HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 2 (2024)

Change Point Detection for Automatic Time Series Forecasting

Siddharth Chander[1], Ricky Liang[2] and Johnathan P. Chen[#]

[1]Dougherty Valley High School, USA
[2]YK Pao School, China
[#]Advisor

ABSTRACT

Time series analysis, the process of learning patterns in time series data and making future predictions, is a challenging machine learning endeavor: the intricate and error-prone process of manual data fitting limits efficiency and scalability to larger datasets. Our research builds upon structure discovery research, paving the way for more effective short-term forecasting. We combine the adaptability of automatic change point detection with a fixed kernel composition, achieving accuracy comparable to traditional manual methods while reducing analysis time. This hybrid approach offers the best of both worlds: leveraging human expertise for precise fitting and capturing specific trends while utilizing an automated technique to recognize shifts and model complex relationships within the data dynamically. By demonstrating the effectiveness of automatic change point detection in conjunction with kernel composition, we work to develop time series usage in data analysis.

Introduction

Time series challenges, seen in fields like finance and global weather predictions, require a solid grasp of structure in data for effective forecasting (Hyndman & Athanasopoulos, 2013) due to significant and sporadic variations that do not happen consistently compared to other types of data. Traditionally, understanding these structures involved complex and error-prone bespoke data science, relying on manual data fitting (Wang et al., 2022). Manual forecasting, relying on human expertise, is a longstanding method for predicting trends but has drawbacks due to its subjective nature and time-intensive process (Armstrong, 2001). To address the need for simpler and automated approaches, researchers developed methods to streamline this process. Such methods include the fully automatic fixed kernel composition method, which models the data without having to input kernels to use, but takes up more computational resources to provide a more accurate result (Corani et al., 2021).

Automating the kernel selection process can make time series forecasting more powerful and accessible to data scientists and software engineers (Corani et al., 2021). In our study, we propose a new method by combining automatic change point detection with a Gaussian Process with a composite kernel from (Corani et al., 2021). This approach competes effectively with manual methods and fully automatic forecasting, providing accuracy and potential time savings in scenarios like modelling sales data that can fluctuate on a near-daily basis or sensor-provided data, which would have uneven patterns depending on what is being measured (Hyndman & Athanasopoulos, 2013). Our research marks progress in making time series analysis more accessible and efficient, offering insights for researchers in this complex field.

Specifically, our exploration involves combining Gaussian Processes (GPs) with change point detection methods and assessing their performance against manual forecasting and fixed kernel composition (FKC) baselines. GPs are adept at handling uncertain or noisy data, providing valuable predictions where confidence is crucial (Rasmussen, 2006). Kernel selection allows for precise fitting, capturing specific trends, and modeling complex relationships. Choosing suitable kernels and determining the similarity between data points is crucial

for constructing effective GP models (Rasmussen, 2006). By combining a fixed kernel composition with change point detection, our newest model excels at identifying specific points in a time series where significant statistical changes occur (Burg & Williams, 2020). Unlike using only an FKC, modeling change points dynamically recognizes shifts in trends, improving overall model fit (Burg & Williams, 2020). We show on the Kaggle M4 and M5 forecasting competitions that for comparable run times, our automatic model is competitive with manual methods with improved performance ("Kaggle Competitions," n.d.).

Gaussian Processes for Time Series

A time series problem involves predicting future values of a variable based on its past observations, where each observation is recorded at discrete time steps.

Time series data can be represented as follows:

$$X = (x_1, \ldots, x_n)$$
$$Y = (y_1, \ldots, y_n)$$

where x_i represents the observation at time step i in the input sequence, and y_i represents the corresponding output. The forecasting problem involves predicting future values y_{n+h} which involves computing:

$$p(y_{n+1}, \ldots, y_{n+h} | y_1, \ldots, y_n)$$

Gaussian Process

A GP is a set of random variables such that the joint distribution of any finite subset of these variables follows a Gaussian distribution. We can think of a GP as a distribution over functions, where our goal is to learn a posterior over all possible functions that can explain the data (Carl Edward Rasmussen, 2005). A GP is fully defined by a mean and covariance function, the latter of which is referred to as a kernel.

The mean function in GPs can take various forms. A commonly used choice, and the one used in our research, is a constant $\mu(x) = \hat{\mu}$. When fitting a GP model to data, $\hat{\mu}$ is often estimated using the available data. For other works, beyond using a constant for the mean, a more general mean function $\mu(x)$ of a Gaussian process is given by the expected value of the process at a specific point x (Carl Edward Rasmussen, 2005):

$$\mu(x) = \mathbb{E}[f(x)]$$

We use a constant mean function in our paper, and rely on the kernels described below to capture the data patterns.

Covariance Functions

The kernel determines the strength of the correlation between points (Carl Edward Rasmussen, 2005) . Mathematically, the covariance function $k(x, x')$ is defined as the expected value of the product of the differences between the target values and their respective mean values:

$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$$

where the variables x and x' represent input data points in our context, $f(x)$ and $f(x')$ are the function values at those points, and $\mu(x)$ and $\mu(x')$ are the mean values of the function at those points.

Kernel selection and discovery is an open area of research. The most common kernel is the RBF kernel, also known as the squared exponential kernel:

$$K_{\text{RBF}}(x, x') = \exp\left(-\frac{\| x - x' \|^2}{2l^2}\right)$$

This kernel is effective for modeling non-linear trends, which computes the distance between pairs of data points in a feature space. The lengthscale parameter l in equation [rbfkerneleq] determines how relevant a feature (parameter) is for the learned function, which is visually represented by the smoothness of the functions. Higher lengthscale will yield smoother functions, while lower lengthscale yields rougher functions. It is worth noting that a shorter lengthscale in Gaussian processes increases sensitivity to local variations in the training data, raising the risk of overfitting (fitting too precisely to one dataset and performing poorly on other datasets) by capturing noise rather than underlying patterns, which can result in poor performance on new, unseen data.

The Matérn kernel, which is interestingly the generalization of the RBF kernel although it appears less frequently, is:

$$K_{\text{Matern}}(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\,\| x - x' \|}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}\,\| x - x' \|}{l}\right)$$

This kernel has an additional parameter ν, which controls the smoothness of the function. The smaller ν is, the less smooth the function is. As $\nu \to \infty$, the kernel becomes equivalent to the RBF kernel. In contrast, the lengthscale parameter l primarily governs the scale of variations, with smaller values capturing finer details and larger values promoting smoother behavior.

A linear kernel, a simpler kernel that captures linear trends, is:

$$K_{\text{Linear}}(x, x') = x^T x'$$

It is best used to capture linear relationships between data points but is ineffective when dealing with non-linear ones due to not having a lengthscale parameter.

Another common kernel is the Periodic (PER kernel):

$$K_{\text{Periodic}}(x, x') = \exp\left(-\frac{2\sin^2(\pi \| x - x' \|/p)}{l^2}\right)$$

Most time series often consist of periodic patterns, whether daily, monthly, or annually, and a periodic kernel can capture the repeated trends within the data. The lengthscale parameter l here operates similarly to the lengthscale parameter in the RBF kernel, where a longer lengthscale decreases the local variance within repetition and captures less detail.

The Rational Quadratic (RQ) kernel is defined as:

$$K_{\text{RQ}}(x, x') = \left(1 + \frac{\| x - x' \|^2}{2\alpha l^2}\right)^{-\alpha}$$

This kernel can be interpreted as an infinite number of different RBF kernels with different lengthscales added together. This makes it suitable for dealing with complex data sets that don't have a simple pattern. The lengthscale parameter here controls the spread of the covariance. It has a positive correlation, while

the scale-mixture manipulates the number of local variations, increasing the scale-mixture and reducing local variations.

The Spectral Mixture (SM) kernel is:

$$K_{\mathrm{SM}}(x, x') = \sum_{k=1}^{K} w_k \cos(2\pi f_k \parallel x - x' \parallel + \phi_k) \exp(-2\pi^2 f_k^2 l^2 \parallel x - x' \parallel^2)$$

where

1. K: Number of mixture components.
2. w_k: Weight associated with the k-th component.
3. f_k: Frequency of the k-th component.
4. ϕ_k: Phase of the k-th component.
5. l: Lengthscale parameter governing the scale of variations.
6. $\parallel x - x' \parallel$: Euclidean distance between input data points x and x'.

This kernel is powerful because it learns all the frequencies in the data by learning the spectral density. The spectral density indicates the probability density of the corresponding kernel or how likely each frequency is in the data. Hence, the SM kernel can capture extremely complex patterns.

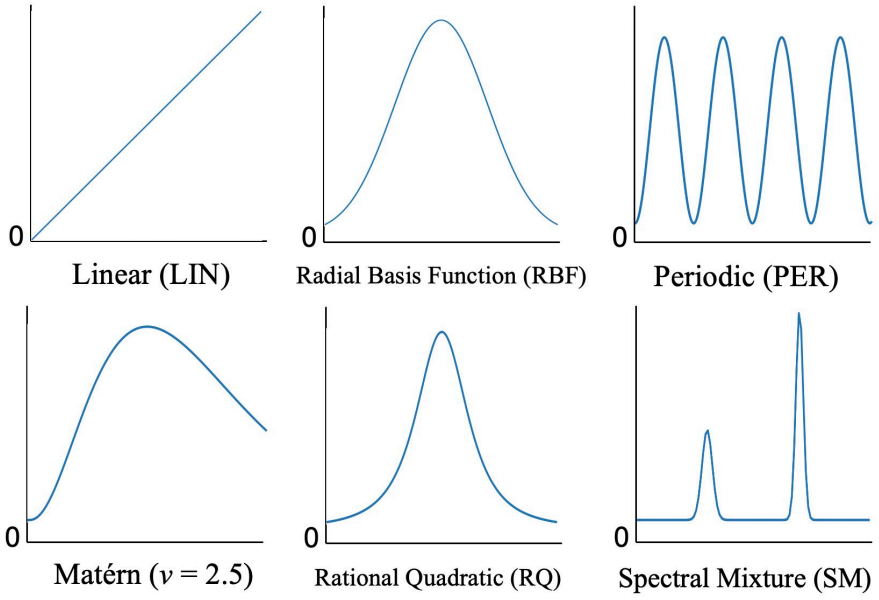

Figure1. Functions that GP kernels model.

Models

Manual Forecasting

We refer to "manual forecasting" as the trial and error process of fitting a model (in our case a GP kernel) to data. In this approach, researchers carefully analyze the data to discern trends and try suitable kernels. The process of kernel composition then takes place, involving the addition or multiplication of kernels to construct

a model capable of capturing a diverse range of patterns. This is particularly advantageous when dealing with data exhibiting complex functions.

Similar to architecture search in deep learning, the resulting fit is a local optima conditioned on the quality of the kernels the modeler selects. It is still the predominant method of fitting short term time series since researchers can make minute adjustments at every step of the modeling process. Unlike automated methods, researchers may possess information not present in the data, enabling them to make more comprehensive predictions. Though time-consuming, their unique understanding of the data may yield unexpected outcomes that automated predictions cannot replicate.

Fixed Kernel Composition

Fixed kernel composition (FKC) uses a single fixed kernel for all problems of a certain characteristic, such as time series data. Instead of manually selecting and adjusting kernels to construct forecasting models, we use a single fixed kernel for all the data sets as in (Corani et al., 2021). The biggest change is removing the PER kernel where no seasonal pattern was observed in the M4 data.

$$\begin{aligned} \mathrm{K_{M4}} &= \mathrm{RBF} + \mathrm{SM1} + \mathrm{SM2} \\ \mathrm{K_{M5}} &= \mathrm{RBF} + \mathrm{SM1} + \mathrm{SM2} + \mathrm{PER} \end{aligned}$$

Corani et al. also adopted a hierarchical Bayesian perspective to FKC, treating the hyperparameters of each kernel as random variables and learning a distribution for each one. This allowed them to quantify the uncertainty associated with the hyperparameter and potentially improve the model's robustness to overfitting. Instead, we employ maximum likelihood estimation (MLE) to learn a single point estimate of the hyperparameter that best explains the observed data. While this method doesn't explicitly account for uncertainty, it is computationally efficient and performed sufficiently well, particularly with the larger datasets we used. Fixed kernel composition may lack the depth of human insights and domain-specific knowledge present in manual forecasting, potentially limiting accuracy. Finally, relying solely on statistical measures in automation, such as training a machine learning model exclusively on historical data, can lead to overfitting. For instance, a stock prediction model may become too tailored to past trends, limiting its adaptability to new market conditions.

Change Point Detection

Change point detection is a method used to identify points in a time series where abrupt and significant variations have occurred . We use binary segmentation as described below for change points and then fit RBF kernels on each window of data split by the changepoints.

$$K_{\text{input\_change}}(x_1, x_2; \{\lambda, \sigma_1, \sigma_2\}) = \lambda^2 \exp\left(-\frac{1}{2\sigma_2^2}|x_1 - x_2|\right), \quad \text{for } x_1 \geq \text{change point}$$

where:

λ :Scaling parameter, it scales the overall magnitude of the kernel.
σ_2 :Width parameter, controls how quickly the kernel decreases with distance.

Since the change points separate time series into segments with similar characteristics, it can more accurately and precisely capture the trend and use it to forecast future events. This becomes especially practical in real-world applications where data are constantly affected by external factors that lead to abrupt changes within the dataset. Change points also segment data into different periods by characteristics, giving researchers

deeper insights into understanding the properties and behaviors of each segment, potentially allowing businesses to make better decisions.

Our model demonstrates novelty by combining change point detection with fixed kernel composition. Specifically, we have incorporated change point detection using binary segmentation into a GP regression model with change point kernels constructed from 3 RBF kernels. Here, each RBF model captures a different trend, increasing the flexibility and trend-capturing ability of the model.

Binary Segmentation

We use a greedy binary segmentation method to identify points in a sequence of data where a significant change or "break" occurs, which is the change point. After detecting a change point within the entire time series, the time series is split into two parts around this change point. Then, the detection process is repeated on the two resulting segmented time series. This process is repeated until no further change point is detected

The initial change point, denoted as $\hat{t}^{(1)}$, is given by:

$$\hat{t}^{(1)} := \arg\min_{1 \le t < T-1} [c(y_{0..t}) + c(y_{t..T})]$$

where $c(\cdot)$ is the cost function. This operation is "greedy," searching for the change point that minimizes the sum of costs the most. The signal is then split into two at the position of $\hat{t}^{(1)}$. This process is repeated on the resulting sub-signals until a stopping criterion is met. The algorithm stops when further partitioning of the data ceases to yield a substantial reduction in the total cost. The algorithm's efficiency is of the order $O(T\log T)$.

Experiments

The datasets we have chosen for this experiment are snippets of data from the M4 (Makridakis et al., 2020) and M5 (Makridakis et al., 2022) datasets, which are datasets used in the series of the Makridakis competition that evaluate and compare different forecasting methods. We select 1871 time series in the M4 dataset with 1853 data points for training and 18 data points for testing, and 1000 time series in the M5 dataset with the first 900 data points for training and the last 100 data points for testing. We applied 3 different forecasting approaches to both datasets: manual kernel construction, fixed kernel composition, and change point detection. The first two approaches are the baselines, while we are particularly interested in the results of the change point detection approach. We used GPytorch (Gardner et al., 2018) and the autoforecasting library (Chen et al., n.d.) with the following manual kernels to implement and run the experiments.

$$\begin{aligned} \mathrm{K_{M4}} &= \mathrm{RBF} \times \mathrm{PER} + \mathrm{RQ} \\ \mathrm{K_{M5}} &= \mathrm{RBF} \times \mathrm{PER} + \mathrm{MAT} \end{aligned}$$

Require: Signal $\{y_t\}$, T, cost function $c(\cdot)$, stopping criterion.
Initialize $L \leftarrow \{\}$. ▷ Estimated breakpoints.
repeat
  $k \leftarrow |L|$. ▷ Number of breakpoints
  $t_0 \leftarrow 0$ and $t_{k+1} \leftarrow T$. ▷ Dummy variables.
  if $k > 0$ **then**
    Denote by t_i (for $i = 1, \ldots, k$) the elements in ascending order of L, i.e., $L = \{t_1, \ldots, t_k\}$.
  end if
  Initialize G as a $(k+1)$-long array. ▷ List of gains
  for $i = 0, \ldots, k$ **do**
    $G[i] \leftarrow c(y_{t_i}..t_{i+1}) - \min_{t_i < t < t_{i+1}} [c(y_{t_i}..t) + c(y_t..t_{i+1})]$.
  end for
  $b_i \leftarrow \arg\max_i G[i]$.
  $\hat{t} \leftarrow \arg\min_{t_{b_i} < t < t_{b_i+1}} [c(y_{t_{b_i}}..t) + c(y_t..t_{b_i+1})]$.
  $L \leftarrow L \cup \{\hat{t}\}$.
until stopping criterion is met.
Ensure: Set L of estimated breakpoint indexes.

Figure2. BinSeg Algorithm

Multiplying the RBF and Periodic kernels and adding the result to the RQ kernel creates a composite kernel structure. This combination allows the model to capture long-term and periodic patterns in the data through the RBF-Periodic component while accounting for more complex and irregular relationships with the RQ component. This combination is advantageous because it provides a flexible way to model data that exhibits both smooth long-term trends and periodic behavior and enhances the model's robustness to noisy data and irregularities in the data, using the detailed set of kernels in the specific combination, the GP model was trained with about 98-99 percent of the selected M4 row, and the rest of the data was used for testing, with 1000 iterations of training being used for the model.

Metrics

Mean Absolute Error (MAE) is a metric commonly used to evaluate the accuracy of predictive models, particularly in regression problems. It measures the average absolute difference between the predicted values ($\hat{y}_t$) and the true values (y_t) as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T} | y_t - \hat{y}_t |$$

Here, T represents the number of data points or time instances in the test set. A lower MAE indicates a better fit of the model of the data, as it quantifies the average magnitude of errors between predictions and actual values.

Continuous-Ranked Probability Score (CRPS) is a metric used to assess the accuracy of probabilistic forecasts, such as predictive distribution functions. It compares these probabilistic forecasts with the observed data. CRPS is defined as follows:

$$CRPS(F_t, y_t) = -\int_{-\infty}^{\infty} \left(F_t(z) - \mathbf{1}_{z \geq y_t}\right)^2 \, dz$$

Here, $F_t(z)$ represents the cumulative predictive distribution function at time t, and y_t is the observed target value at time t. The indicator function $\mathbf{1}_{z \geq y_t}$ is 1 for $z \geq y_t$ and 0 otherwise. A lower CRPS value indicates a better fit of the probabilistic forecast to the actual observations.

Log-likelihood (LL) is a measure used to evaluate how well a statistical model describes the observed data, particularly in the context of models providing probabilistic predictions. The LL for a test set is defined as follows:

$$LL = \frac{1}{T}\left(-\frac{1}{2}\sum_{t=1}^{T} \log\left(2\pi\sigma_t^2\right) - \frac{1}{2\sigma_t^2}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2\right)$$

Here, T is the number of data points in the test set, σ_t^2 represents the variance of the predictive distribution at time t, y_t is the observed value, and $\hat{y}_t$ is the predicted value. A higher LL indicates a better match between the model's predictions and the observed data, as it represents the likelihood.

Our study employed three distinct forecasting methods—manual forecasting, FKC, and change point—on randomly selected time series from the M4 and M5 datasets. Each method was for 1,000 iterations, utilizing 90% of the data, and 10% withheld for test. We measured MAE, CRPS, LL, and CPU run times averaged over five runs.

Results

Analyzing the change point model's performance on the M4 dataset reveals the model's observations. The model's confidence bounds are larger than the manual model's and FKC's. Yet, they encapsulate the dataset's dynamics without succumbing to the overfitting of the dataset seen with the FKC model. This suggests flexibility in the model's representation of general trends and avoids an overly close adherence to noise or outliers.

Examining the shape of the confidence bounds and their alignment with the test mean (3) indicates that the model adeptly reflects the underlying structure of the M4 data. This ability to capture data dynamics without overfitting distinguishes the model from FKC counterparts, which might tend to closely follow training data at the risk of losing generalization. Regarding test data predictions, the model follows the overall trend of the M4 data but falls short in capturing a final small upward trend present in the actual training data. This discrepancy is reflected in an increasing variance between the predicted test mean and the real test data, highlighting a limitation in the model's ability to forecast subtle shifts in data and make better general trend predictions.

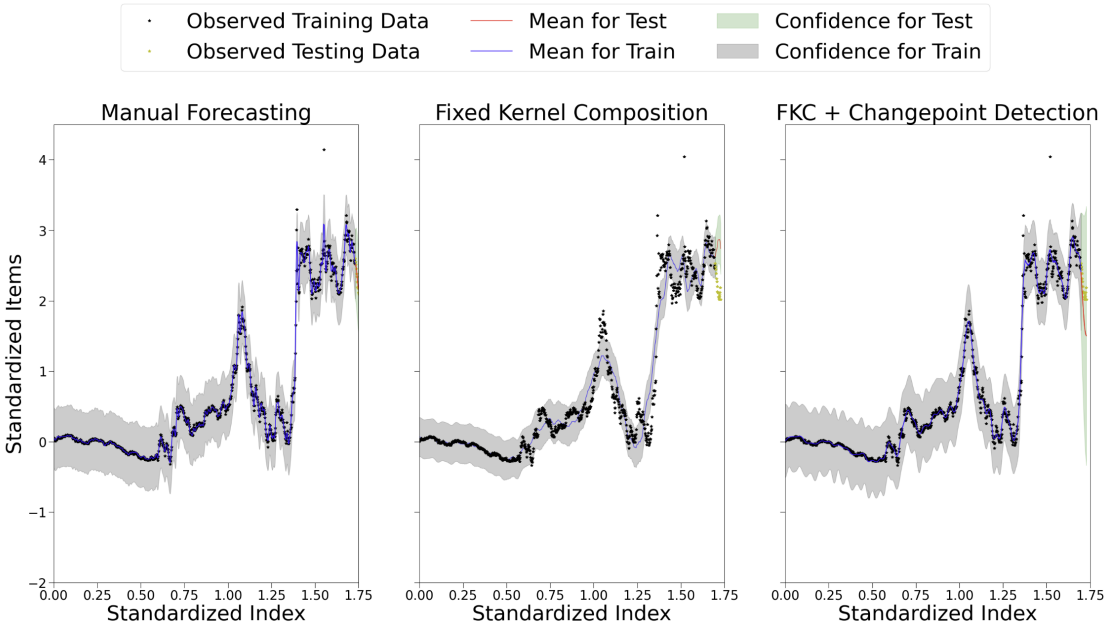

Figure3. This is the overview of the M4 dataset that analyzes anonymous monthly sales figures, showing training and testing means calculated by all three models, along with the data used for training and testing, and the confidence bounds of each model.

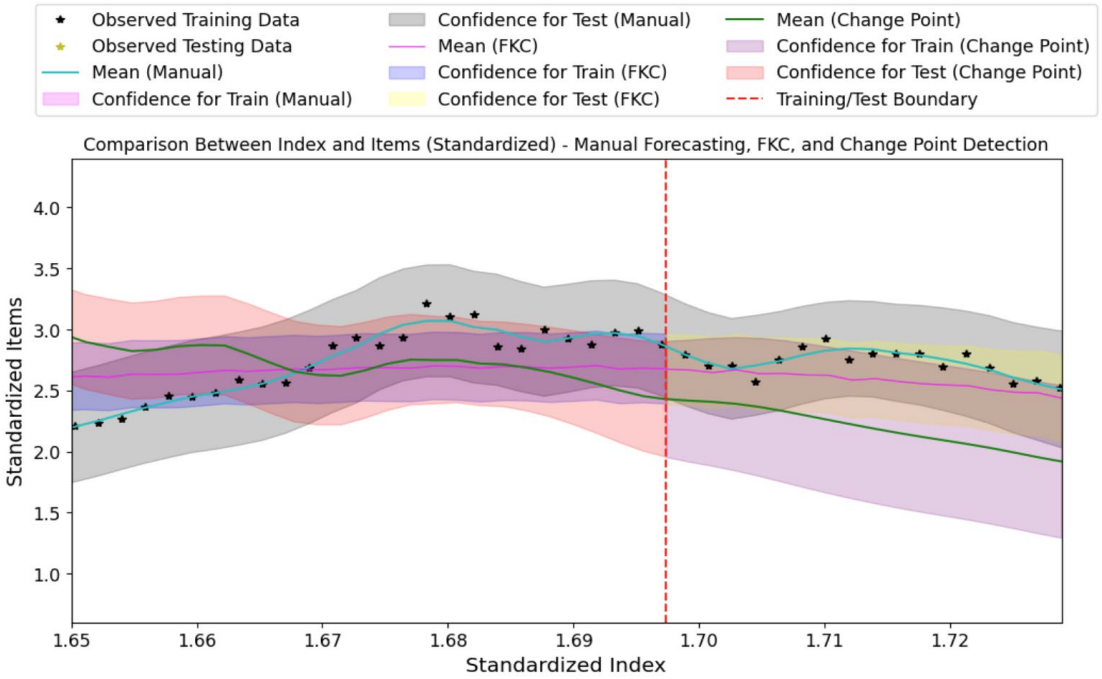

Figure4. This is the testing portion of the M4 dataset, showing the testing means calculated for each model, along with the test data, and the confidence bounds for each model.

Table1. M4 Numerical Results

	Manual	**FKC**	**FKC + CP**
MAE	$\mathbf{0.0852 \pm 0.0168}$	0.412 ± 0.0115	0.1563 ± 0.0295
CRPS	0.084 ± 0.0051	0.3124 ± 0.0105	$\mathbf{0.0803 \pm 0.0087}$
NLL	$\mathbf{0.2916 \pm 0.0077}$	0.7544 ± 0.0090	0.3142 ± 0.0221
Average CPU Time Per Iteration	$\mathbf{7.32 \pm 0.31}$	25.65 ± 0.94	8.71 ± 0.84

A noteworthy aspect of the change point model's behavior is its handling of outliers. Instead of becoming confused or overcompensating for these anomalies, the model strategically uses them. Outliers are incorporated to subtly adjust the training mean, guiding it towards a more comprehensive fit that considers the entire dataset, including outliers. This approach indicates adaptability without succumbing to the influence of outliers that might lead to overfitting.

In contrast to the M4 dataset, the Change-Point detection model encountered challenges when applied to the more periodic M5 dataset. The periodic patterns inherent in M5 seemed to elude the model, highlighting its inability to discern and adapt to the intricacies of such cyclical variations (4). This contrast is noteworthy as the less periodic M4 dataset showcased a more favorable performance in analyzing specific trends of at least parts of the data.

One discernible observation was the model's efficacy in capturing the general trend in transitioning from the training to the test phase in the M5 dataset (5). This ability to grasp overarching trends indicates a certain level of adaptability in understanding the broader patterns within the data. However, this proficiency came at the cost of neglecting the finer periodic trends that characterize the M5 dataset.

A difference emerged when comparing the confidence bounds between the M4 and M5 results. The M5 dataset exhibited smaller confidence bounds, indicative of a more conservative approach in the model's predictions or better confidence in its model's training and testing means. This conservatism could be attributed to the model's inclination towards generalizing the data patterns at the expense of accommodating the pronounced periodicity in the M5 dataset.

Further scrutiny of the M5 data revealed variations aligning with relative outliers, particularly at the maxima or minima of the apparent "periods" in the dataset. These variations might correspond to critical points in the periodic cycles. The Change-Point detection model, however, seemed to overlook periodic nuances, emphasizing a tendency toward a more generalized representation of the data, both for the M4 and M5 dataset results.

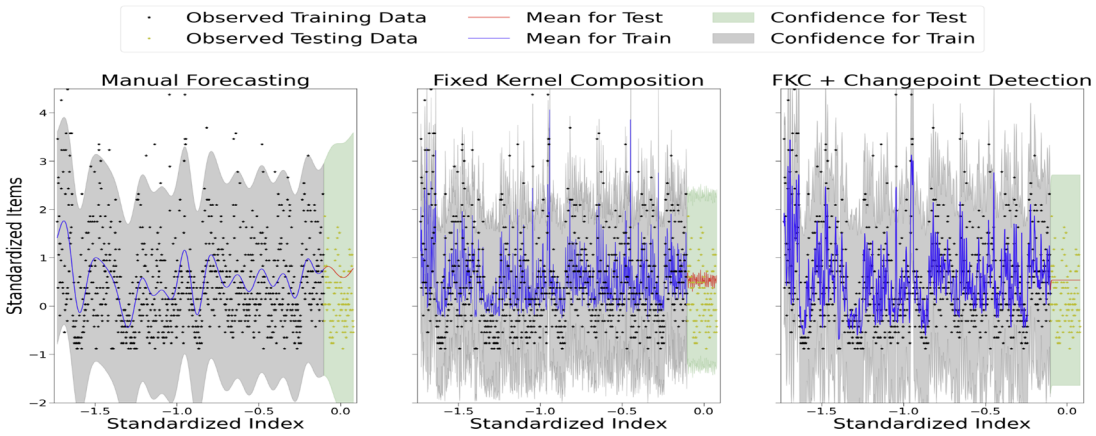

Figure5. This is the overview of one of the rows of the M5 dataset of sales data from a store in California. The figures shows the training (blue) and testing (red) means along with the confidence bounds of the models.

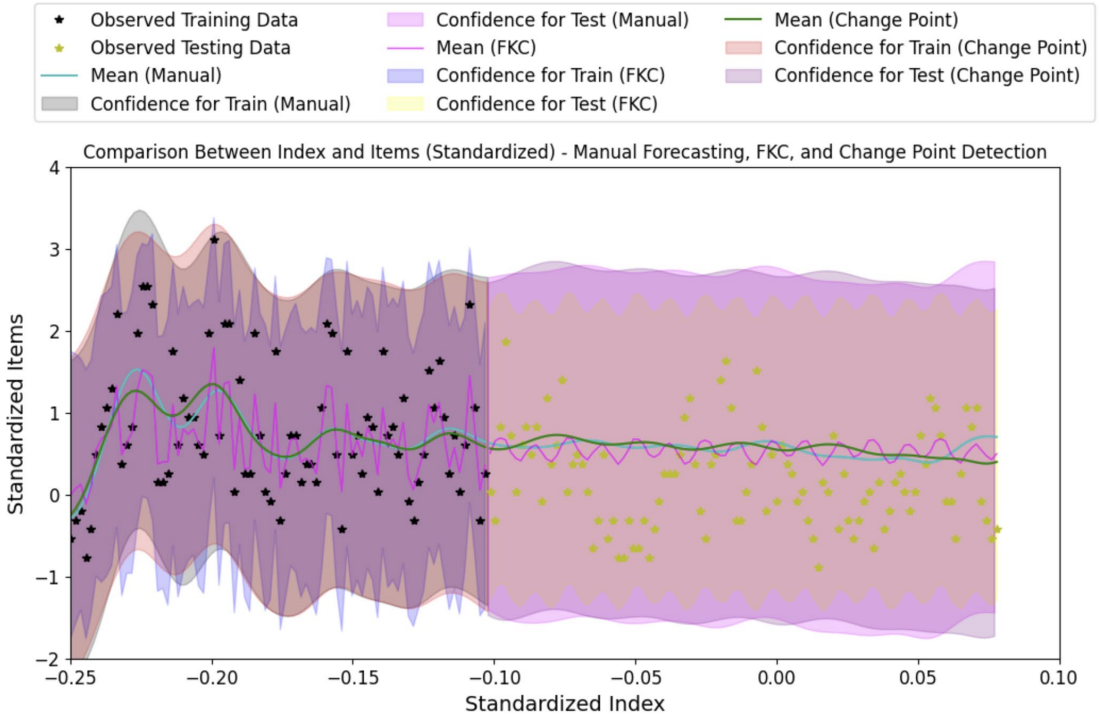

Figure6. This is the testing portion of the M5 dataset, showing the testing means calculated by each model, along with the test data and the confidence bounds of each model

In our comprehensive analysis of three distinct models applied to both the M4 and M5 datasets, we reviewed their numerical results across four crucial measurement factors of MAE, CRPS, LL (or negative-LL), and CPU time. The models in focus were the manual-fitting mode, the FKC model, and the change point model.

Beginning with the M4 dataset, the manual-fitting model demonstrated superior performance by achieving the best mean and standard deviation combination across the MAE, LL, and CPU run time metrics. However, regarding the CRPS result, the change point model outperformed the others. Notably, change point

consistently secured the second position across all other c, approaching the performance of the best model in each metric.

These findings suggest that the manual forecasting model, particularly the variant selected through a meticulous process involving specific kernels, was well optimized for the M4 dataset. Simultaneously, the consistently strong performance of change point across all metrics implies that, for datasets characterized by fewer periodic trends and more extended-scale patterns, change point emerges as an optimal model. This is especially true when considering the time investment required for designing and implementing a custom-made manual-fitting model, which demands substantial development efforts.

Table2. M5 Numerical Results

	Manual	**FKC**	**FKC + CP**
MAE	0.6186 ± 0.0256	$\mathbf{0.5854 \pm 0.0376}$	0.5976 ± 0.0393
CRPS	0.419 ± 0.00587	$\mathbf{0.4184 \pm 0.0172}$	0.4232 ± 0.0165
NLL	1.2588 ± 0.00594	0.9872 ± 0.04	$\mathbf{0.9868 \pm 0.0364}$
Average CPU Time Per Iteration	$\mathbf{2.06 \pm 1.06}$	3.36 ± 1.20	2.45 ± 0.07

Turning our attention to the M5 dataset, a different pattern emerges. FKC takes the lead in terms of MAE and CRPS (comparing mean and standard deviations once again), showcasing its effectiveness in capturing the complexities of the dataset. Change point excels for the Negative Log Likelihood. Unsurprisingly, manual forecasting outshines the others regarding CPU time, as less computational work is needed. In this dataset, FKC is the most successful, closely followed by change point, while manual forecasting lags noticeably behind.

These results highlight the strength of FKC, despite its relatively slower run time due to its fully automated modeling system. Its ability to yield the best results in the periodic M5 dataset, characterized by numerous localized and general pattern shifts, underscores its effectiveness. Change point, with its adept performance, secures the second position, particularly given the dataset's periodicity and evident general patterns in the data. Finally, despite exhaustive efforts and kernel combinations, manual forecasting falls short of replicating the high-performing results achieved in the M4 dataset when applied to the M5 dataset.

Ultimately, change point emerges as a robust performer when confronted with broader data parameters and certain periodic trends. On the other hand, FKC maintains consistent good performance across diverse data types especially for more periodic data, albeit at the cost of extended run times for FKC models. The manual forecasting approach introduces an element of user-dependent variability. However, in our specific case, we successfully created manual forecasting models that, at the very least, matched the performance levels exhibited by both the FKC and change point models.

Our experiments generally ran successfully in the M4 dataset, while exhibiting errors and flaws in the M5 dataset. For example, the confidence bounds demonstrated the same periodic trend in fixed kernel composition, hence producing a questionable forecast. This may be caused by noisy data in conjunction with a suboptimal kernel composition and inadequate training of hyperparameters. Then, this leads to an inaccurate kernel composition that cannot precisely capture the trend and results in erroneous confidence bounds. The time series selected here are visibly noisy and appear abundant amount of outliers, which may disrupt the data's preexisting pattern, hindering the model's performance to capture the underlying trend. This may indicate insufficient and ineffective data preprocessing before we use them for training, especially in removing outliers and adjusting noise levels.

Related Work

Attempts at creating kernel compositions for GPs are not new. For time series problems, Corani et al. used the fixed kernel composition we compared with but emphasized the use of hierarchical priors for the hyperparameters. For the class of problems they benchmark against, they show that performance improves with this setup.

Structure discovery with GPs has been an active research area over the last decade (Duvenaud et al., 2013), (Mansinghka, 2023). (Duvenaud et al., 2013) uses a greedy search algorithm, while (Mansinghka, 2023) introduced a new structure learning algorithm based on sequential Monte Carlo sampling.

Change point detection for time series is also an active area of research (Bosc et al., 2003) has already implemented change point detection for the automatic analysis of subtle changes within MRI scans, and it turns out to be much less error-prone than a manual approach by experts. (Dehning et al., 2020) used change point detection to infer the effectiveness of government interventions in COVID-19 by evaluating the forecasted value with the actual reported data. More recently, (Smejkalová et al., 2023) adopted change point detection to analyze many short time series in waste management.

Conclusion

We compared change point detection and composite kernels with manually selected kernels and showed competitive results in terms of accuracy and runtime on 2 open-source time series data. As a relatively new model, change point detection already demonstrates its practical use in automatic time series forecasting and displays potential for future development. Future directions could include more robust detection of longer periods of change windows and developing benchmark datasets for fairer and more objective experiments with change point detection in forecasting.

Acknowledgments

The authors would like to thank Jonathan P. Chen for research mentorship and revisions, Jonathan C. Chen for guidance on writing, Kavinayan Sivakumar for providing guidance on general machine learning content and concepts, Neeraj Pradhan and Zhen Cao for the implementation of the GPytorch kernels, and the National High School Journal of Science Research Internship program for the research opportunity.

References

Aminikhanghahi, S., & Cook, D. J. (2016). A survey of methods for time series change point detection. Knowledge and Information Systems, 51(2), 339–367. https://doi.org/10.1007/s10115-016-0987-z

Armstrong, J. S. (2001). Principles of Forecasting: A Handbook for Researchers and Practitioners,.

Bergstra, J., Bardenet, R., Kégl, B., & Bengio, Y. (2011, December). Algorithms for Hyper-Parameter Optimization.

Bosc, M., Heitz, F., Armspach, J.-P., Namer, I., Gounot, D., & Rumbach, L. (2003). Automatic change detection in multimodal serial MRI: application to multiple sclerosis lesion evolution. NeuroImage, 20(2), 643–656. https://doi.org/10.1016/S1053-8119(03)00406-3

Burg, G. J. J. van den, & Williams, C. K. I. (2020). An Evaluation of Change Point Detection Algorithms. In arXiv.org. https://arxiv.org/abs/2003.06222

Carl Edward Rasmussen, C. K. I. W. (2005). Introduction. In Gaussian Processes for Machine Learning. The MIT Press. http://dx.doi.org/10.7551/mitpress/3206.003.0004

Chen, J. P., Pradhan, N., & Cao, Z. (n.d.). Autoforecasting library. In Computer Software. https://github.com/jpchen/autoforecaster

Corani, G., Benavoli, A., & Zaffalon, M. (2021). Time Series Forecasting with Gaussian Processes Needs Priors. In Lecture Notes in Computer Science (pp. 103–117). Springer International Publishing. https://doi.org/10.1007/978-3-030-86514-6_7

Dehning, J., Zierenberg, J., Spitzner, F. P., Wibral, M., Neto, J. P., Wilczek, M., & Priesemann, V. (2020). Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. Science, 369(6500), eabb9789. https://doi.org/10.1126/science.abb9789

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., & Ghahramani, Z. (2013). Structure Discovery in Nonparametric Regression through Compositional Kernel Search.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., & Wilson, A. G. (2018). GPyTorch: Blackbox Matrix-Matrix

Gaussian Process Inference with GPU Acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), Advances in Neural Information Processing Systems (Vol. 31). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/27e8e17134dd7083b050476733207ea1-Paper.pdf

Garnett, R., Osborne, M. A., & Roberts, S. J. (2009, June). Sequential Bayesian prediction in the presence of changepoints. Proceedings of the 26th Annual International Conference on Machine Learning. http://dx.doi.org/10.1145/1553374.1553418

Hyndman, R. J., & Athanasopoulos, G. (2013). Forecasting: Principles and Practice. Otexts.

Kaggle Competitions. (n.d.). In Kaggle. https://www.kaggle.com/competitions

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting, 36(1), 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). The M5 competition: Background, organization, and implementation. International Journal of Forecasting, 38(4), 1325–1336. https://doi.org/10.1016/j.ijforecast.2021.07.007

Mansinghka, B. J. P., Feras A. Saad. (2023). Sequential Monte Carlo Learning for Time Series Structure Discovery.

Nesreen K. Ahmed, N. E. G., Amir F. Atiya, & El-Shishiny, H. (2010). An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econometric Reviews, 29(5–6), 594–621. https://doi.org/10.1080/07474938.2010.481556

Rasmussen, C. Edward. (2006). Gaussian Processes for Machine Learning.
Smejkalová, V., Šomplák, R., Rosecký, M., & Šramková, K. (2023). Machine Learning Method for Changepoint Detection in Short Time Series Data. Machine Learning and Knowledge Extraction, 5(4), 1407–1432. https://doi.org/10.3390/make5040071

Vayatis;, C. T. L. O. N. (2020). Selective review of offline change point detection methods.

Wang, C., Baratchi, M., Bäck, T., Hoos, H. H., Limmer, S., & Olhofer, M. (2022, June). Towards Time-Series Feature Engineering in Automated Machine Learning for Multi-Step-Ahead Forecasting. ITISE 2022. http://dx.doi.org/10.3390/engproc2022018017

(2020). Machine Learning for Time Series Forecasting with Python®, 167–196. https://doi.org/10.1002/9781119682394.ch6