Dual Interpretation of Machine Learning Forecasts

Philippe Goulet Coulombe
Université du Québec à Montréal

Maximilian Göbel
Lingua Franca Economics

Karin Klieber
European Central Bank

FinEML & BdF AI Methods, Oct 2025

An Ensemble of Recent and Related Papers

I. **TODAY: Interpret ML models macroeconomic *forecasts* through duality, i.e., portfolio weights on the target variable**

   "Dual Interpretation of Machine Learning Forecasts," Goulet Coulombe, Göbel, and Klieber, 2024. https://arxiv.org/abs/2412.13076.

II. **Interpreting *dynamic causal effects* obtained from differences of conditional expectations**

   "Opening the Black Box of Local Projections," Goulet Coulombe and Klieber, 2025. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5237376

III. **Leveraging the proximity-based interpretation of OLS to link it to attention modules in large language models**

   "OLS as an Attention Mechanism," Goulet Coulombe, 2025. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5200864

Motivation

- A conditional mean f is typically interpreted through $\frac{\partial f(\boldsymbol{X})}{\partial X}$. In linear models, this is $\boldsymbol{\beta}$.

- **Problem:** In even linear regressions, the partial derivative of the predictand with respect to a predictor becomes nearly meaningless in a high-dimensional setup.
  - How to interpret a system with 150 cross-correlated variables?
  - What thought experiment does it correspond to? What does "ceteris paribus" mean?

- Using nonlinear methods, which effectively expand the feature space, makes things worse.

- **Some known solutions:** High-dimensionality challenges are often addressed through reinstating sparsity in the *covariate space*.
  - Regularization: In macro forecasting, sparsity is often empirically shaky and/or implausible.
  - Factor models: Interpretation of latent factors can be tricky.

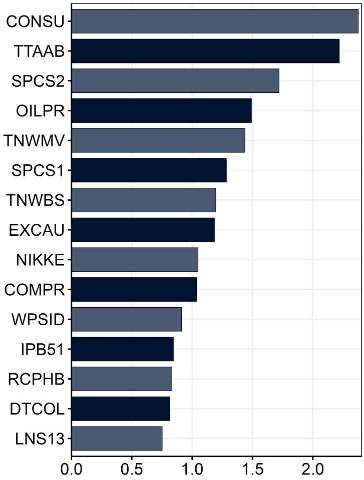
New Avenues?

- **Takeaway:** Some kind of sparsity is key, and linearity is preferable.
- **Our proposition:** Interpreting the model via the *time series dimension*.
  - In macroeconomic forecasting, the number of training observations does not require further sparsification efforts.
  - After some gentle feature engineering (lags, moving average, etc), we often face $P >> N$. Thus, maybe N is more manageable.
  - Many ML methods that are nonlinear in $\boldsymbol{X}$ are linear in $\boldsymbol{y}$.

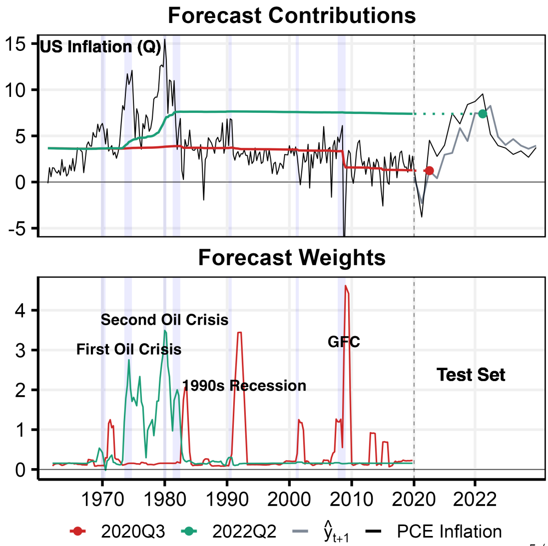
Duality

An out-of-sample prediction can be decomposed in not one, but *two* ways

(a) Primal: $\hat{y}_j = X_j'\hat{\boldsymbol{\beta}}, \;\; \boldsymbol{\beta} \in \mathbb{R}^P$



(b) Dual: $\hat{y}_j = \hat{\boldsymbol{w}}_j \boldsymbol{y}, \;\; \boldsymbol{w} \in \mathbb{R}^N$



Contribution

- We propose a complementary dual interpretation of forecasts, where the sparse and ordered nature of macroeconomic data becomes advantageous.
- We show **how to obtain w** for (Kernel) Ridge Regression, Neural Networks, and tree ensembles – requiring little to no additional computations beyond the estimation of the original model.
- We show **how to interpret w**, i.e., as portfolio weights quantifying pairwise observation proximity, as perceived by the machine learning model.
- Empirical illustrations include forecasting post-Pandemic inflation, GDP growth, and unemployment during the GFC.

- Ridge Regression (RR) coefficients are obtained via

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} (y_i - \boldsymbol{\beta}' X_i)^2 + \lambda ||\boldsymbol{\beta}||_2$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_P)^{-1} \boldsymbol{X}'\boldsymbol{y} \qquad \text{(Primal Solution)}$$

$$\hat{\boldsymbol{\beta}} = \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{X}' + \lambda \boldsymbol{I}_N)^{-1} \boldsymbol{y} \qquad \text{(Dual Solution)}$$

- Prediction for an out-of-sample observation j is obtained via

$$\hat{y}_j = X_j \hat{\boldsymbol{\beta}} = X_j (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_P)^{-1} \boldsymbol{X}'\boldsymbol{y} \qquad \text{(Covariances-Based Prediction)}$$

$$\hat{y}_j = K_j \hat{\boldsymbol{\alpha}} = X_j \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{X}' + \lambda \boldsymbol{I}_N)^{-1} \boldsymbol{y} \qquad \text{(Proximity-Based Prediction)}$$

- Defining data portfolio weights as $\boldsymbol{w}_j \equiv X_j \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{X}' + \lambda \boldsymbol{I}_N)^{-1}$ results in

$$\hat{y}_j = \boldsymbol{w}_j \boldsymbol{y} \qquad \forall\ j \in \text{Test Sample}\,.$$

Some Intuition for the w_j Dual Formula

- We have that

$$\boldsymbol{w}_j = \underbrace{(X_j \boldsymbol{X}')}_{\text{Plain Proximities}} \times \underbrace{(\boldsymbol{X}\boldsymbol{X}' + \lambda \boldsymbol{I}_N)^{-1}}_{\text{Proximity Denominator}}.$$

- Setting $\lambda = 0$ for simplicity, the OLS solution is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$
$$\hat{y}_j = X_j\hat{\boldsymbol{\beta}} = X_j(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}.$$

- We can rewrite this using the eigendecomposition:

$$\hat{y}_j = F_j \boldsymbol{F}'\boldsymbol{y}$$

where

$$\boldsymbol{F} = \boldsymbol{X}\boldsymbol{U}\Lambda^{-1/2},$$
$$F_j = X_j\boldsymbol{U}\Lambda^{-1/2},$$
$$\boldsymbol{U}\Lambda\boldsymbol{U}' = \boldsymbol{X}'\boldsymbol{X}.$$

- We get an orthonormal representation of inputs: $\boldsymbol{F}'\boldsymbol{F} = \boldsymbol{I}_P$.

Some Intuition for the w_j Dual Formula
Inner Product and Cosine Similarity

- The prediction for a test observation j becomes:

$$\hat{y}_j = \sum_{i=1}^{N} \underbrace{\langle F_j, F_i \rangle}_{\equiv w_{ji}} y_i,$$

  where $\langle F_j, F_i \rangle$ is the inner product in the transformed space.

- We can further decompose the inner product:

$$\hat{y}_j = \sum_{i=1}^{N} \underbrace{\|F_j\| \|F_i\|}_{\text{scale}} \underbrace{\cos(\theta_{ji})}_{\text{alignment}} y_i$$

  with θ_{ji} the angle between F_j and F_i.

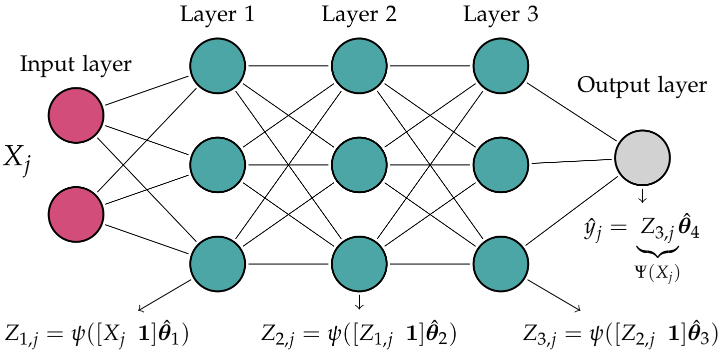
→ **Interpretation:**
  - OLS assigns higher weight to observations that are *similar* in the transformed feature space, à la nearest neighbors.
  - Large w_{ji}'s arise from both vector alignment and magnitude.

Kernel Ridge Regression

- Obtaining $\boldsymbol{w}_j$ is very straightforward, since KRR is already set up in the dual space.
- KRR induces nonlinearities in $\boldsymbol{X}$ by replacing inner products with kernel-based proximities ($\mathcal{K}$).
- This *implicitly* encode pairwise similarities as inner products in an expanded feature space: $Z_i = \Phi(X_i) \in \mathbb{R}^{\tilde{P}}$, with $\tilde{P} > P$.
- Prediction for y_j is then given by:

$$\begin{aligned} \hat{y}_j &= \mathcal{K}(X_j, \boldsymbol{X})(\mathcal{K}(\boldsymbol{X}, \boldsymbol{X}) + \lambda \boldsymbol{I}_N)^{-1} \boldsymbol{y} \\ &= \underbrace{K_j(\mathbf{K} + \lambda \boldsymbol{I}_N)^{-1}}_{\boldsymbol{w}_j} \boldsymbol{y} \,. \end{aligned}$$

- NN's prediction $\hat{y}_j$ is obtained recursively, moving from inputs X_j in the first layer $l = 1$ to *generated* features $Z_{L-1,j} \equiv \Psi(X_j)$ in the final layer L.
- Let ψ denote the activation function and $\boldsymbol{\theta}_l$ the network's parameters for layer l. For $L = 3$, we have:



Neural Networks
Backing out $\boldsymbol{w}_j$

- If the final layer is linear, $\boldsymbol{w}_j$ can be obtained as in Ridge Regression:

$$
\begin{aligned}
\hat{y}_j &= \Psi(X_j)\hat{\boldsymbol{\theta}}_L \\
&\cong \Psi(X_j)\left(\Psi(\boldsymbol{X})'\Psi(\boldsymbol{X}) + \lambda \boldsymbol{I}_{n_L}\right)^{-1}\Psi(\boldsymbol{X})'\boldsymbol{y} \\
&= \Psi(X_j)\Psi(\boldsymbol{X})'\left(\Psi(\boldsymbol{X})\Psi(\boldsymbol{X})' + \lambda \boldsymbol{I}_N\right)^{-1}\boldsymbol{y} \\
&= \underbrace{K_j(\mathbf{K} + \lambda \boldsymbol{I}_N)^{-1}}_{\boldsymbol{w}_j}\boldsymbol{y}
\end{aligned}
$$

- The trick is that, at the "optimum", first-order conditions for NN are equivalent to running a l_2-regularized *linear regression* using $\Psi(\boldsymbol{X})$ as *manually generated regressors*.

- One can reinterpret the NN solutions as one obtained by alternating two optimization steps: optimizing $\hat{\boldsymbol{\theta}}_L$ conditional on $\hat{\boldsymbol{\theta}}_{1:(L-1)}$, and then $\hat{\boldsymbol{\theta}}_{1:(L-1)}$ conditional on $\hat{\boldsymbol{\theta}}_L$.

- Random Forest uses B individual regression trees ($\mathcal{T}_b$) for its prediction:

$$\hat{y}_j = \frac{1}{B}\sum_{b=1}^{B}\mathcal{T}_b(X_j)$$

- RF's prediction is an average of local averages, and thus, a linear combination of $\boldsymbol{y}$:

$$\hat{y}_j = \frac{1}{B}\sum_{b=1}^{B}\mathcal{T}_b(X_j) = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{N} w_{bji} y_i = \sum_{i=1}^{N}\underbrace{\frac{1}{B}\sum_{b=1}^{B} w_{bji}}_{w_{ji}} y_i = \boldsymbol{w}_j\boldsymbol{y},$$

with $w_{bji} = \frac{I\left(i \in \mathcal{P}_b(X_j)\right)}{\sum_{i'=1}^{N} I\left(i' \in \mathcal{P}_b(X_j)\right)}$ and $\mathcal{P}_b$ being the partition implied by the tree.

- Thus, we can back out $\boldsymbol{w}_j$ through accounting operations on trees.
- Boosting is less trivial: we use the algorithm of Geertsema and Lu (2023).

Two Things to Visualize

- **Forecast Weights**: we can plot w_{ji} as a time series, possibly smoothed with a moving average.

- **Forecast Contributions:** we can plot

$$c_{ij} = w_{ji} y_i$$

  through a *cumulative sum* converging to $\hat{y}_j$ as we go from $i = 1$ to $i = N$.

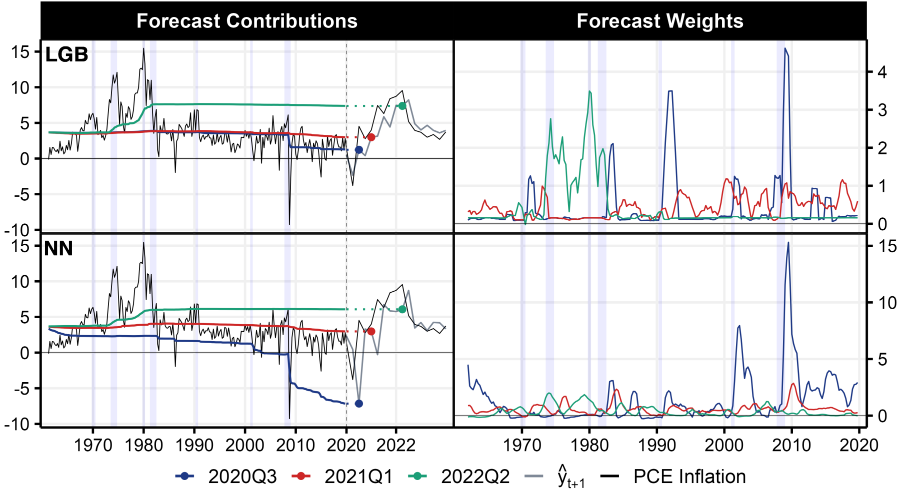
Empirical Application

- Quarterly data from FRED-QD (McCracken and Ng, 2020) with 245 macroeconomic and financial variables, $p = 4$ lags, moving averages of order 2, 4, and 8 (as in Goulet Coulombe et al., 2021). This results in $P = 1732$ regressors.
- Sample runs from 1961Q2 to 2024Q1 ($N = 252$).
- Direct forecasting for various macro variables and multiple horizons.

  Inflation: $h \in \{1\}$ for OOS 2020Q1-2024Q1
  GDP Growth: $h \in \{1, 2, 4\}$ for OOS 2020Q1-2025Q1

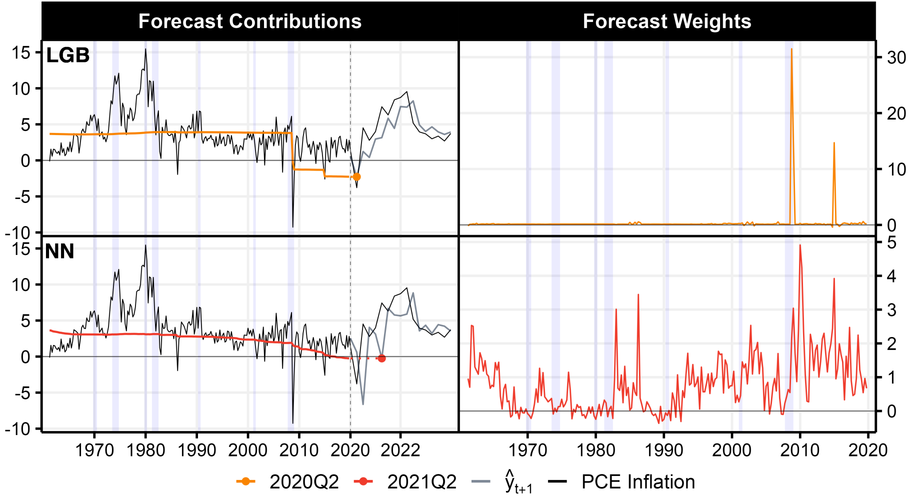
  $\rightarrow$ Training is based on 1961Q2-2019Q4.
- Inclusive set of models.
  1. Linear: FAAR, RR
  2. Kernel-based: KRR
  3. Tree-based: RF, LGB
  4. Deep learning: NN, HNN

Inflation for the Post-Pandemic Surge I



- For 2020Q3, NN severely underpredicts, partly due to overemphasizing the GFC.
- For 2021Q1, both models are slow to recognize parallels with the 1970s. LGB places most of the weight on pre-pandemic years, implying a return to normal.
- For 2022Q2, there is no ambiguity in LGB: it upweights both major inflation spikes from the 1970s.

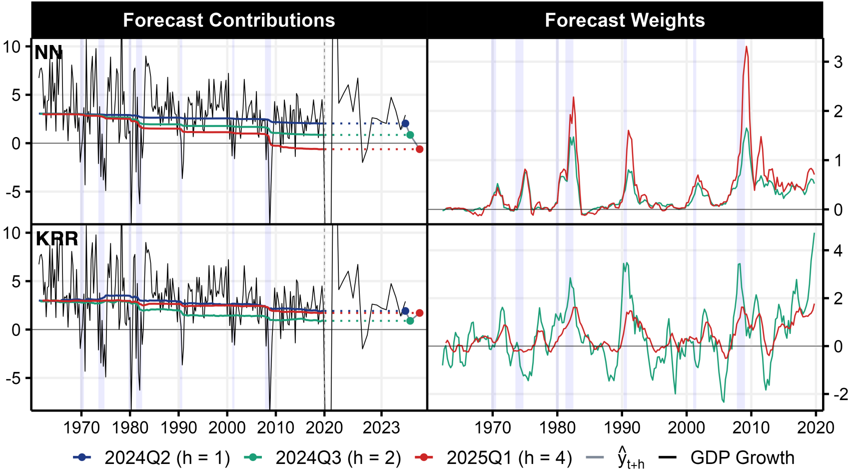
Inflation for the Post-Pandemic Surge II



- LGB's striking prediction for 2020Q2 relies on a highly sparse weighting scheme: 2008Q4, 2009Q1 (→ **GFC**) and 2015Q1 (→ **sovereign debt crisis** in Europe).
- NN's flaw in 2021Q2 is missing parallels with the 1970s high-inflation period.

Recent Predictions for GDP Growth

Are we facing a recession?



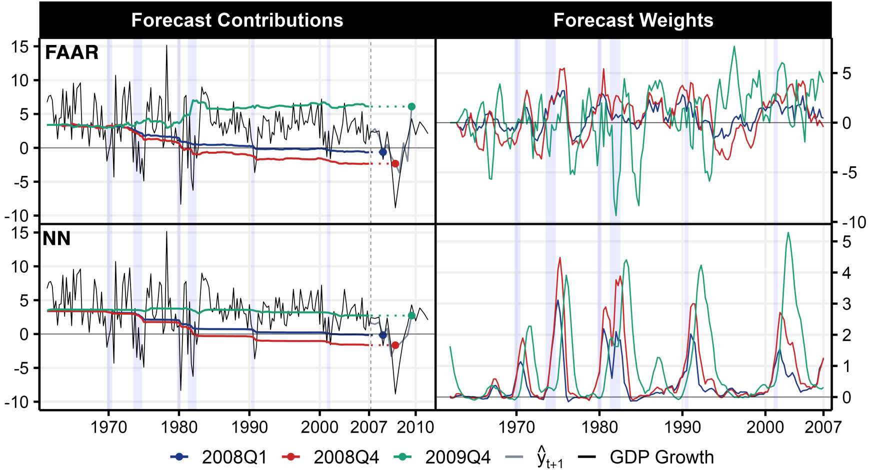
- Both NN and KRR see first signs of a slowdown for $h = 2$, with contributions mainly coming from the GFC, the second oil crisis and the 1990s recession.
- NN finds strong similarities with past recessions for $h = 4$, which is rather rare for longer horizons.

Parting Words

- The dual interpretation enables a narrative reading of otherwise opaque forecasts.
- One avenue for future work is to develop inference methods for w_j—e.g., to construct confidence bands.
- Another is to explore shrinkage or regularization on w_j to further enhance interpretability.

Appendix

GDP Growth for the Great Recession



	Concentration			Leverage			Short Position			Turnover
	2008Q1	2008Q4	2009Q4	2008Q1	2008Q4	2009Q4	2008Q1	2008Q4	2009Q4	Overall
FAAR	0.17	0.16	0.17	1.00	1.00	1.00	-0.93	-1.68	-3.51	99.81
NN	0.37	0.33	0.28	0.80	1.31	1.71	-0.03	-0.04	-0.04	6.81

Point Forecasting Performance

	FAAR	KRR	LGB	NN	RF	RR	HNN
Inflation ($h = 1$)							
2020Q1-2024Q1	4.30	0.90	0.80	1.50	0.98	1.60	1.36
2021Q1-2024Q1	1.82	0.97	0.99	1.35	1.04	1.45	0.90
GDP Growth ($h = 1$)							
2007Q2-2009Q4	0.63	1.11	0.90	0.63	0.78	0.85	–
2020Q1-2024Q2	1.32	0.95	0.89	0.98	0.88	0.95	–
2021Q1-2024Q2	0.96	0.91	0.77	0.99	0.82	0.75	–
GDP Growth ($h = 2$)							
2020Q1-2024Q2	1.16	0.94	0.94	0.95	0.94	0.94	–
2021Q1-2024Q2	1.88	0.85	0.84	0.97	0.83	0.69	–
GDP Growth ($h = 4$)							
2020Q1-2024Q2	0.97	0.95	0.96	0.95	0.97	0.96	–
2021Q1-2024Q2	0.77	0.55	0.54	0.55	0.59	0.52	–
Δ Unemployment (2007Q2-2009Q4)							
$h = 1$	0.70	1.54	0.84	0.78	0.94	1.08	–
$h = 2$	0.90	1.16	1.10	0.69	0.98	0.97	–
$h = 4$	0.85	0.88	0.99	0.91	0.93	0.96	–

Notes: The table shows root mean squared errors (RMSEs) relative to the AR(4) model.

Where does the Dual Solution Come From?

- Alternatively, we can invoke the matrix inversion lemma.
- The primal ridge regression solution is:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_P)^{-1}\boldsymbol{X}'\boldsymbol{y}$$

- Using the matrix inversion lemma:

$$(\boldsymbol{A} + \boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1} + \boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}$$

- Set $\boldsymbol{A} = \lambda \boldsymbol{I}_P$, $\boldsymbol{U} = \boldsymbol{X}'$, $\boldsymbol{V} = \boldsymbol{X}$, and $\boldsymbol{C} = \boldsymbol{I}_N$:

$$(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_P)^{-1} = \frac{1}{\lambda}\boldsymbol{I}_P - \frac{1}{\lambda}\boldsymbol{X}'(\boldsymbol{I}_N + {}^1\!/\!_\lambda \boldsymbol{X}\boldsymbol{X}')^{-1}\frac{1}{\lambda}\boldsymbol{X}$$

- Substituting this into the primal solution:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{X}' \underbrace{(\boldsymbol{X}\boldsymbol{X}' + \lambda \boldsymbol{I}_N)^{-1}\boldsymbol{y}}_{\boldsymbol{\alpha}}$$

- This is the dual solution, expressed in terms of the kernel matrix $\boldsymbol{X}\boldsymbol{X}'$.

Where does the Dual Solution Come From?

- Primal problem can be equivalently formulated as:

$$\arg\min_{\boldsymbol{\beta},\boldsymbol{r}} \frac{1}{2}\left(\boldsymbol{r}'\boldsymbol{r} + \lambda\boldsymbol{\beta}'\boldsymbol{\beta}\right) \quad \text{subject to} \quad \boldsymbol{r} = \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y} \tag{1}$$

- Its Lagrangian:

$$L(\boldsymbol{\beta},\boldsymbol{r},\boldsymbol{a}) = \frac{1}{2}\boldsymbol{r}'\boldsymbol{r} + \frac{\lambda}{2}\boldsymbol{\beta}'\boldsymbol{\beta} + \boldsymbol{a}'(\boldsymbol{r} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{y}) \tag{2}$$

- First-order conditions give:

$$\boldsymbol{\beta} = \frac{1}{\lambda}\boldsymbol{X}'\boldsymbol{a}, \quad \boldsymbol{r} = -\boldsymbol{a}$$

- Substituting $\boldsymbol{\beta}$ and $\boldsymbol{r}$ into Lagrangian leads to the dual problem:

$$\arg\min_{\boldsymbol{a}} -\frac{1}{2}\boldsymbol{a}'\boldsymbol{a} - \frac{1}{2\lambda}(\boldsymbol{X}\boldsymbol{a})'(\boldsymbol{X}\boldsymbol{a}) + \boldsymbol{a}'\boldsymbol{y} \tag{3}$$

- Reparametrizing with $\boldsymbol{\alpha} = \frac{1}{\lambda}\boldsymbol{a}$, $\boldsymbol{K} \equiv \boldsymbol{X}\boldsymbol{X}'$, we obtain:

$$\min_{\boldsymbol{\alpha}} (\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha})'(\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}) + \lambda\boldsymbol{\alpha}'\boldsymbol{K}\boldsymbol{\alpha}$$