The C-BLP: Constrained Linear Predictors

June 22, 2023

The Goal

Consider a second order random process X, such that at each value of $t \in \mathbb{R}$, we have a random variable X_t.

We may randomly sample this vector at n points, gaining a vector $\vec{T} = (t_1, t_2, t_3, \ldots)$ of times at which the samples were made, and $\vec{X} = (X_{t_i})$. Strictly speaking these are both random variables in and of themselves, up until the moment that we 'realise' them. We can index into these vectors using the the integer $0 \leq i < n$, and we assume without loss of generality that the samples are sorted in time, such that $t_i < t_{i+1} \forall i$.

We wish to find a predictor, $\hat{X}_t$, which will predict the values of X_t on a set of 'prediction points', $t \in T$, subject to three further conditions:

- We are willing to present an *a priori* guess at the functional form of the predictor, in the form of a 'prior function' $g(t)$.
- The only thing we 'know' (or are willing to *ansatz*) about X_t is the second moment kernel (a generalisation of the covariance):
$$\langle (X_t - g(t))(X_s - g(s)) \rangle = k(t, s)$$
- Our predictor should be linear, such that:
$$\hat{X}_t = g(t) + \vec{a}_t \cdot \left(\vec{X} - \vec{G} \right)$$
Where $G_i = g(t_i)$

We again reiterate that $X_t, \vec{X}$ and $\hat{X}_t$ are - strictly speaking - random variables until we make them into real numbers at the moment we wish to actually make a prediction. $\vec{a}_t$ is a real n-tuple, which takes on different values at each value of t.

These are the ingredients of the standard BLP. The goal of this work is to extend this by adding the knowledge that the underlying process – and hence the predictions – should obey a number of constraints.

1 Deriving the C-BLP

We define the C-BLP as the linear predictor which minimises the Mean Squared Error, averaged across all realisations of the random variable, computed at the set T of points at which we wish to make predictions, and which obeys our constraints.

Therefore, the C-BLP minimises the following Lagrangian:

$$\mathcal{L} = \sum_{t \in T} \langle (X_t - \hat{X}_t)^2 \rangle - \sum_j \lambda_j h_j(\{\hat{X}\}) \tag{1}$$

Here $h_j(\{\hat{X}_t\})$ is the j^{th} constraint on the *prediction points*[1], such that $h_j = 0$ when the constraint is met, and is non-zero otherwise, with the sum running over all such constraints. $\lambda_j \in \mathbb{R}$ are the associated Lagrange Multipliers. In the standard BLP we are able to treat the Lagrangian as separable in each element of T - minimising the MSE individually at each $t \in T$ is equivalent to performing a global minimisation: in the C-BLP this is not true, and we must consider the global case.

The issue at present is that we do not know what the behaviour of X_t is – we might have an initial guess (i.e. our prior, $g(t)$), but the entire purpose of this exercise is that we do not know X_t. However, by expanding out the brackets, we are able to write the Lagrangian in the following form:

$$\begin{aligned} \mathcal{L} &= \left[\sum_{t \in T} \left\langle {X'_t}^2 \right\rangle - 2\vec{a}_t \cdot \left\langle X'_t \vec{X}' \right\rangle + \left\langle (\vec{a}_t \cdot \vec{X}')^2 \right\rangle \right] - \sum_j \lambda_j h_j(\{\hat{X}\}) \\ &= \left[\sum_{t \in T} \left\langle {X'_t}^2 \right\rangle - 2\vec{a}_t \cdot \vec{k}_t + \vec{a}_t \cdot (K\vec{a}_t) \right] - \sum_j \lambda_j h_j(\{\hat{X}\}) \end{aligned} \tag{2}$$

Where:

$$\begin{aligned} X'_t &= X_t - g(t) \\ \vec{X}' &= \vec{X} - \vec{G} \\ \vec{k}_t &\in \mathbb{R}^n \text{ such that } \left[\vec{k}_t\right]_i = k(t, t_i) \\ K &\in \mathbb{R}^{n \times n} \text{ such that } K_{ij} = k(t_i, t_j) \end{aligned} \tag{3}$$

Note that since the kernel is, by definition, symmetric in its arguments, $K^T = K$. Note that we have also taken the explicit step of writing our kernel as a relationship between the *transformed* data - i.e. X' - the imposition of different functions $g(t)$ might therefore warrant different kernels. This is true even if the transform is the (commonly used) constant 'mean scaling', $g(t) = \langle X_t \rangle \approx \frac{1}{n} \vec{X} \cdot \mathbb{1}$.

By performing this transform we have placed the incomputable terms - that of $\left\langle (X'_t)^2 \right\rangle$ into a constant term. Since Lagrangians are invariant under constant scalings, it is possible to find an optimal value of $\vec{a}_t$ using only the remaining computable terms.

However - as we shall see - we are in the uncomfortable position of trying to impose conditions on the predicted values, $P_i = \hat{X}_{t_i} = g(t_i) + \vec{a}_{t_i} \cdot \vec{X}'$ whilst our object of interest is now the vector $\vec{a}_{t_i}$.

We therefore limit ourselves to the case of *linear constraints*, i.e., those which can be written in the following

[1] For clarity and avoidance of symbol-collision with the other X-s, we will denote the prediction points as $P_i = \hat{X}_{t_i} = g(t_i) + \vec{a}_t \cdot \vec{X}'$

form:

$$\begin{aligned} h_j(\{P\}) &= c_j - \sum_k d_{jk} P_k \\ &= c_j - \sum_k d_{jk} \left(g(t_k) + \vec{a}_{t_k} \cdot \vec{X}' \right) \end{aligned} \tag{4}$$

We can then take the derivative of the Lagrangian with respect to $\vec{a}_{t_i}$, and find that:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \vec{a}_{t_i}} &= 2K\vec{a}_{t_i} - 2\vec{k}_i - \sum_j \lambda_j \frac{\partial h_j}{\partial \vec{a}_{t_i}} \\ &= 2K\vec{a}_{t_i} - 2\vec{k}_i + \left(\sum_j \lambda_j b_{ji} \right) \vec{X}' \\ &= 2K\vec{a}_{t_i} - 2\vec{k}_i + \eta_i \vec{X}' \end{aligned} \tag{5}$$

Hence, the optimal value of $\vec{a}_{t_i}$ is:

$$\begin{aligned} \vec{a}_{t_i} &= K^{-1} \left(\vec{k}_i - \frac{\eta_i}{2} \vec{X}' \right) \\ &= \vec{v}_i - \frac{\eta_i}{2} \vec{w} \end{aligned} \tag{6}$$

The optimal predicted value is:

$$\begin{aligned} P_i &= g(t_i) + \vec{a}_{t_i} \cdot \vec{X}' \\ &= g(t_i) + \vec{v}_i \cdot \vec{X}' - \frac{\eta_i}{2} \vec{w} \cdot \vec{X}' \\ &= g(t_i) + A_i - \frac{\eta_i}{2} B \end{aligned} \tag{7}$$

1.1 Exact Constraints

In the case where the constraints h_j are exact – i.e. the sets $\{c\}$ and $\{d\}$ are exactly determined, we may therefore analytically solve to find the set of Lagrange multipliers, then $\vec{\eta}$, and hence compute the predictor. We note that $\vec{\eta}$ can be written as:

$$\vec{\eta} = D^T \vec{\lambda} \tag{8}$$

Where $D_{ij} = d_{ij}$ is the constraint matrix, $\vec{\eta}_k = \eta_k$ is a vector on $\mathbb{R}^N$ and $\vec{\lambda}_k = \lambda_k$ is a vector on $\mathbb{R}^m$, where m is the number of constraints. The requirement that the constraints are met can be written as:

$$D\vec{p} = \vec{c} \tag{9}$$

Where $\vec{p}_i = P_i$ is another vector on $\mathbb{R}^n$ and $\vec{c}_i = c_i \in \mathbb{R}^m$. Writing $g(t_i) + A_i = q_i$, this is then:

$$D\left(\vec{q} - \frac{B}{2} D^T \vec{\lambda} \right) = \vec{c} \iff \vec{\lambda} = \frac{2}{B} \left(DD^T \right)^{-1} \left(D\vec{q} - \vec{c} \right) \tag{10}$$

Therefore:

$$\vec{p} = \left(\mathbb{1}_N - D^T(DD^T)^{-1}D\right)\vec{q} + D^T(DD^T)^{-1}\vec{c} \tag{11}$$

In the case where there is only a single constraint ($m = 1$), this simplifies such that $D \to \vec{d}^T$:

$$\vec{p} = \vec{q} + \frac{c - \vec{q}\cdot\vec{d}}{\vec{d}^2}\vec{d} \tag{12}$$

1.2 Inexact Constraints

In the case where the constraints are not exact, but serve to enforce bounds – i.e. monotonicity or positivity – there is a problem since the parameters of the constraint are not fixed. We may not care, for example, how much greater X_{i+1} is than X_i is, only that it *is* greater.

We could enforce this through slack variables and utilise the KKT conditions, however for our purposes it is better to *parameterise* the constraint.

Various parameterisations are possible, but perhaps the most comprehensible is to consider that the *prediction* points, P_i are a function of some other parameters $\vec{\theta} \in R^m$, such that:

$$\begin{aligned} P_i &= \mathcal{T}_i(\vec{\theta}) \\ h_j(\mathcal{T}_i(\vec{\theta})) &= 0 \ \forall \ i, j, \vec{\theta} \end{aligned} \tag{13}$$

For example, in the case of enforcing positivity, we might have that $P_i = e^{z_i}$, which is equivalent to asserting that $d_{ij} = \delta_{ij}$ and $c_i = e^{z_i}$. Rearranging Eq. (7), we are able to write η_i as a function of this Transform, and hence write $\vec{a}_{t_i}$ in the following form:

$$\vec{a}_{t_i} = \vec{v}_i + \frac{P_i(\vec{\theta}) - A_i - g(t_i)}{B}\vec{w} \tag{14}$$

This might seem somewhat tautological - we have written $\vec{a}_{t_i}$ in terms of the prediction values - but the entire purpose of $\vec{a}_{t_i}$ is to make predictions!

The usefulness of this comes evident when we insert Eq. (14) back into the Lagrangian – essentially performing a change of coordinates from $\mathcal{L}(\vec{a}, \vec{\theta})$ to $\mathcal{L}(\vec{\theta})$, since we have now ensured that $\vec{a}_t$ will always be at its optimal value for each value of $\vec{\theta}$.

$$\vec{k}_i \cdot \vec{a}_{t_i} = \vec{v}_i \cdot \vec{k}_i + \frac{P_i(\vec{\theta}) - A_i - g(t_i)}{B}\vec{w}\cdot\vec{k}_i \tag{15}$$

$$\begin{aligned} \vec{a}_{t_i} \cdot (K\vec{a}_{t_i}) &= \left(\vec{v}_i + \frac{P_i(\vec{\theta}) - A_i - g(t_i)}{B}\vec{w}\right)\cdot\left(\vec{k}_i + \frac{P_i(\vec{\theta}) - A_i - g(t_i)}{B}\vec{X}'\right) \\ &= \vec{v}_i \cdot \vec{k}_i + \left(\frac{P_i(\vec{\theta}) - A_i - g(t_i)}{B}\right)\left(\vec{w}\cdot\vec{k}_i + A_i\right) + \frac{\left(P_i(\vec{\theta}) - A_i - g(t_i)\right)^2}{B} \end{aligned} \tag{16}$$

Since $\vec{w}\cdot\vec{k}_i = (K^{-1}\vec{X}')\vec{k}_i = (K^{-1}\vec{k}_i)\vec{X}_{\prime} = \vec{v}_i \cdot \vec{X}' = A_i$ due to the symmetry of K, and the constraints are all automatically satisfied thanks to our parameterisation, we find that the Lagrangian simplifies to:

$$\begin{aligned}
\mathcal{L}(\vec{\theta}) &= \sum_i \left(\left\langle (X_i')^2 \right\rangle - \vec{k}_i \cdot \vec{v}_i \right) + \frac{1}{B} \left(P_i(\theta) - A_i - g(t_i) \right)^2 \\
&= \text{const in } \vec{\theta} + \frac{1}{B} \sum_i \left(P_i(\theta) - A_i - g(t_i) \right)^2 \qquad (17) \\
\mathcal{L}' &= \sum_i P_i \left(P_i(\theta) - 2(A_i + g(t_i)) \right)
\end{aligned}$$

Where in the final line we took the opportunity to perform a rescaling (recalling that $B > 0$ is enforced by the positive definiteness of K) which leaves the optimum invariant. In some cases it is trivial to identify the optimal values of P_i - for example, in the case where $P_i = e^{\theta_i}$, the maximum is evidently:

$$P_i = \begin{cases} A_i + g(t_i) & \text{if this is } > 0 \\ 0 & \text{else} \end{cases} \tag{18}$$

In short, the C-BLP is equal to the BLP except when the condition is violated, at which point a hard cut is placed on it.

More complex conditions however, can lead to more complex behaviour - the monotonicity constraint, for example, exhibits the obvious behaviour that it again follows the BLP when it is monotonic, and is flat when the BLP has a negative gradient – but the *location* where the C-BLP becomes flat is non-trivial, with flatness necessarily occuring *before* the BLP changes direction: a tradeoff in following the BLP locally versus becoming too large too early without the ability to decrease due to the monotonic constraint.

In these cases a more complex search is required – where the behaviour of the constraint is evident *a priori* (such as the monotonic constraint), one can limit the space of the search. In the general case, however, a numerical optimisation is required.

The derivative of the Lagrangian with respect to the constraint parameters is:

$$\frac{\partial \mathcal{L}'}{\partial \theta_m} = 2 \sum_i \left(P_i - A_i - g(t_i) \right) \frac{\partial P_i}{\partial \theta_m} \tag{19}$$

This can be used to numerically optimise the values of $\vec{\theta}$

1.3 Inexact Constraints (Redux)

We note that we performed a fairly drastic change in approach between the exact constraints and the inexact constraints – is it possible to maintain the same approach for both?

We consider now that the parameters $\vec{c}$ of the constraints are functions of an (unconstrained) external parameter, $\vec{z} \in \mathbb{R}^m$ - letting $\vec{c} = $ const recovers the condition of the exact equalities. However, in any other case we must still find the values of $\vec{z}$ which optimise the global Lagrangian - and hence we need to rewrite our Lagrangrian in terms of $\vec{c}$.

From Eq. (11), we can rewrite the predicted value-vector (recalling that $\vec{p}_i = P_i = \hat{X}_{t_i}$) as:

$$\begin{aligned} \vec{p} &= \vec{j} + R\vec{c}(\vec{z}) \\ R &= D^T(DD^T)^{-1} \\ \vec{j} = (\mathbb{1}_N - RD)\,\vec{q} \iff & \quad j_i = g(t_i) + A_i + \sum_{j,k} R_{ij} D_{jk}(g(t_k) + A_k) \end{aligned} \tag{20}$$

We note that from a conceptual standpoint it is not a problem for the 'mixing' constraints D_{ij} to be the functions of $\vec{z}$, but this assumption allows us to precompute many of the otherwise troublesome entities. We can also rewrite $\vec{a}_{t_i}$ as:

$$\begin{aligned} \vec{a}_{t_i} &= \vec{v}_i - \frac{\eta_i}{2}\vec{w} \\ &= \vec{v}_i + \frac{[R(\vec{c} - D\vec{q})] \cdot \hat{e}_i}{B}\vec{w} \\ &= \vec{j}_i + \frac{(R\vec{c}) \cdot \hat{e}_i}{B}\vec{w} \end{aligned} \tag{21}$$

Where

$$\begin{aligned} R &= D^T(DD^T)^{-1} \\ \vec{j}_i &= \vec{v}_i - \frac{(RD\vec{q}) \cdot \hat{e}_i}{B}\vec{w} \end{aligned} \tag{22}$$

We therefore have:

$$\begin{aligned} \vec{k}_i \cdot \vec{a}_{t_i} &= \vec{v}_i \cdot \vec{k}_i + \frac{A_i}{B}\left((R\vec{c}) \cdot \hat{e}_i - (RD\vec{q}) \cdot \hat{e}_i\right) \\ &= \text{const in } \vec{c} + \frac{A_i}{B}(R\vec{c}) \cdot \hat{e}_i \\ \vec{a}_{t_i} \cdot (K\vec{a}_{t_i}) &= \left(\vec{j}_i + \frac{(R\vec{c}) \cdot \hat{e}_i}{B}\vec{w}\right) \cdot \left(K\vec{j}_i + \frac{(R\vec{c}) \cdot \hat{e}_i}{B}\vec{X}\right) \\ &= \text{const in } \vec{c} + 2\frac{(R\vec{c}) \cdot \hat{e}_i}{B}\vec{j}_i \cdot \vec{X} + \frac{1}{B}((R\vec{c}) \cdot \hat{e}_i)^2 \\ &= \text{const in } \vec{c} + 2\frac{(R\vec{c}) \cdot \hat{e}_i}{B}\left(A_i - (RD\vec{q}) \cdot \hat{e}_i\right) + \frac{1}{B}((R\vec{c}) \cdot \hat{e}_i)^2 \end{aligned} \tag{23}$$

Therefore:

$$\begin{aligned} \mathcal{L}' &= \sum_i \vec{a}_{t_i} \cdot K\vec{a}_{t_i} - 2\vec{k}_i \cdot \vec{a}_{t_i} \\ &= \text{const in } \vec{c} + \sum_i ((R\vec{c}) \cdot \hat{e}_i)^2 - 2(R\vec{c}) \cdot \hat{e}_i(RD\vec{q}) \cdot \hat{e}_i \\ &= \text{const in } \vec{c} + (R\vec{c})^2 - 2(R\vec{c}) \cdot (RD\vec{q}) \\ &= \text{const in } \vec{c} + \left(R\vec{c}(\vec{z}) - RD\vec{q}\right)^2 \end{aligned} \tag{24}$$

The derivative with respect to the (unconstrained) vectors $\vec{z}$ is:

$$\begin{aligned}\frac{\partial \mathcal{L}'}{\partial z_m} &= (R\vec{c}(\vec{z}) - RD\vec{q}) \cdot R\frac{\partial \vec{c}}{\partial z_m} \\ &= (\vec{p}(\vec{z}) - \vec{q}) \cdot R\frac{\partial \vec{c}}{\partial z_m}\end{aligned} \tag{25}$$

Since $\vec{q}$ is the BLP prediction we can once again see that the derivative is zero if the BLP obeys the constraints ($\vec{c} - D\vec{q} = 0$), so the C-BLP will always revert to the BLP if this meets our constraints.