On Multivariate Recursive Least Squares and Extensions

W. Cannon Lewis II[*]

April 8, 2020

Abstract

Least squares estimation—also known as linear regression—is one of the fundamental tools underlying modern data science and machine learning. Its typical exposition assumes a fixed dataset which is analyzed as a whole, but this assumption is violated when data arrives in a stream over time. The least squares estimate can instead be computed online using an algorithm known as recursive least squares. In this note we will derive the update equations for recursive least squares applied to both centered and uncentered data. Additionally, we will draw connections between practical implementations of recursive least squares and l^2-regularized least squares, which is also known as ridge regression.

Contents

1 Notation **2**

2 Problem Formulation **2**

2.1 Static Least Squares 2

3 Centered Data **2**

3.1 Breaking Up the Normal Equation 3

3.2 Deriving the $\hat{\Theta}$ Update 3

3.3 Deriving the P_T Update 3

4 Uncentered Data **4**

4.1 Centered X, Uncentered Y 4

4.2 Uncentered X, Centered Y 5

4.2.1 Deriving the Q_T Update 5

4.2.2 Deriving the $\hat{\Theta}$ Update 6

4.3 Uncentered X, Uncentered Y 7

5 Practical Extensions **7**

5.1 Why Can't We Just Modify the Feature Vector? 7

5.2 Initializing P_T and Q_T and Connections to Ridge Regression 8

5.3 Forgetting Factors for Time-Varying Systems 8

[*] Rice University Computer Science Department (cannontwo.com)

1 Notation

Symbol	Space	Meaning
n	$\mathbb{N}$	Input feature dimension
m	$\mathbb{N}$	Output dimension
φ_t	$\mathbb{R}^{1\times n}$	Feature vector at time t
$\mu_x(t)$	$\mathbb{R}^{1\times n}$	Feature mean at time t
X_t	$\mathbb{R}^{t\times n}$	Design matrix of stacked feature vectors from timesteps 0 to t
y_t	$\mathbb{R}^{1\times m}$	Output vector at time t
$\mu_y(t)$	$\mathbb{R}^{1\times n}$	Output mean at time t
Y_t	$\mathbb{R}^{t\times m}$	Output matrix of stacked output vectors from timesteps 0 to t
Θ	$\mathbb{R}^{n\times m}$	True linear model coefficients
$\hat{\Theta}$	$\mathbb{R}^{n\times m}$	Estimated model coefficients
P_t	$\mathbb{R}^{n\times n}$	Inverse sample covariance matrix
Q_t	$\mathbb{R}^{n\times n}$	Inverse mean-corrected sample covariance matrix
R_t	$\mathbb{R}^{n\times n}$	Rank-2 update to corrected sample covariance matrix
V_t	$\mathbb{R}^{2\times n}$	$\begin{bmatrix}\mu_x(t-1)^\top & \varphi_t^\top\end{bmatrix}^\top$
$\bar{1}$	$\mathbb{R}^{t\times 1}$	A vector whose entries are all 1

Table 1: Notation used in this paper

In addition to the table above, we use the notation $x^\top$ to represent the transpose of a matrix x.

2 Problem Formulation

Let us define a data stream as a function $F(t) : \mathbb{N} \to \mathbb{R}^{1\times n} \times \mathbb{R}^{1\times m}$ that, for $t \in \mathbb{N}$, can be defined as

$$F(t) := (\varphi_t, y_t) \tag{1}$$

where φ_t and y_t are related according to

$$y_t = \varphi_t\Theta + \epsilon_t, \quad \mathbb{E}\left[\epsilon_t\right] = 0 \tag{2}$$

Note that we assume that the output of the underlying linear model with true parameters Θ is corrupted at each time step by some additive, zero-mean white noise ϵ_t. Since the purpose of this note is not to explore the statistical properties of recursive least squares, we skip over further clarification of the properties of this noise; results follow from the standard statistical analysis of linear regression [1].

2.1 Static Least Squares

If we wait to observe $T \in \mathbb{N}$ time steps of input-output pairs, we can assemble the matrices X_T and Y_T by vertically stacking the φ_t and y_t observations for $t = 1, \ldots, T$. Given these matrices, the least squares estimation problem is formulated as

$$\min_{\hat{\Theta}} = \sum_{t=1}^{\top} ||y_t - \varphi_t\hat{\Theta}||_2^2 \tag{3}$$

$$= ||Y_T - X_T\hat{\Theta}||_2^2 \tag{4}$$

It is well known (see, e.g., [1]) that the solution to this problem is given by the "normal equation"

$$\hat{\Theta}_{LS}(T) := \left(X_T^\top X_T\right)^{-1} X_T^\top Y_T \tag{5}$$

3 Centered Data

From Equation 5 we can begin to derive the recursive least squares estimator. It is worth noting at the beginning of this derivation that we have implicitly assumed that our data is already centered; in other words, the relationship between φ_t

and y_t has no offset term, and so our model $\hat{\Theta}_{LS}$ will always predict an output vector of all zeros for an input vector of all zeros. This is a fine assumption when all of the data has been collected ahead of time, but breaks down when we want to do recursive least squares because we cannot estimate the means of our inputs and outputs ahead of time. Since it is easier to derive the recursive least squares estimator in the centered case than in the uncentered case, we tackle this limited version first.

3.1 Breaking Up the Normal Equation

We begin by writing out the normal equation as two sums multiplied together

$$\begin{aligned}\hat{\Theta}_{LS}(T) &= (X_T^\top X_T)^{-1} X_T^\top Y_T && (6)\\ &= \left[\sum_{t=1}^{T} \varphi_t^\top \varphi_t\right]^{-1} \left[\sum_{t=1}^{T} \varphi_t^\top y_t\right] && (7)\end{aligned}$$

Let us define the inverse sample covariance matrix P_T to be the left-hand term in Equation 7, so that we then have

$$\begin{aligned}P_T^{-1} &= \sum_{t=1}^{T} \varphi_t^\top \varphi_t && (8)\\ &= P_{T-1}^{-1} + \varphi_T^\top \varphi_T && (9)\\ \implies P_{T-1}^{-1} &= P_T^{-1} - \varphi_T^\top \varphi_T && (10)\end{aligned}$$

Similarly, we can break up the right-hand term in Equation 7:

$$\sum_{t=1}^{T} \varphi_t^\top y_t = \sum_{t=1}^{T-1} \varphi_t^\top y_t + \varphi_T^\top y_T \quad (11)$$

3.2 Deriving the $\hat{\Theta}$ Update

Our normal equation is now

$$\hat{\Theta}_{LS} = P_T \cdot \left[\sum_{t=1}^{T-1} \varphi_t^\top y_t + \varphi_T^\top y_T\right] \quad (12)$$

Using the definition of $\hat{\Theta}_{LS}(T-1)$, we get

$$\hat{\Theta}_{LS}(T) = P_T \cdot \left[P_{T-1}^{-1} \hat{\Theta}_{LS}(T-1) + \varphi_T^\top y_T\right] \quad (13)$$

Substituting in Equation 10:

$$\begin{aligned}\hat{\Theta}_{LS}(T) &= P_T \cdot \left[(P_T^{-1} - \varphi_T^\top \varphi_T)\hat{\Theta}_{LS}(T-1) + \varphi_T^\top y_T\right] && (14)\\ &= \hat{\Theta}_{LS}(T-1) - P_T \varphi_T^\top \varphi_T \hat{\Theta}_{LS}(T-1) + P_T \varphi_T^\top y_T && (15)\\ &= \hat{\Theta}_{LS}(T-1) + P_T \varphi_T^\top \left[y_T - \varphi_T \hat{\Theta}_{LS}(T-1)\right] && (16)\end{aligned}$$

Note that the last term in Equation 16 ($y_T - \varphi_T \hat{\Theta}_{LS}(T-1)$) is the prediction error of our model at timestep $T-1$ on the new datum, so the new estimate of Θ that we get is the old estimate plus the prediction error on a new datum filtered by $P_T \varphi_T^\top$. Intuitively, this represents a reweighting of the prediction error using our existing estimate of the sample covariance, which effectively rescales the update to Θ to take into account the scales of the coordinates of the features.

3.3 Deriving the P_T Update

Though Equation 9 gives us a way to update P_T^{-1} easily with each new datum, to recover P_T and update $\hat{\theta}_{LS}$ we would need to invert an $n \times n$ matrix at on each time step. This is not only computationally expensive for all but small values of n, it also introduces the risk of running into floating-point errors if P_T^{-1} ever becomes ill-conditioned[1].

We can get around these issues by doing away with P_T^{-1} all together and deriving a direct update for P_T. We do this with the Woodbury matrix identity [3], also known as the "matrix inversion lemma." This result tells us that for matrices

[1] These sorts of issues can also be dealt with using any number of techniques from numerical linear algebra [2]

A, U, C, V such that UCV has rank k, the inverse of the rank-k update is given by:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1} \tag{17}$$

In our case, since we are doing a rank-1 update $\varphi_T^\top \varphi_T$ to P_{T-1}^{-1}, this lemma gives us that

$$P_T = \left(P_{T-1}^{-1} + \varphi_T^\top \varphi_T\right)^{-1} = P_{T-1} - \frac{P_{T-1}\varphi_T^\top \varphi_T P_{T-1}}{1 + \varphi_T P_{T-1}\varphi_T^\top} \tag{18}$$

And just like that, we're done! Equations 16 and 18 give us the update equations that define the recursive least squares algorithm. At each timestep t, we simply need to:

1. Record φ_t and y_t from our datastream $F(t)$.
2. Calculate P_t from P_{t-1} and φ_t using Equation 18.
3. Calculate $\hat{\Theta}_{LS}(t)$ from $\hat{\Theta}_{LS}(t-1)$, φ_t, y_t, and P_t using Equation 16.

4 Uncentered Data

In the general linear regression setting, we cannot assume that the data is centered. We might have a persistent constant offset vector added to the input features or the output which will cause the centered recursive least squares estimator to be inaccurate or have worse generalization. In the static data regime, estimation of this constant offset can be done prior to solving the least squares problem by calculating the feature and output means. In online estimation, we need to not only update the means at each timestep but also to correct the previous parameter estimate with respect to the new mean estimate. While the algebra becomes a bit more complex, the eventual structure of the update equations is remarkably similar to the uncentered case.

In order to contain the complexity of the following derivation, we proceed in stages. First, we will consider the case when the input features are centered but the output is not. Then we will consider the inverse case, where the input features are not centered but the output is. We will then see how these two cases can be combined in the general uncentered recursive least squares estimator.

4.1 Centered X, Uncentered Y

In this case we want to solve the problem

$$(y_T - \mu_y(T)) = \varphi_T \hat{\Theta}(T) \tag{19}$$

When the output data we are provided by $F(t)$ are not already centered, we center it prior to solving the least squares problem. Note that it is simple to calculate the output mean online, since

$$\mu_y(T) = \frac{1}{T}\sum_{t=1}^{T} y_t \tag{20}$$

$$= \frac{T-1}{T}\mu_y(T-1) + \frac{1}{T}y_T \tag{21}$$

$$\implies T\mu_y(T) = (T-1)\mu_y(T-1) + y_T \tag{22}$$

$$= T\mu_y(T-1) + y_T - \mu_y(T-1) \tag{23}$$

$$\implies \mu_y(T) = \mu_y(T-1) + \frac{1}{T}(y_T - \mu_y(T-1)) \tag{24}$$

Thus the normal equation in this case is given by

$$(X_T^\top X_T)^{-1}X_T^\top\left(Y_T - \bar{1}\mu_y(T)\right) = (X_T^\top X_T)^{-1}X_T^\top\left(Y_T - \bar{1}\frac{1}{T}\sum_{t=1}^{T} y_t\right) \tag{25}$$

$$= (X_T^\top X_T)^{-1}X_T^\top Y_T - (X_T^\top X_T)^{-1}X_T^\top \bar{1}\frac{1}{T}\sum_{t=1}^{T} y_t \tag{26}$$

Note that the first term in Equation 26 is just the normal equation for the centered least squares problem that we already derived recursive update equations for, so all that remains is expanding the second term as

$$\begin{aligned}(X_T^\top X_T)^{-1} X_T^\top \bar{1}\mu_y(T) &= P_T \sum_{t=1}^{T} \varphi_t^\top \mu_y(T) \qquad (27)\\ &= T \cdot P_T \mu_x(T)^\top \mu_y(T) \qquad (28)\end{aligned}$$

Of course, in the current case $\mu_x = \bar{0}$, so this correction term is actually zero and we recover the same update equations as previously derived. However, Equation 28 will come in handy later when we derive the general uncentered update.

Even though the update equations are the same, the final prediction of our model includes a constant offset term that we can derive from the model equation

$$\begin{aligned}(y_T - \mu_y(T)) &= \varphi_T \hat{\Theta}_{LS}(T) \qquad (29)\\ \implies y_T &= \varphi_T \hat{\Theta}_{LS}(T) + \mu_y(T) \qquad (30)\end{aligned}$$

4.2 Uncentered X, Centered Y

In this case we want to solve the problem

$$y_T = (\varphi_T - \mu_x(T))\hat{\Theta}(T) \tag{31}$$

In the same way that we calculated the update equation for μ_y, we calculate

$$\mu_x(T) = \mu_x(T-1) + \frac{1}{T}(\varphi_T - \mu_x(T-1)) \tag{32}$$

The normal equation for this case is given by

$$\begin{aligned}&\left[(X_T - \bar{1}\mu_x(t))^\top (X_T - \bar{1}\mu_x(t))\right]^{-1} (X_T - \bar{1}\mu_x(t))^\top Y = \\ &\qquad \left[(X_T^\top X_T - X_T^\top \bar{1}\mu_x(T) - (\bar{1}\mu_x(T))^\top X_T + \mu_x(T)^\top \mu_x(T)\right]^{-1} (X_T - \bar{1}\mu_x(t))^\top Y \qquad (33)\end{aligned}$$

Note that

$$\begin{aligned}X_T^\top \bar{1}\mu_x(T) &= \sum_{t=1}^{T} \varphi_t^\top \mu_x(T) \qquad (34)\\ &= T\mu_x(T)^\top \mu_x(T) \qquad (35)\end{aligned}$$

Thus Equation 33 becomes

$$\begin{aligned}&\left[(X_T^\top X_T - 2T\mu_x(T)^\top \mu_x(T) + \mu_x(T)^\top \mu_x(T)\right]^{-1} (X_T - \bar{1}\mu_x(t))^\top Y = \\ &\qquad \left[(P_T^{-1} - (2T-1)\mu_x(T)^\top \mu_x(T)\right]^{-1} (X_T - \bar{1}\mu_x(t))^\top Y \qquad (36)\end{aligned}$$

Let us define, analogously to P_T from before,

$$Q_T := \left[P_T^{-1} - (2T-1)\mu_x(T)^\top \mu_x(T)\right]^{-1} \tag{37}$$

4.2.1 Deriving the Q_T Update

As with P_T, we begin by developing an update for Q_T^{-1} in terms of Q_{T-1}^{-1}

$$\begin{aligned}Q_T^{-1} &= P_T^{-1} - (2T-1)\mu_x(T)^\top \mu_x(T) \qquad (38)\\ &= P_{T-1}^{-1} + \varphi_T^\top \varphi_T - \frac{2T-1}{T^2}((T-1)\mu_x(T-1) + \varphi_T)^\top ((T-1)\mu_x(T-1) + \varphi_T) \qquad (39)\end{aligned}$$

As a side computation, and to avoid stacking even longer equations, let $\mu := \mu_x(T-1)$ and note

$$((T-1)\mu + \varphi_T)^\top ((T-1)\mu + \varphi_T) = (T-1)^2 \mu^\top \mu + (T-1)\mu^\top \varphi_T + (T-1)\varphi_T^\top \mu + \varphi_T^\top \varphi_T \tag{40}$$

With this, Equation 39 can be written as

$$
\begin{aligned}
Q_T^{-1} &= (P_{T-1}^{-1} - (2T-1)\mu^\top \mu + 2\mu^\top \mu) - 2\mu^\top \mu + \varphi_T^\top \varphi_T - \frac{2T-1}{T^2} \left[(-2T+1)\mu^\top \mu + (T-1)\mu^\top \varphi_T + (T-1)\varphi_T^\top \mu + \varphi_T^\top \varphi_T\right] && (41) \\
&= (P_{T-1}^{-1} - (2(T-1)-1)\mu^\top \mu) - 2\mu^\top \mu + \varphi_T^\top \varphi_T + \frac{2T-1}{T^2} \left[(-2T+1)\mu^\top \mu + (T-1)\mu^\top \varphi_T + (T-1)\varphi_T^\top \mu + \varphi_T^\top \varphi_T\right] && (42) \\
&= Q_{T-1}^{-1} - 2\mu^\top \mu + \varphi_T^\top \varphi_T - \frac{2T-1}{T^2} \left[(-2T+1)\mu^\top \mu + (T-1)\mu^\top \varphi_T + (T-1)\varphi_T^\top \mu + \varphi_T^\top \varphi_T\right] && (43)
\end{aligned}
$$

One final expansion:

$$
Q_T^{-1} = Q_{T-1}^{-1} + \frac{1}{T^2} \left[((2T+1)^2 - 2T^2)\mu^\top \mu - (2T-1)(T-1)\mu^\top \varphi_T - (2T-1)(T-1)\varphi_T^\top \mu + (T^2 - 2T + 1)\varphi_T^\top \varphi_T\right] \quad (44)
$$

And now we can see that this can be written as

$$
Q_T^{-1} = Q_{T-1}^{-1} + \frac{1}{T^2} \begin{bmatrix} \mu_x(T-1)^\top & \varphi_T^\top \end{bmatrix} \begin{bmatrix} (2T-1)^2 - 2T^2 & -(2T-1)(T-1) \\ -(2T-1)(T-1) & (T-1)^2 \end{bmatrix} \begin{bmatrix} \mu_x(T-1) \\ \varphi_T \end{bmatrix} \quad (45)
$$

Let us define

$$
\begin{aligned}
C_T &:= \frac{1}{T^2} \begin{bmatrix} (2T-1)^2 - 2T^2 & -(2T-1)(T-1) \\ -(2T-1)(T-1) & (T-1)^2 \end{bmatrix} && (46) \\
V_T &:= \begin{bmatrix} \mu_x(T-1) \\ \varphi_T \end{bmatrix} && (47) \\
R_T &:= V_T^\top C_T V_T && (48)
\end{aligned}
$$

So that

$$
Q_T^{-1} = Q_{T-1}^{-1} + R_T \quad (49)
$$

The Woodbury matrix identity (Equation 17) gives us, at the end of all this, an update rule for Q_T:

$$
Q_T = Q_{T-1} - Q_{T-1} V_T^\top \left(C_T^{-1} + V_T Q_{T-1} V_T^\top\right)^{-1} V_T Q_{T-1} \quad (50)
$$

Note that this is a rank-2 update to Q_{T-1}, since we are using both the sample mean at time $T-1$ and the new data at time T to compute the update.

4.2.2 Deriving the $\hat{\Theta}$ Update

Returning to the normal equation

$$
\begin{aligned}
\hat{\Theta}_{LS}(T) &= Q_T(X_T - \bar{1}\mu_x(T))^\top Y && (51) \\
&= Q_T X^\top Y - Q_T \mu_x(T)^\top \bar{1}^\top Y && (52) \\
&= Q_T \left[Q_{T-1}^{-1}(t-1)\hat{\Theta}_{LS}(T-1) + \varphi_T^\top y_T\right] - T Q_T \mu_x(T)^\top \mu_y(T) && (53) \\
&= Q_T \varphi_T^\top y_T + Q_T \left[Q_T^{-1} - R_T\right] \hat{\Theta}_{LS}(T-1) - T Q_T \mu_x(T)^\top \mu_y(T) && (54) \\
&= \hat{\Theta}_{LS}(T-1) + Q_T \left[\varphi_T^\top y_T - R_T \hat{\Theta}_{LS}(T-1)\right] - T Q_T \mu_x(T)^\top \mu_y(T) && (55)
\end{aligned}
$$

Since in the current case we are assuming that Y is already centered, $\mu_y(T) = \bar{0}$ and the above reduces to

$$
\hat{\Theta}_{LS}(T) = \hat{\Theta}_{LS}(T-1) + Q_T \left[\varphi_T^\top y_T - R_T \hat{\Theta}_{LS}(T-1)\right] \quad (56)
$$

This corresponds roughly to the $\hat{\Theta}$ update in the centered case (Equation 16).

4.3 Uncentered X, Uncentered Y

We have finally arrived at the general case, in which our problem is expressed as

$$(y_T - \mu_y(T)) = (\varphi_T - \mu_x(T))\hat{\Theta}(T) \tag{57}$$

The normal equation for this case is

$$\begin{aligned}
\left[(X_t - \bar{1}\mu_x(T))^\top (X_T - \bar{1}\mu_x(T))\right]^{-1} & (X_T - \bar{1}\mu_x(T))^\top (Y_T - \bar{1}\mu_y(T)) \\
&= Q_T (X_T - \bar{1}\mu_x(T))^\top (Y_T - \bar{1}\mu_y(T)) \qquad (58) \\
&= Q_T (X_T - \bar{1}\mu_x(T))^\top Y - Q_T (X_T - \bar{1}\mu_x(T))^\top \bar{1}\mu_y(T) \qquad (59)
\end{aligned}$$

Note that the first term in Equation 59 is the normal equation for the uncentered X, centered Y case previously analyzed. Thus all that we need to do in order to derive the update for $\hat{\Theta}$ is expand the second term. This is easy, though, because

$$\begin{aligned}
Q_T (X_T - \bar{1}\mu_x(T))^\top \bar{1}\mu_y(T) &= Q_T X_T^\top \bar{1}\mu_y(T) - Q_T \mu_x(T)^\top \bar{1}^\top \bar{1}\mu_y(T) && (60) \\
&= T Q_T \mu_x(T)^\top \mu_y(T) - Q_T \mu_x(T)^\top \mu_y(T) && (61) \\
&= (T-1) Q_T \mu_x(T)^\top \mu_y(T) && (62)
\end{aligned}$$

This, combined with the correction term in Equation 55 (which is now nonzero since we assume that $\mu_Y(T) \neq 0$), gives us a total correction of $(2T-1)Q_T \mu_x(T)^\top \mu_y(T)$.

Since the update equation for Q_T depends only on φ_T and $\mu_x(T)$, it only remains to give the final update equation for $\hat{\Theta}(T)$ in the uncentered X, uncentered Y case. Combining Equation 56 with the previously stated correction term, we get the following update:

$$\begin{aligned}
\hat{\Theta}_{RAW}(T) &= \hat{\Theta}_{RAW}(T-1) + Q_T \left[\varphi_T^\top y_T - R_T \hat{\Theta}_{RAW}(T-1)\right] && (63) \\
\hat{\Theta}_{LS}(T) &= \hat{\Theta}_{RAW}(T) - (2T-1) Q_T \mu_x(T)^\top \mu_y(T) && (64)
\end{aligned}$$

Recall that our original problem in the uncentered case was

$$(y_T - \mu_y(T)) = (\varphi_T - \mu_x(T))\hat{\Theta} \tag{65}$$

Thus the prediction of the recursive least squares filter in the uncentered X, uncentered Y case for a new φ' is given by

$$\begin{aligned}
(\hat{y}' - \mu_y(T)) &= (\varphi' - \mu_x(T))\hat{\Theta}_{LS}(T) && (66) \\
\implies \hat{y}' &= \varphi' \hat{\Theta}_{LS}(T) + (\mu_y(T) - \mu_x(T)\hat{\Theta}_{LS}(T)) && (67)
\end{aligned}$$

5 Practical Extensions

5.1 Why Can't We Just Modify the Feature Vector?

In some data science and machine learning circles, a common ad hoc way to avoid centering data prior to solving the least squares problem is to append a constant dimension to the feature vector, such that the new feature vectors are given by

$$\lambda_t' = \begin{bmatrix} \lambda_t & 1 \end{bmatrix} \tag{68}$$

Intuitively, the intention of this technique is to add an additional offset parameter θ_0 to the least squares estimated parameters, which should then capture the constant offset term $\mu_y - \mu_x \hat{\Theta}$ which we calculated analytically above. In practice, though, this technique gives rise to an entire one-dimensional subspace of possible solutions to the least-squares problem, as can be shown using a bit of linear algebra, where the actual solution returned by solving the normal equations is determined by the particular numerical algorithm used to invert the covariance matrix. The addition of a constant feature to all input data makes the sample covariance matrix $X_T^\top X_T$ low-rank, so that the inverse $(X_T^\top X_T)^{-1}$ is ill-posed and gives rise to a subspace of possible solutions. The best solution of this subspace in expectation is precisely the one derived in Section 4, but the feature vector augmentation technique gives no guarantees of recovering this solution.

5.2 Initializing P_T and Q_T and Connections to Ridge Regression

The update equations derived in Sections 3 and 4 tell use how to move from the least-squares solution at timestep $T-1$ to the least-squares solution at timestep T, but they don't tell us how the relevant matrices should be initialized. For all involved matrices except P_T (or Q_T, in the uncentered case), it is reasonable to initialize with matrices whose elements are all 0. If we initialize P_T or Q_T to zero matrices, however, Equations 18 and 50 show that these matrices will never be updated at all (since the update equations are multiplicative in P_T and Q_T, respectively).

This means that we need to initialize P_T and Q_T to some nonzero matrix before beginning the recursive least squares algorithm. In practice, this initialization matrix is usually chosen as some multiple of the identity, so that $P_0 = \alpha I$. This has a significant effect on the computed $\hat{\Theta}$, however, as can be seen if we examine the real normal equation in this situation:

$$\hat{\Theta}_{REAL}(T) = (X_T^\top X_T + \frac{1}{\alpha} I)^{-1} X_T^\top Y \tag{69}$$

This is precisely the solution for the ridge regression problem (also known as l^2 regularized least squares or Tikhonov regularization) with regularization coefficient $\frac{1}{\alpha}$. Thus any practical implementation of recursive least squares which keeps around an estimate of the inverse covariance matrix (P_T or Q_T in our notation) is in fact computing a ridge regression estimator. This explains the common advice to use $\alpha \approx 10^6$; a large value for α corresponds to a low amount of l^2 regularization, and hence closer approximation to the unregularized least squares solution.

5.3 Forgetting Factors for Time-Varying Systems

When the relationship between φ_t and y_t is assumed to change over time, we need some way of prioritizing recent data over historical data in recursive least squares. This is commonly done via a "forgetting factor" $\lambda \in [0, 1]$. λ is used to give exponentially smaller weight to older samples in the regression in a way that can be intuitively explained by its extremal values: when $\lambda = 0$ no datum prior to the current timestep is taken into account, and when $\lambda = 1$ we recover the recursive least squares algorithm derived above. Common values of λ lie between 0.95 and 0.99.

The way that this is practically done is by reweighting the rows of the data matrix X and target matrix Y. Where previously these were defined as simply the vertically stacked samples, we now define them as

$$X = \begin{bmatrix} \varphi_T \\ \lambda\varphi_{T-1} \\ \vdots \\ \lambda^{T-2}\varphi_2 \\ \lambda^{T-1}\varphi_1 \end{bmatrix} \qquad Y = \begin{bmatrix} y_T \\ \lambda y_{T-1} \\ \vdots \\ \lambda^{T-2} y_2 \\ \lambda^{T-1} y_1 \end{bmatrix} \tag{70}$$

Similarly we redefine $\mu_x(T)$ and $\mu_y(T)$ to be the means of these new matrices:

$$\mu_x(T) = \frac{1}{T}\sum_{t=1}^{T} \lambda^{T-t}\varphi_t \qquad \mu_y(T) = \frac{1}{T}\sum_{t=1}^{T} \lambda^{T-t} y_t \tag{71}$$

By carrying this new X and Y through the same derivation as in the uncentered X, uncentered Y case above, we can derive analogous matrices and update rules:

$$C_T := \frac{1}{\lambda^2 T^2} \begin{bmatrix} (2T-1)^2 - 2T^2 & -(2T-1)(T-1) \\ -(2T-1)(T-1) & (T-1)^2 \end{bmatrix} \qquad V_T := \begin{bmatrix} \lambda\mu_x(T-1) \\ \varphi_T \end{bmatrix} \tag{72}$$

$$R_T := V_T^\top C_T V_T \tag{73}$$

$$Q_T = \frac{1}{\lambda^2}\left[Q_{T-1} - Q_{T-1}V_T^\top \left(C_T^{-1} + V_T Q_{T-1} V_T^\top\right)^{-1} V_T Q_{T-1}\right] \tag{74}$$

$$\hat{\Theta}_{RAW}(T) = \hat{\Theta}_{RAW}(T-1) + Q_T\left[\varphi_T^\top y_T - R_T \hat{\Theta}_{RAW}(T-1)\right] \tag{75}$$

$$\hat{\Theta}_{LS}(T) = \hat{\Theta}_{RAW}(T) - (2T-1)Q_T \mu_x(T)^\top \mu_y(T) \tag{76}$$

One final note is in order about the recursive least squares algorithm with λ-forgetting: The connection that we drew earlier between the initialization of Q_T and l^2 regularized least squares no longer holds, as the initial setting of Q_T is multiplied by λ^2 at each timestep, and so the equivalent regularization coefficient gets smaller with each new datapoint. More precisely, the recursive least squares solution with λ-forgetting and an initialization of $Q_0 = \alpha I$ will, at time T, compute the equivalent ridge regression solution

$$\hat{\Theta}_{REAL} = (X_T^\top X_T + \frac{1}{\alpha \lambda^{2T}})^{-1} X_T^\top Y \tag{77}$$

This may be desirable if one wishes to smoothly interpolate between the ridge regression solution when little data is available and the unregularized least squares solution in the limit of infinite data, but we are not aware of any existing statistical analysis of this interpolation.

References

[1] Christopher M Bishop. *Pattern recognition and machine learning.* springer, 2006.

[2] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[3] Max A Woodbury. Inverting modified matrices. 1950.