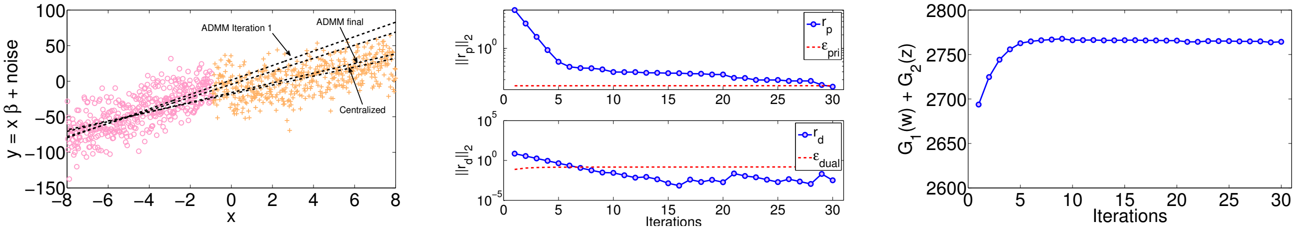
Based on domain expert feedback, these parameters are expected to influence fuel flow the most. The study that we conducted is based on tail numbers. The target variable for us is the average fuel flow in the two engines.

The fuel flow anomaly detection study has been designed based on tail numbers. For the last 6 month period of flights, for each tail number we train on the first three months of data and we test on the remaining 3 months. We get rid of anomalous flights in the training data by running the model on the training set and identifying all flights that have a mean error (instantaneous RMSEs averaged over all time instances for a flight) higher than the 3-σ bounds for error for that tail number. Also, given that the profiles for the different flight phases are significantly different from one another, our hypothesis is that the fuel flow models corresponding to the different phases will be significantly different from each other. Therefore, in this study, we focus only on the cruise phase of flights.

In our data set there are a total of 188 tail numbers with at anywhere between 440 to 560 flights in the training period (first three months of data) and approximately 400 to 500 flights in the test period (last three months of the observation data). For every tail number, once we build the model, we test the remaining flights for instantaneous fuel flow predictions using the model. We then compute the RMSE for every prediction for a flight. We then compute the mean RMSE and the standard deviation (σ) of RMSE across all flights. Anomalous flights are those for which the mean RMSE over all time instances is outside the 3-σ bound for the error distribution. For the 188 tail numbers that we study, the number of anomalous flights for any tail number varies anywhere between 1 to 14 anomalies based on the 3-σ condition violation. In this report we present the results for one of those 188 tail numbers which has the most anomalies.

Out of 495 test flights over a 3-month period, 14 flights were determined to be anomalous. Figure 3(a) shows the mean squared errors for each of the flights in blue and the 3-σ bounds in magenta. The instantaneous fuel flow for the top ranked anomalous flight among these 14 flights is shown in Figure 3(b). The red graph depicting observed fuel flow is significantly higher than the predicted fuel consumption, shown in blue. Fig.3(c) shows the same for a normal flight.

Figure 4 shows the distribution of the 29 parameters for the nominal test flights as opposed to the 14 anomalous test flights. It is somewhat obvious that for certain parameter combinations, the distribution of parameters deviate from that of the nominal flights. On further investigation it was revealed that the airspeed, gross weight, and altitude configurations for the 14 anomalous flights are clearly, distinct from the rest of the nominal flights. Figure 5 shows these configurations in a scatter plot against those of the nominal flights. On further analysis of the anomalous flights it was also revealed that out of the 14 flights, the top anomalous flight was probably a grounded flight since its origin and destination airport was the same. An earlier flight on the same day has also been detected as

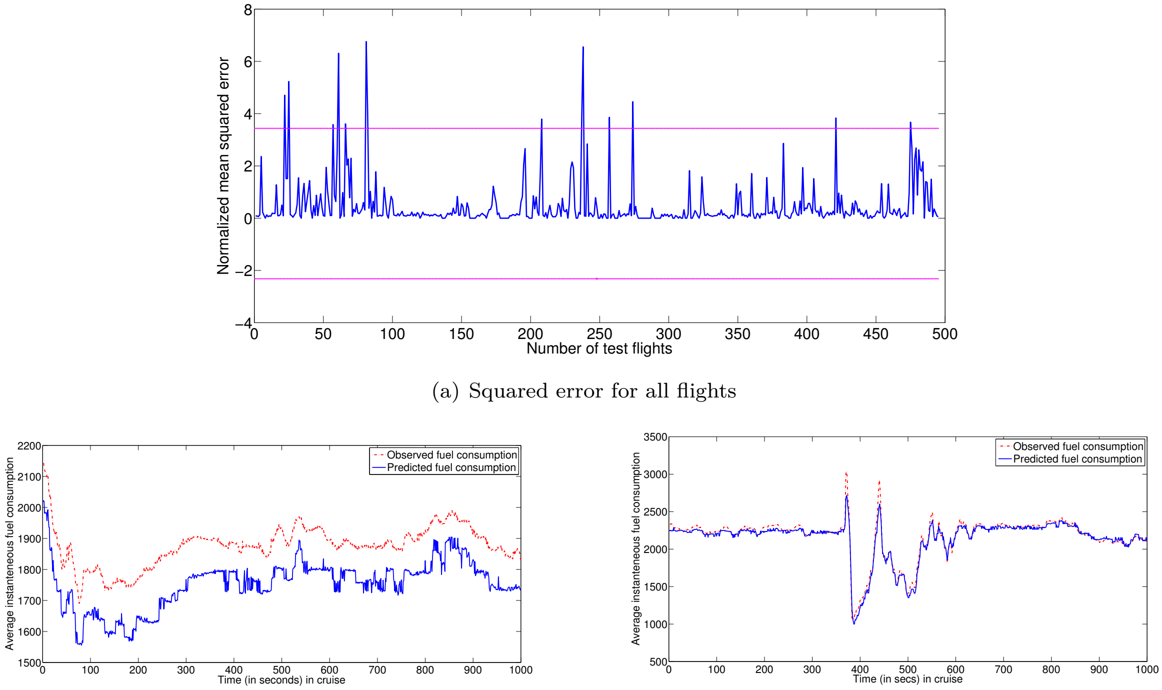


(a) Synthetic dataset

(b) Primal and dual residuals

(c) Objective function

Fig. 2. (a) Synthetic dataset. Also shown are the models formed by node 1 as the algorithm progresses. (b) Primal and dual residuals (blue solid line) for different iterations of ParitoSVR algorithm. The red dotted lines show the ϵ_{pri} and ϵ_{dual}. (c) Change in objective value with different iterations.



(a) Squared error for all flights

(b) Outlier flight fuel consumption

(c) Normal flight fuel consumption

Fig. 3. Fuel flow study on CarrierX dataset. Fig. (a) shows squared error for all test flights, the 3-σ bound and flights which cross the threshold. Fig. (b) shows the observed and predicted fuel flow of top ranked anomalous flight. Fig. (c) shows the same for a normal flight.

an anomaly. Similarly there are two other pairs of flights that have been reported in the top 14 anomalies that flew consecutive flights on the same day or in two consecutive days to the same airport pairs. We are currently working with the airlines to identify additional information so that we can make a decision in terms of the root causes for these anomalous fuel consumptions in these flights. It needs to be investigated whether it was pilot or air traffic controller decisions, or mechanical issues, or external conditions such was weather that was responsible for the excess fuel consumption in these flights compared to other flights for the same tail numbers.

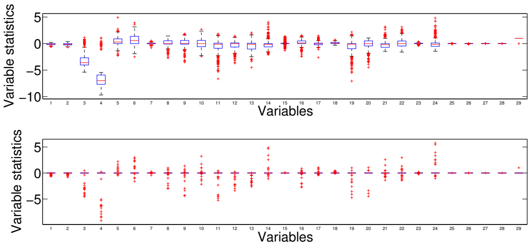
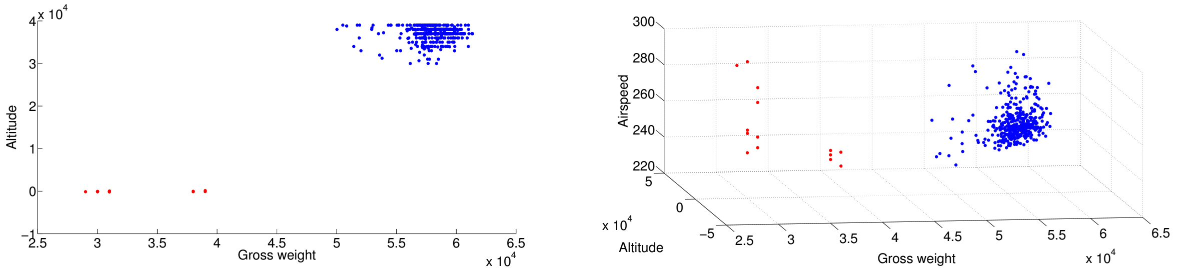


Fig. 4. Distribution of parameters for nominal test flights and anomalous test flights

VI. Conclusion and future work

In this report we describe ParitoSVR — a parallel iterated optimizer based support vector regression technique



(a) Gross weight - altitude configurations for the anomalous flights

(b) Gross weight - airspeed - altitude configurations for the anomalous flights

Fig. 5. Joint distribution of gross weight, altitude, and airspeed parameters for the test flights. Red dots indicate identified anomalies. Blue dots are all other test flights.

that uses Virtual Sensors technology for identifying flights that consume more than nominal fuel. The algorithm is highly parallelizable and therefore leads to good scalability on very large aviation datasets. Unlike many related approaches, we solve the actual optimization problem in the primal, leading to very low error rates. However, analyzing the anomalies for root cause identification is still a challenge and requires human intervention at every step. As part of future work, we plan to develop a suite of technique that would allow us to localize the variables which cause the anomalous behavior. We also plan to extend the method to learn non-linear kernel based SVR which might improve accuracy due to better modeling. Finally, we would like to run a through study for comparing the performance of ParitoSVR with the ensemble based method developed during phase 1 of this project and identify the strengths of each method.

References

[1] J. E. Penner and D. H. Lister, “Global aviation and the atmosphere in intergovernmental panel on climate change„” Tech. Rep., 1999.

[2] J. A. Dutton and D. Bahr, “For greener skies: Reducing environmental impacts of aviation,” Tech. Rep., 2002.

[3] V. V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Found. and Trends in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.