Forecasting and Trading Futures Contract Spreads with Gaussian Process Regression

Xuan Li

April 5, 2018

Carnegie Mellon University
Civil & Environmental Engineering

Outline

I. **Motivation (why GPR)**

II. **Gaussian Process Regression (GPR) in General (Framework)**

III. **Adapting GPR towards Futures Trading Strategy**

   I. Kernel Covariance Selection (Augmented Functional Representation)
   II. Feature Selection & Engineering & Preprocessing
   III. Training (Parameter Learning)
   IV. Model Evaluation & Comparison
   V. From Prediction to Trading Decision

IV. **Potential Improvements**

I.1 Motivation

1. Definition of futures contracts spread

To profit from the change in the differential of buying/selling the two **related** contracts. Essentially, you consider the **risk** in the difference between two prices (contract) rather than the risk of an immediate futures contract. In other words, **maximizing difference, minimizing risk**.

2. Trading Decision with Risk

Forecast information ratio

$$\widehat{IR} = \frac{\mathbb{E}[\tilde{p}_{t2} - \tilde{p}_{t1}|\mathcal{I}_{t_0}]}{\sqrt{\mathrm{Var}[\tilde{p}_{t2} - \tilde{p}_{t1}|\mathcal{I}_{t_0}]}} \qquad \frac{\text{difference}}{\text{risk}}$$

where

$$\begin{aligned}\mathrm{Var}[\tilde{p}_{t2} - \tilde{p}_{t1}|\mathcal{I}_{t_0}] &= \mathrm{Var}[\tilde{p}_{t2}|\mathcal{I}_{t_0}] + \mathrm{Var}[\tilde{p}_{t1}|\mathcal{I}_{t_0}] - 2\,\mathrm{Cov}[\tilde{p}_{t1}, \tilde{p}_{t2}|\mathcal{I}_{t_0}] \\ &= \mathrm{Cov}[\tilde{p}_{t2}|\mathcal{I}_{t_0}] + \mathrm{Cov}[\tilde{p}_{t1}|\mathcal{I}_{t_0}] - 2\,\mathrm{Cov}[\tilde{p}_{t1}, \tilde{p}_{t2}|\mathcal{I}_{t_0}]\end{aligned}$$

II. Gaussian Processing Regression (GPR)

1. Property of Gaussian Distribution

1.1.1 Marginal Gaussian Distribution

Suppose $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ is jointed Gaussian $\sim N(\mu, \Sigma)$, where

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

and

$$\Sigma = \begin{pmatrix} \Sigma_{11}, \Sigma_{12} \\ \Sigma_{21}, \Sigma_{22} \end{pmatrix}$$

Thus, we have

$$\mathbb{P}(x_1|x_2) \sim N(\mu_{1|2}, \Sigma_{1|2})$$

where

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

and

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

1.1.2 Sum of Gaussian

Given $x_1, x_2 \in \mathbb{R}^d$, $x_1 \perp\!\!\!\perp x_2$, $x_1 \sim N(\mu_1, \Sigma_1)$, $x_2 \sim N(\mu_2, \Sigma_2)$, then we have

$$x_1 + x_2 \sim N(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$$

II. Gaussian Processing Regression (GPR)

2. Inference of GPR

Let $\boldsymbol{X}$ be the matrix of training input, $\boldsymbol{y}$ the vector of output. Assume

$$y_i = f(\boldsymbol{x_i}) + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma_n^2)$$

and the functional operator

$$\boldsymbol{f} = [f(\boldsymbol{x_1}), f(\boldsymbol{x_2}), ..., f(\boldsymbol{x_N})] \sim N(\boldsymbol{0}, K(\boldsymbol{X}, \boldsymbol{X})) \qquad K(\boldsymbol{X}, \boldsymbol{X})_{ij} = k(\boldsymbol{X}_i, \boldsymbol{X}_j)$$

where $k(\boldsymbol{u}, \boldsymbol{v})$ is kernel covariance operator that for example

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp(-\frac{||\boldsymbol{u} - \boldsymbol{v}||^2}{2\sigma_l^2})$$

Thus, the regression problem is that given $\boldsymbol{X}, \boldsymbol{y}$, and $\boldsymbol{X}_*$, how could we estimate $\mathbb{E}[\boldsymbol{y}_*|\boldsymbol{y}]$ and $\mathrm{Cov}[\boldsymbol{y}_*|\boldsymbol{y}]$?

II. Gaussian Processing Regression (GPR)

2. Inference of GPR, cont

First, use *'sum of Gaussian'* property, suppose $\boldsymbol{y} \in \mathbb{R}^N, \boldsymbol{y}_* \in \mathbb{R}^M$,we know that

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{X}) \\ \boldsymbol{f}(\boldsymbol{X}_*) \end{pmatrix} + \begin{pmatrix} \sigma_n^2 \mathbf{1}_{\boldsymbol{N}} \\ \sigma_n^2 \mathbf{1}_{\boldsymbol{M}} \end{pmatrix} \sim N\left(\mathbf{0}, \begin{pmatrix} K(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}_N, & K(\boldsymbol{X},\boldsymbol{X}_*) \\ K(\boldsymbol{X}_*,\boldsymbol{X}), & K(\boldsymbol{X}_*,\boldsymbol{X}_*) + \sigma_n^2 \boldsymbol{I}_M \end{pmatrix}\right)$$

Second, use *'Marginal Gaussian Distribution'* property, we have

$$\mathbb{E}[\boldsymbol{y}_*|\boldsymbol{y}] = K(\boldsymbol{X}_*,\boldsymbol{X})(K(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}_N)^{-1}\boldsymbol{y}$$

and

$$\mathrm{Cov}[\boldsymbol{y}_*|\boldsymbol{y}] = K(\boldsymbol{X}_*,\boldsymbol{X}_*) - K(\boldsymbol{X}_*,\boldsymbol{X})(K(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}_N)^{-1}K(\boldsymbol{X},\boldsymbol{X}_*)$$

III. Adapting GPR towards Futures Trading Strategy

1. Kernel covariance selection (Augmented functional representation)

The key idea of GPR lies in the choice of kernel covariance operator (function), which models the relationship between corresponding observations (input, x). In this paper, the authors choose:

$$k(\boldsymbol{u}, \boldsymbol{v}|\ell, \alpha, \sigma_f, \sigma_{TS}) = \sigma_f^2 \left(1 + \frac{1}{2\alpha} \sum_{k=1}^{d} \frac{(\boldsymbol{u}_k - \boldsymbol{v}_k)^2}{\ell_k^2}\right)^{-\alpha} + \sigma_{TS}^2 \delta(\boldsymbol{u}_i, \boldsymbol{v}_i)$$

where $\delta(\boldsymbol{u}_i, \boldsymbol{v}_i)$ (or $\delta_{\boldsymbol{u}_i, \boldsymbol{v}_i}$) is the Kronecker delta that

$$\delta(\boldsymbol{u}_i, \boldsymbol{v}_i) = \begin{cases} 1, & \boldsymbol{u}_i = \boldsymbol{v}_i \\ 0, & \boldsymbol{u}_i \neq \boldsymbol{v}_i \end{cases}$$

The authors considered the covariance based on contribution of each dimension of the input features separately, and the effect of a dominant feature using Kronecker delta function.

III. Adapting GPR towards Futures Trading Strategy

2. Feature Selection & Engineering & Preprocessing

In all, the feature used in the paper include:

- current time-series index or year: the Kronecker delta term is applied on this dimension
- operation time: the time at which the forecast is made
- target time: the time at which the forecast is being made
- current spread price
- the price of the three nearest future contracts
- stock-to-use ratio:
- year-over-year difference in total ending stocks

All of these features are standardized with zero mean and unit deviation. For the price target ***y,*** the trajectory is standardized to start at zero at the start of every year, by subtracting the first price value.

III. Adapting GPR towards Futures Trading Strategy

3. Training (parameter learning)

Given the kernel covariance function, the total hyper-parameter needed to be learned are

- $\alpha \in \mathbb{R}$
- $\sigma_f \in \mathbb{R}$
- $\sigma_{TS} \in \mathbb{R}$
- $\sigma_n \in \mathbb{R}$
- $\ell \in \mathbb{R}^d$

$$k(\boldsymbol{u}, \boldsymbol{v}|\boldsymbol{\ell}, \alpha, \sigma_f, \sigma_{TS}) = \sigma_f^2 \left(1 + \frac{1}{2\alpha} \sum_{k=1}^{d} \frac{(\boldsymbol{u}_k - \boldsymbol{v}_k)^2}{\boldsymbol{\ell}_k^2}\right)^{-\alpha} + \sigma_{TS}^2 \delta(\boldsymbol{u}_i, \boldsymbol{v}_i)$$

As the authors have claimed in the paper, one efficient approach to find the hyper-parameter for Gaussian Process is to maximize the **marginal likelihood** of the observed data (given the hyper-parameter). To be specific, define $\boldsymbol{\theta}$ as the hyper-parameter, $\boldsymbol{y} \in \mathbb{R}^N$ are the observation, and $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ are the input features, then

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \log \mathbb{P}(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta})$$

Since from Part one we know that $\mathbb{P}(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta})$ follows Multivariate Gaussian (normal) Distribution $N(\boldsymbol{0}, \boldsymbol{\Sigma})$ $(\boldsymbol{\Sigma} = K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}_N)$ given the kernel function $k(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\theta})$, which is

III. Adapting GPR towards Futures Trading Strategy

3. Training (parameter learning) cont.

$$\mathbb{P}(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^N|\boldsymbol{\Sigma}|}} \exp\big(-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}\big)$$

where $|\boldsymbol{\Sigma}| = \det(\boldsymbol{\Sigma})$. Thus, to put it back to equation above, we have that

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\text{argmax}} \ -\frac{1}{2}\boldsymbol{y}^T(K(\boldsymbol{X},\boldsymbol{X};\boldsymbol{\theta}) + \sigma_n^2\boldsymbol{I}_N)^{-1}\boldsymbol{y} - \frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X};\boldsymbol{\theta}) + \sigma_n^2\boldsymbol{I}_N| - \frac{N}{2}\log(2\pi)$$

which is equivalent to

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\text{argmax}} \ -\frac{1}{2}\boldsymbol{y}^T(K(\boldsymbol{X},\boldsymbol{X};\boldsymbol{\theta}) + \sigma_n^2\boldsymbol{I}_N)^{-1}\boldsymbol{y} - \frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X};\boldsymbol{\theta}) + \sigma_n^2\boldsymbol{I}_N|$$

III. Adapting GPR towards Futures Trading Strategy

3. Training (parameter learning) cont.

Gradient Descent

Given the expression of objective and form of variables, the problem is a non convex optimization problem and could be potentially solved by gradient descent. First, we can get the gradient of the objective function as

$$\frac{\partial \log \mathbb{P}(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Sigma}^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\theta}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y} - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\cdot\frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\theta}})$$

Thus, the key component to get the gradient is to calculate the matrix $\frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\theta}}$. Given the definition in Section 2.2.2, and

$$\boldsymbol{\Sigma} = K(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2\boldsymbol{I}_N$$

for each of the parameter in $\boldsymbol{\theta}$, we have

$$(\frac{\partial \boldsymbol{\Sigma}}{\partial \alpha})_{i,j} = \sigma_f^2\left(1+\frac{c}{2\alpha}\right)^{-\alpha}\cdot(-\ln(1+\frac{c}{2\alpha}) + \frac{1}{1+2\alpha/c}), \quad c = \sum_{k=1}^{d}\frac{(\boldsymbol{X}_k^i - \boldsymbol{X}_k^j)^2}{\ell_k^2}$$

$$(\frac{\partial \boldsymbol{\Sigma}}{\partial \sigma_f})_{i,j} = 2\sigma_f\left(1+\frac{1}{2\alpha}\sum_{k=1}^{d}\frac{(\boldsymbol{X}_k^i - \boldsymbol{X}_k^j)^2}{\ell_k^2}\right)^{-\alpha}$$

$$(\frac{\partial \boldsymbol{\Sigma}}{\partial \sigma_{TS}})_{i,j} = 2\sigma_{TS}\delta(\boldsymbol{X}_d^i, \boldsymbol{X}_d^j)$$

III. Adapting GPR towards Futures Trading Strategy

3. Training (parameter learning) cont.

Gradient Descent

$$(\frac{\partial \Sigma}{\partial \sigma_n})_{i,j} = 2\sigma_n \delta(i,j)$$

$$(\frac{\partial \Sigma}{\partial \ell_{\bar{d}}})_{i,j} = \sigma_f^2 \left(1 + \frac{1}{2\alpha}\sum_{k=1}^{d} \frac{(X_k^i - X_k^j)^2}{\ell_k^2}\right)^{-\alpha-1} \cdot \frac{(X_{\bar{d}}^i - X_{\bar{d}}^j)^2}{\ell_{\bar{d}}^3}$$

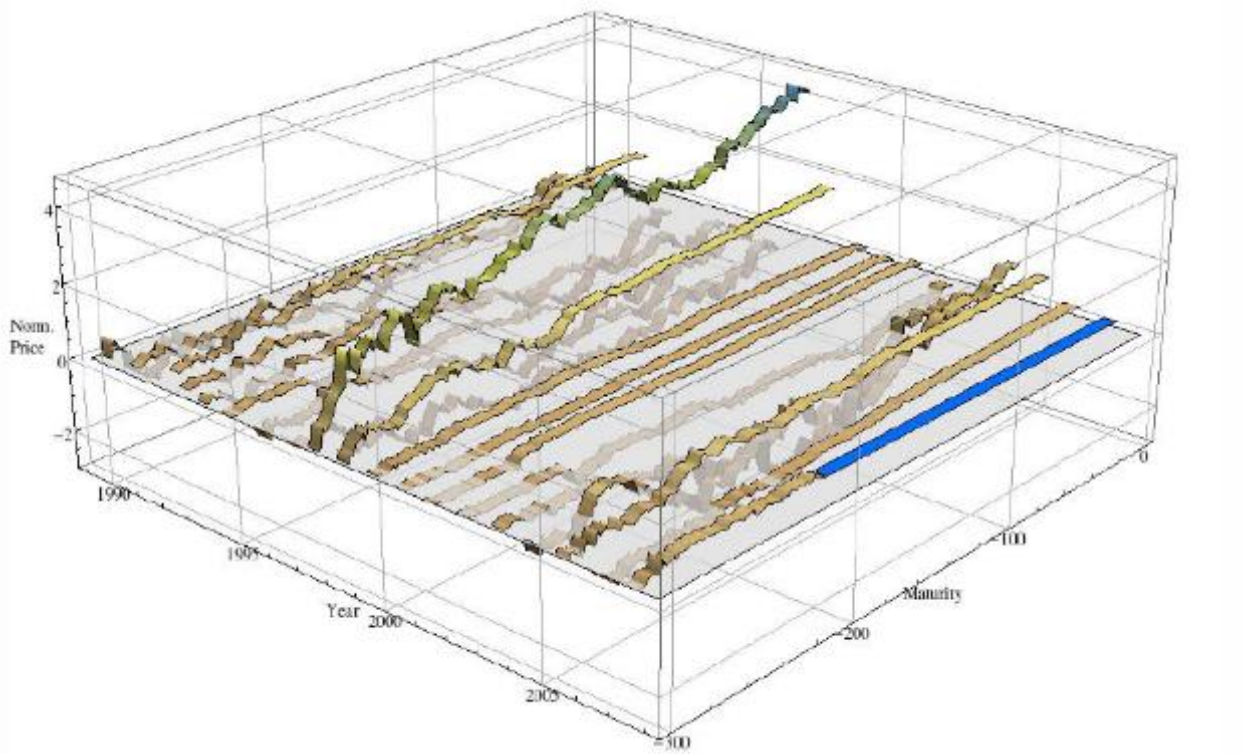
Algorithm 1: Gradient Descent (Gaussian Process)

Data: X, y
Parameter: θ
initialization: $\text{epoch}^{\max}, \text{epoch} = 0, \theta, \omega, \epsilon$;
while $\text{epoch} < \text{epoch}^{\max}$ **do**
  $\theta_i = \theta_i + \omega \cdot \frac{\partial \log \mathbb{P}(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i}$;
  if $|\log \mathbb{P}(y|X;\theta) - \log \mathbb{P}(y|X;\theta')| > \epsilon$ **then**
    epoch ++;
  else
    break;
  end
end

III. Adapting GPR towards Futures Trading Strategy

3. Training (parameter learning) cont.

multi-stage training + sub-sampling



III. Adapting GPR towards Futures Trading Strategy

4. Model Evaluation & Comparison between models

4.1 Significance Test

cross-covariance-corrected Diebold-Mariano test for sequential correlated sequence (error trend), with the assumption of covariance stationarity.

Let d_t be the sequence of error difference between two models to be compared. The tested statistic

$$\mathrm{DM} = \bar{d} / \sqrt{\hat{v}_{\mathrm{CCC-DM}}}$$

is asymptotically distributed as $N(0,1)$, where

$$\bar{d} = \frac{1}{M} \sum_t d_t$$

and

$$\hat{v}_{\mathrm{CCC-DM}} = \frac{1}{M^2} \left(\sum_i M_i \sum_{k=-K}^{K} \hat{\gamma}_k^i + \sum_i \sum_{j \neq i} M_{i \cap j} \sum_{k=-K'}^{K'} \hat{\gamma}_k^{i,j} \right)$$

where M_i is the number of samples in test set i, M is the total number of samples, $\hat{\gamma}_k^i$ is the estimated lag-k autocovariance, and $\hat{\gamma}_k^{i,j}$ is the estimated lag-k cross-covariance.

III. Adapting GPR towards Futures Trading Strategy

4. Model Evaluation & Comparison, cont.

4.2 Criterion

The criterion used are square error (SE) and negative log-likelihood (NLL). SE is normalized by dividing it with the standard deviation of the test target. NLL is normalized by subtracting the likelihood of a univariate Gaussian distribution estimated on the test target.

4.3 Baseline Models

- **AugRQ/all-inp**: proposed model
- **AugRQ/less-inp**: do not include the economic variables
- **AugRQ/no-inp**: only with time relevant variables
- **stdRQ/all-inp**: single length parameter
- **stdRQ/no-inp**: single time variable Δ
- **Linear/all-inp**: Bayesian linear regression

III. Adapting GPR towards Futures Trading Strategy

5. From Prediction to Trading Decision

5.1 Prediction

holding the time series index constant to N, the operation time constant to the time M_N of the last observation, the other input variable constant to their last-observed values $x_{M_N}^N$ (slow-moving variables that represent a "level"), and varying the target time over the forecasting period Δ.

5.2 Trading Decision

Forecast information ratio

$$\widehat{IR} = \frac{\mathbb{E}[\tilde{p}_{t2} - \tilde{p}_{t1}|\mathcal{I}_{t_0}]]}{\sqrt{\mathrm{Var}[\tilde{p}_{t2} - \tilde{p}_{t1}|\mathcal{I}_{t_0}]}}$$

where

$$\mathrm{Var}[\tilde{p}_{t2} - \tilde{p}_{t1}|\mathcal{I}_{t_0}] = \mathrm{Var}[\tilde{p}_{t2}|\mathcal{I}_{t_0}] + \mathrm{Var}[\tilde{p}_{t1}|\mathcal{I}_{t_0}] - 2\,\mathrm{Cov}[\tilde{p}_{t1}, \tilde{p}_{t2}|\mathcal{I}_{t_0}]$$

IV. Potential Improvements

- **Scalability** with large dataset, and efficient training algorithm. Implemented in the paper: random sampling subset + retraining, might introduce bias of prediction accuracy towards the selected input. Potential refinement:

  (a) approximate $\boldsymbol{\Sigma}$ with lower rank matrix [2,3]. For example,

$$\boldsymbol{\Sigma} \approx \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}[:, \mathcal{I}] \ \boldsymbol{\Sigma}[\mathcal{I}, \mathcal{I}]^{-1} \ \boldsymbol{\Sigma}[\mathcal{I}, :]$$

- **Assumption** in prediction: the slow-moving assumption of features. "for slow-moving variables that represent a level, one can conceivably keep their value constant to the last known realization across the forecasting period". Unused features, could be calibrated in the kernel function as well.
- **Feature** selection: any better, or more (temporal) features that are not 'slow-moving'? any better modelling of kernel?
- **extro-covariance** with other type of futures contract.