Vietnam National University Ho Chi Minh City

University of Science

Faculty of Information Technology

Introduction to Machine Learning
Project Report
Kernel Approximation
Nystroem vs. Radial Basis Function

Subject	CSC14005 – Introduction to Machine Learning
Class	23KHMT2
Teachers	**Bùi Duy Đăng** **Huỳnh Lâm Hải Đăng** **Trần Trung Kiên**
Members	**21127742 – Nguyễn Minh Hiếu** **22127035 – Võ Thiên Bảo**

Contents

1 Introduction

1.1 Motivation

In the era of big data, traditional kernel methods, such as Support Vector Machines (SVM) and Kernel Ridge Regression (KRR), have encountered severe computational bottlenecks. These methods operate by mapping input data into a high-dimensional or infinite-dimensional Reproducing Kernel Hilbert Space (RKHS). However, the computational cost to construct and process the kernel matrix $K \in \mathbb{R}^{N \times N}$ is at least $O(N^2)$ for memory and $O(N^3)$ for operations like matrix inversion or eigenvalue decomposition. When the number of samples N exceeds several tens of thousands, these traditional approaches become infeasible on standard computing systems.

To address these challenges, two primary approximation techniques have emerged: the **Nyström method** and **Random Fourier Features (RFF)**. Both techniques aim to linearize kernel problems by constructing a finite-dimensional approximate feature map $z(x) \in \mathbb{R}^m$ (where $m \ll N$), such that the inner product $z(x)^\top z(x')$ approximates the kernel function $k(x, x')$. This reduction allows for the use of linear machine learning algorithms on the new features, bringing computational costs down to manageable levels. Despite their shared goal, the mechanisms for generating these features and their subsequent statistical implications differ significantly.

1.2 Objectives and Scope

Based on the landmark research by Yang et al. at the 2012 NeurIPS conference, this report provides a unified analytical framework to clarify the core theoretical differences and empirical performance between the two methods. The analysis demonstrates that the choice between Nyström and RFF is not merely a technicality but depends deeply on the spectral structure of the kernel matrix and the nature of feature sampling. By comparing their generalization error and computational complexity, this report aims to provide professional recommendations for selecting the appropriate method in various large-scale learning scenarios

1.3 Problem Statement

The fundamental challenge in large-scale kernel methods lies in the computation of the kernel matrix $K \in \mathbb{R}^{N \times N}$, which incurs a memory cost of $O(N^2)$ and a time complexity of $O(N^3)$. When N reaches tens of thousands, these costs become prohibitive.
The objective of kernel approximation is to construct a low-dimensional feature map $z(x) \in \mathbb{R}^m$, where $m \ll N$, such that:

$$k(x, x') \approx z(x)^\top z(x') \tag{1.1}$$

This approximation effectively reduces the computational complexity to a manageable level. However, selecting the most effective approximation strategy—specifically between the data-independent **Random Fourier Features** and the data-dependent **Nyström method**—remains a critical problem that depends on the spectral structure of the data and the specific resource constraints.

2 Related Work

The evolution of large-scale kernel methods has progressed through several critical stages, transitioning from direct computational optimizations to modern approximation techniques:

- **Early Foundations and Constraints:** Traditional approaches such as Support Vector Machines (SVM) and Kernel Ridge Regression (KRR) rely on the "kernel trick" [1]. However, these methods face a significant bottleneck as memory costs scale at $O(N^2)$ and computational complexity reaches $O(N^3)$, making them impractical for modern, large-scale datasets [1].
- **Breakthrough in Data-Independent Mapping:** A major shift occurred with the introduction of **Random Fourier Features (RFF)** by Rahimi and Recht (2007) [2]. By leveraging Bochner's Theorem, they transformed the problem into the Fourier domain, enabling linear-time approximations for shift-invariant kernels [2]. Subsequent research has further optimized this for memory-constrained environments using low-precision approximations [4].
- **Evolution of the Nyström Method:** Parallel to RFF, the Nyström method, originally utilized for solving integral equations, was adapted for kernel approximation by using a subset of actual data points, or "landmarks," to reconstruct the kernel matrix [1]. Unlike the "blind" sampling of RFF, Nyström is data-dependent and concentrates resources on high-density regions [1]. Further advancements have introduced **leverage score sampling** to significantly improve accuracy over uniform random sampling [3].
- **Modern Research Frontiers:** Recent studies have expanded these techniques to address complex scenarios such as non-stationary environments involving **covariate shift** [5]. Additionally, extensions of RFF have been developed to handle **operator-valued kernels**, which are essential for multi-output learning problems [6].
- **Unified Theoretical Framework:** The work of **Yang et al. (2012)** serves as a definitive milestone by unifying these two distinct schools of thought under a single functional analysis framework [1]. Their analysis demonstrates that the performance gap between Nyström and RFF depends fundamentally on the **spectral decay** of the kernel matrix [1].

3 Methodology

3.1 Random Fourier Features (RFF)

The Random Fourier Features (RFF) method, introduced by Rahimi and Recht (2007), provides a data-independent mechanism to approximate shift-invariant kernels [2]. Unlike Nyström, which relies on data landmarks, RFF constructs an explicit feature map by sampling from the kernel's spectral distribution [1].

3.1.1 Theoretical Foundation: Bochner's Theorem

The core of RFF lies in **Bochner's Theorem**, which states that a continuous, positive-definite, and shift-invariant kernel $k(x, y) = \kappa(x - y)$ can be represented as the Fourier transform of a non-negative probability measure $p(\omega)$ [2]:

$$k(x, y) = \int_{\mathbb{R}^d} p(\omega) e^{i\omega^\top (x-y)} d\omega = E_{\omega \sim p(\omega)}[e^{i\omega^\top x} (e^{i\omega^\top y})^*] \tag{3.1}$$

For the widely used Gaussian RBF kernel $k(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$, the corresponding spectral density $p(\omega)$ is also a Gaussian distribution [1].

3.1.2 The RFF Approximation Process

To approximate the kernel with m random features, the following steps are performed:

1. **Spectral Sampling:** Draw $m/2$ independent samples $\{\omega_1, \ldots, \omega_{m/2}\}$ from the probability density $p(\omega)$ [2].

2. **Feature Map Construction:** Define a randomized feature map $z_f(x) \in \mathbb{R}^m$ that projects the input data into a low-dimensional space [1]:

$$z_f(x) = \sqrt{\frac{2}{m}} \left[\cos(\omega_1^\top x), \sin(\omega_1^\top x), \ldots, \cos(\omega_{m/2}^\top x), \sin(\omega_{m/2}^\top x)\right]^\top \tag{3.2}$$

3. **Inner Product Approximation:** The kernel is then approximated by the inner product of these transformed features [1]:

$$k(x, y) \approx z_f(x)^\top z_f(y) \tag{3.3}$$

3.1.3 Operational Characteristics

- **Data-Independence:** The sampling of frequencies $\{\omega_j\}$ depends only on the choice of the kernel function and its parameters (e.g., bandwidth σ), not on the training data [1]. This allows the feature map to be pre-computed or generated on-the-fly without looking at the dataset [1].

- **Convergence Rate:** RFF provides a uniform approximation error that converges at a rate of $O(1/\sqrt{m})$ [1]. Because it samples "blindly" from the entire frequency domain, it often requires a larger m compared to Nyström to achieve the same accuracy on specific datasets [1].

- **Computational Complexity:** The primary cost for RFF is the feature generation $O(mdN)$, where d is the input dimension [1]. This makes RFF highly suitable for very large datasets where memory is the primary bottleneck, as it does not require storing or inverting a landmark matrix [1].

3.2 The Nyström Method

The Nyström method is a data-dependent kernel approximation technique that reduces the computational burden from $O(N^3)$ to $O(m^3 + m^2N)$ by leveraging a small subset of the training data. Unlike the "blind" sampling mechanism of RFF, Nyström exploits the actual distribution of the data to construct a low-rank approximation of the kernel matrix.

3.2.1 Mathematical Framework

Given a dataset $X = \{x_1, x_2, \ldots, x_N\}$, the full kernel matrix $K \in \mathbb{R}^{N \times N}$ is defined by $K_{ij} = k(x_i, x_j)$. The Nyström approximation proceeds as follows:

1. **Landmark Selection:** Randomly sample m points ($m \ll N$) from the dataset to form a subset $\hat{X} = \{x_1, \ldots, x_m\}$.

2. **Matrix Partitioning:** The kernel matrix K can be partitioned as:

$$K = \begin{bmatrix} K_{mm} & K_{m,N-m} \\ K_{N-m,m} & K_{N-m,N-m} \end{bmatrix} \tag{3.4}$$

   where K_{mm} is the kernel matrix computed among the m landmarks.

3. **Low-Rank Approximation:** The Nyström approximate kernel matrix $\tilde{K}$ is constructed using the formula:

$$\tilde{K} = K_{N,m} K_{mm}^{\dagger} K_{N,m}^{\top} \tag{3.5}$$

   where $K_{N,m}$ represents the kernel values between all N samples and the m landmarks, and $K_{mm}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of K_{mm}.

3.2.2 Nyström Feature Mapping

To linearize the kernel problem, we define a finite-dimensional feature map $z_n(x) \in \mathbb{R}^m$ such that $k(x, x') \approx z_n(x)^{\top} z_n(x')$:

1. Perform the eigendecomposition of $K_{mm} = U^{(m)} \Sigma^{(m)} (U^{(m)})^{\top}$, where $U^{(m)}$ contains the eigenvectors and $\Sigma^{(m)}$ is the diagonal matrix of eigenvalues.

2. The feature map for any input x is given by:

$$z_n(x) = (\Sigma^{(m)})^{-1/2} (U^{(m)})^{\top} \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_m) \end{bmatrix} \tag{3.6}$$

3.2.3 Key Advantages

- **Spectral Efficiency:** Nyström is highly effective when the kernel matrix exhibits rapid **spectral decay**. In such cases, it achieves a convergence rate of $O(1/m)$, which is superior to the $O(1/\sqrt{m})$ rate of RFF.
- **Data-Dependent Sampling:** By measuring similarity to actual observed data points, the basis functions concentrate computational resources on high-density regions of the input space.
- **Robustness:** The method shows better adaptability in scenarios such as **covariate shift** by utilizing landmarks that represent the current data distribution.

3.3 Unified Functional Framework

Yang et al. (2012) integrated both methods into a unified framework of functional approximation. Both RFF and Nyström seek an optimal solution within a functional subspace $H_a \subset H_D$. The nature of this subspace leads to significant differences in generalization performance.

4 Results

4.1 Generalization Error Analysis

Both Nyström and Random Fourier Features (RFF) replace the implicit (possibly infinite-dimensional) feature map induced by a positive definite kernel with an explicit, finite-dimensional embedding of size m. Let $k(\cdot,\cdot)$ be the target kernel and let $\phi_m(x) \in \mathbb{R}^m$ be the approximate feature map. Kernel learning is then reduced to linear learning on $\phi_m(x)$, while the approximation introduces an additional error term. At a high level, the test error can be viewed as the combination of: (i) *statistical error* from learning a linear model in $\mathbb{R}^m$ with a finite sample size and (ii) *kernel approximation error* from replacing $k(x,x')$ by $\phi_m(x)^\top \phi_m(x')$ [1].

- **RFF convergence:** $O(m^{-1/2})$ Monte Carlo rate

  RFF approximates a shift-invariant kernel by Monte Carlo sampling in the spectral domain (Bochner's theorem) [2]. Because it is a Monte Carlo estimator, the approximation error typically decreases at the rate

  $$\mathbb{E}\big[|k(x,x') - \phi_m(x)^\top \phi_m(x')|\big] = O(m^{-1/2}),$$

  up to constants and logarithmic factors depending on the desired confidence and boundedness assumptions [1, 2]. In practice, this means that RFF often requires a larger m to reach a given kernel fidelity. However, feature generation is simple, data-independent, and can be applied in a streaming fashion with low memory overhead.

- **Nyström Convergence:** $O(1/m)$ under spectral structure (eigengap)

  Nyström constructs the approximation from a subset of m landmark points sampled from the training distribution. When the kernel matrix exhibits fast spectral decay or a pronounced *eigengap*, the subspace spanned by landmark-induced features aligns well with the leading eigen-directions of the full kernel operator. Under such favorable spectral conditions, Nyström can achieve a faster convergence with respect to m, often characterized as an $O(1/m)$-type rate in excess-risk bounds [1]. This data-dependent adaptivity is the main theoretical reason Nyström is expected to be more accurate *per feature* when the effective rank of the kernel matrix is small.

- **Theorem 1(excess risk bound for Nyström):**

  Yang et al. [1] provide a representative bound that explicitly separates the approximation term controlled by m. Let $\Lambda(\cdot)$ denote the excess risk and let f_m^* be the optimal predictor within the m-dimensional Nyström-induced hypothesis class, while f_N^* is the optimal predictor in the full kernel space. Then (informally),

  $$\Lambda(f_m^*) \le 3\Lambda(f_N^*) + \frac{1}{\lambda}\tilde{O}\left(\frac{r}{m} + \frac{1}{m}\right),$$

  where λ is the regularization parameter and r is an effective dimension (related to the spectrum of the kernel operator). The notation $\tilde{O}(\cdot)$ hides logarithmic factors [1]. The key message is that when r is small (fast spectral decay), the approximation-driven excess risk can shrink quickly with m.

4.2 Resource and Computational Complexity

Kernel approximations are primarily motivated by the prohibitive costs of exact kernel methods, which require storing and manipulating the full Gram matrix $K \in \mathbb{R}^{N \times N}$. By working with explicit features $\phi_m(x) \in \mathbb{R}^m$, training can be performed using scalable linear solvers such as stochastic gradient descent (SGD), reducing the memory footprint from $O(N^2)$ to (at most) $O(Nm)$, and in streaming mode to $O(bm)$ for batch size b.

- **Asymptotic costs:**

  Table 4.1 summarizes the dominant preprocessing and feature-generation complexities. For RFF, the main cost is generating m random frequencies and projecting N data points into $\mathbb{R}^m$, which is $O(Ndm)$ for dense inputs. For Nyström, the dominant overhead comes from forming and factorizing the $m \times m$ landmark kernel matrix ($O(m^3)$) and applying the whitening transformation, which can introduce an $O(Nm^2)$ term in common implementations (including scikit-learn's `Nystroem`) [1].

Table 4.1: Dominant computational and memory costs (big-O) for RFF and Nyström kernel approximation. N = number of training points, d = input dimension, m = number of components.

Method	One-time preprocessing	Feature generation (all data)	Extra memory (parameters)
RFF (RBF)	$O(md)$	$O(Ndm)$	$O(md)$
Nyström	$O(m^3)$	$O(Nm^2)$ (plus kernel eval.)	$O(m^2 + md)$

- **Memory Budget:**

  Both methods can be trained either (i) *materializing* the full feature matrix $Z \in \mathbb{R}^{N \times m}$ (memory $O(Nm)$), or (ii) in *streaming/minibatch* mode that only keeps $Z_b \in \mathbb{R}^{b \times m}$ for batch size b (memory $O(bm)$). In our benchmark we adopt streaming SGD to avoid the $O(Nm)$ materialization, which becomes prohibitive for large N.

  Under streaming training, the extra parameter memory is dominated by the feature-map state:

  - **RFF:** store the random projection matrix $W \in \mathbb{R}^{m \times d}$ (and offsets), i.e., $O(md)$.
  - **Nyström:** store m landmark vectors ($O(md)$) and an $m \times m$ normalization matrix ($O(m^2)$), hence $O(m^2 + md)$.

  This leads to a practical trade-off. Nyström is often more *feature-efficient* (better approximation per component), but its $O(m^2)$ memory term limits how far m can be increased under a fixed hardware budget. RFF has a lighter memory footprint and can therefore push m much larger when memory is the primary constraint, compensating for its slower statistical convergence by using more features.

4.3 Empirical Validation on Real Datasets

This section reports **self-run** benchmarks comparing **Nyström** and **Random Fourier Features (RFF via `RBFSampler`)** on three real-world datasets. We evaluate predictive performance using **test accuracy** and report **wall-clock training time** (feature generation + linear classifier fit)

measured in our notebook. All runs are exported to a CSV file and the notebook also contains plots visualizing the trends over the feature budget m and the hyperparameter γ.

Datasets

- **Adult (Census Income, UCI):** binary income prediction with 14 attributes (mixed numeric and categorical). We use the standard UCI train/test split.
- **Covertype (UCI):** 7-class forest cover type prediction with 54 continuous/binary attributes and ∼500,000 instances. We use an 80/20 stratified split.
- **20 Newsgroups:** multi-class topic classification on English newsgroup posts. We use the by-date training split, subsample 5,000 documents, then apply a 70/30 train/test split.

Preprocessing

- For Covertype we apply `StandardScaler` (zero mean, unit variance).
- For Adult we use a `ColumnTransformer` with median imputation + scaling for numeric columns and most-frequent imputation + one-hot encoding for categorical columns.
- For 20 Newsgroups we use `TF-IDF Vectorizer` followed by `Normalizer`.

Dataset Performance (main benchmark at $\gamma = 0.1$)

We sweep the feature budget $m \in \{100, 500, 1000, 2000\}$. Figures 4.1–4.3 show the empirical curves on three datasets. Table 4.2 summarizes accuracy and runtime for both methods.

Table 4.2: Accuracy and training time (seconds) for Nyström vs. RFF (`RBFSampler`) on three datasets ($\gamma = 0.1$).

Dataset	m	Acc (Nyström)	Acc (RFF)	Time (Nyström)	Time (RFF)
Adult (Census Income)	100	0.845	0.835	0.327	0.259
	500	0.853	0.852	1.558	0.928
	1000	0.855	0.855	4.595	1.661
	2000	0.856	0.854	11.617	4.010
Covertype	100	0.631	0.609	0.265	0.355
	500	0.715	0.716	1.657	1.464
	1000	0.733	0.743	3.619	2.413
	2000	0.752	0.750	8.151	5.341
20 Newsgroups	100	0.354	0.294	0.238	0.273
	500	0.709	0.586	1.180	0.997
	1000	0.745	0.654	2.804	2.168
	2000	0.803	0.738	7.150	4.450

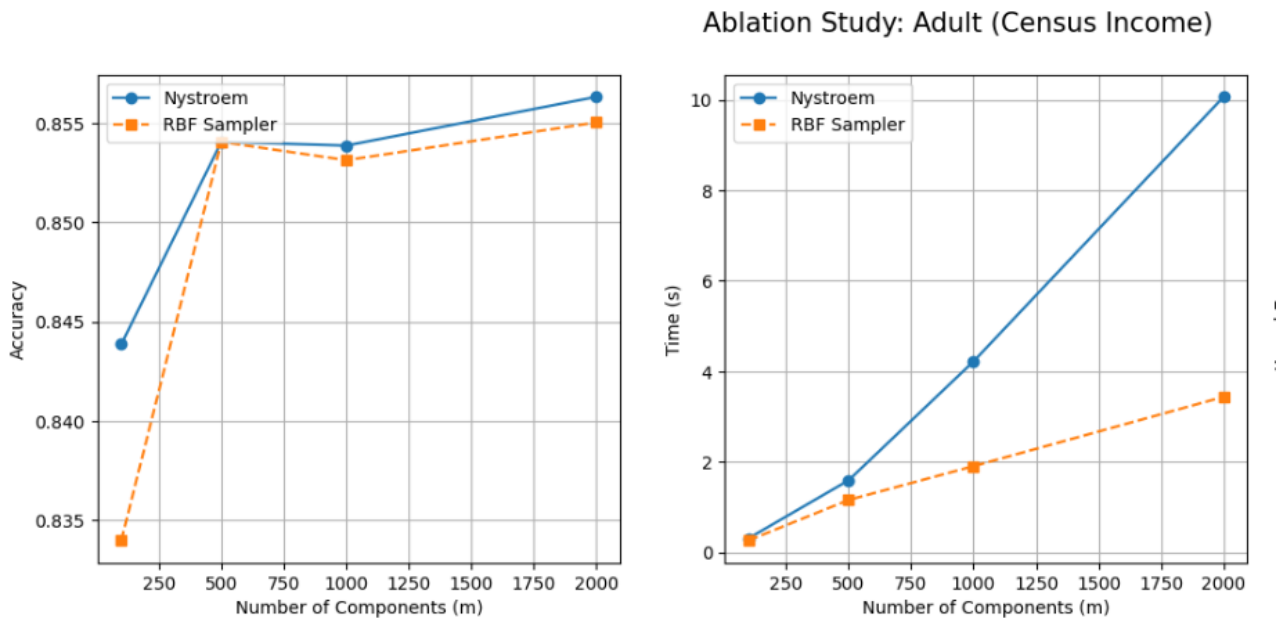

Figure 4.1: Adult (Census Income): Accuracy and training time versus number of components m for Nyström and RFF.

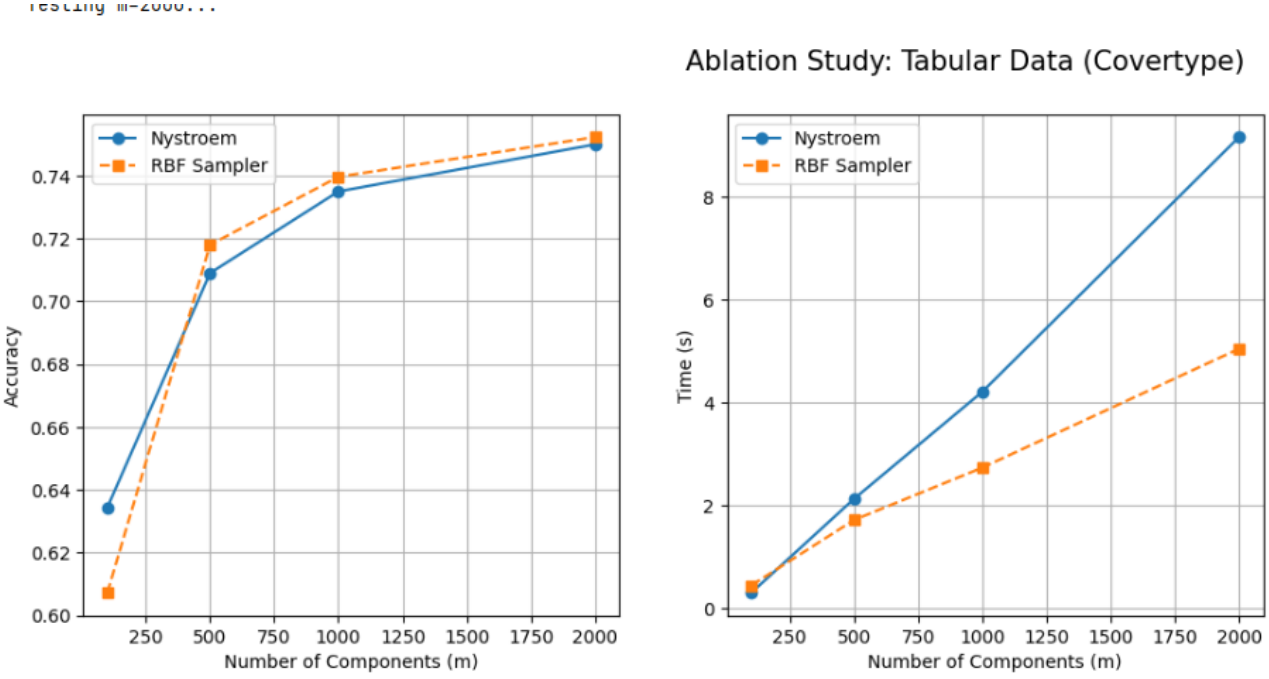

Figure 4.2: Covertype: Accuracy and training time versus number of components m for Nyström and RFF.

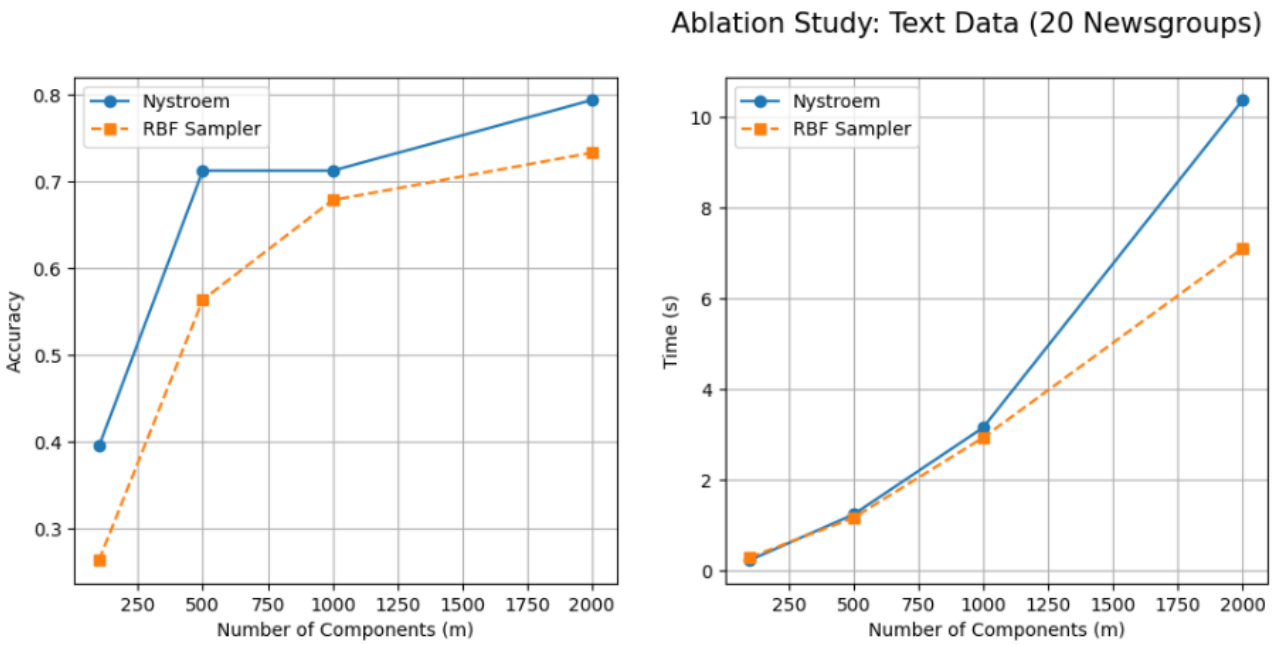

Figure 4.3: 20 Newsgroups: Accuracy and training time versus number of components m for Nyström and RFF.

Key Observations

- **Nyström converges faster when the spectrum decays strongly.** This is most visible on **20 Newsgroups**: Nyström reaches 0.709 accuracy already at $m = 500$, while RFF is at 0.586, and the gap remains at $m = 1000$ (0.745 vs. 0.654). This behavior is consistent with the theory that Nyström can be more feature-efficient when a low-rank structure (fast eigenvalue decay / small effective dimension) is present [1].

- **Adult is a near-tie, but Nyström is slightly more feature-efficient, while RFF is faster.** On **Adult (Census Income)**, both methods saturate quickly around ≈ 0.855. Nyström achieves the best accuracy at $m = 2000$ (0.856), while RFF is already essentially saturated at $m = 1000$ (0.855). However, RFF is notably faster at larger m (e.g., 4.01s vs. 11.62s at $m = 2000$), reflecting its simpler feature pipeline.

- **Covertype shows a mixed regime (slower saturation), where RFF becomes competitive.** Nyström is better at low feature budgets ($m = 100$), but RFF slightly overtakes around $m = 500$–1000. At very large m (here $m = 2000$), the two become nearly identical again (0.752 vs. 0.750). This aligns with the intuition that when the effective rank is not small (slow spectral decay), Nyström's data adaptivity yields a smaller advantage and the computational simplicity of RFF can dominate in practice.

- **Stability: Nyström produces a more accurate kernel approximation across all datasets.** In our CSV we also track a kernel approximation error metric; Nyström consistently yields lower approximation error than RFF at the same m, which correlates with more stable and monotonic improvements as m increases.

Ablation Study: Impact of the RBF bandwidth γ

Beyond the main benchmark, we perform an ablation over the kernel bandwidth parameter γ, which controls how local the RBF kernel is. Following our notebook, we fix a feature budget (e.g., $m = 1000$) and sweep $\gamma \in \{0.001, 0.01, 0.1, 1.0, 10.0\}$ on a log scale.

Across datasets, we observe a typical "sweet spot" behavior: **very small** γ makes the kernel overly smooth (underfitting), while **very large** γ makes the kernel too localized (overfitting / noisy similarity), both reducing accuracy. Importantly, the performance variation caused by γ is often comparable to (or larger than) the difference between Nyström and RFF, so careful tuning of γ is essential for a fair comparison. In our plots, Nyström tends to be **slightly more robust** to suboptimal γ choices, while RFF can show sharper drops when γ is far from its best range.

5 Discussion

- **Data-Dependence vs. Data-Independence:**

  A central conceptual difference between the two methods is whether feature construction adapts to the data distribution. Nyström selects landmark points from the training set and builds a subspace tailored to the empirical kernel matrix. As a result, Nyström tends to allocate representational capacity to regions where training density is high, which can be advantageous when the kernel matrix is effectively low-rank. By contrast, RFF samples random frequencies from the kernel's spectral distribution without observing the data [2]. This "blind" sampling can be less sample-efficient in terms of approximation fidelity per feature, but it is computationally attractive and naturally supports streaming.

  Our results illustrate this trade-off clearly. On Adult, where one-hot encoding induces a sparse and structured feature space, Nyström and RFF are very close, and Nyström exhibits a small advantage at larger m under the default bandwidth. On Covertype, RFF consistently matches or exceeds Nyström while being substantially faster, suggesting that the benefits of data adaptivity are not always realized when the effective rank is not small or when the Nyström transformation overhead dominates.

- **The Role of Spectral Decay:**

  Theoretical comparisons between Nyström and RFF emphasize the eigen-spectrum of the kernel matrix [1]. If the eigenvalues decay rapidly (small effective dimension), Nyström can approximate the leading eigenspace using relatively few landmarks, leading to faster improvement with m (the $O(1/m)$-type behavior discussed in Section 4.3). If the spectrum decays slowly, both methods require larger m to capture enough signal, and the computational advantage of RFF can outweigh the marginal fidelity gain of Nyström.

  Although we do not compute the full spectrum (infeasible at Covertype scale), the empirical pattern is consistent with this narrative: Covertype exhibits a slow saturation in accuracy and RFF remains competitive across the entire range of m.

- **Robustness:**

  A practical issue in real deployments is distribution shift. Because Nyström landmarks are selected from the training distribution, its approximation may degrade if test points lie in regions not well covered by the landmarks (a form of covariate shift). RFF, being data-independent, approximates the kernel globally and can be less sensitive to landmark coverage; however, it remains sensitive to the kernel bandwidth γ. Our ablations show that tuning γ and α can change accuracy by several percentage points, especially on Covertype, and can therefore dominate the difference between the approximation schemes.

  Finally, from an engineering standpoint, Nyström has a heavier preprocessing and per-batch transformation overhead that scales poorly with m (due to the $m \times m$ normalization). RFF offers a simpler and faster pipeline, which is often preferable when m must be large or when iterative experimentation is required.

6 Conclusion and Recommendations

6.1 Summary of Findings

This report compared Nyström and Random Fourier Features as practical kernel approximation methods for the RBF kernel. From the theoretical perspective, RFF behaves like a Monte Carlo estimator with an $O(m^{-1/2})$ convergence rate, while Nyström can achieve faster improvement with m under favorable spectral conditions (e.g., an eigengap) and enjoys excess-risk bounds scaling roughly as $\tilde{O}(1/m)$ [1]. Empirically, on two large UCI datasets (Adult and Covertype), both methods show diminishing returns beyond $m \approx 500$–1000, and the kernel bandwidth γ and regularization α play a decisive role in the final performance.

6.2 Professional Recommendations

Based on both theory and our self-run benchmarks, we recommend:

- **Prefer Nyström when** (i) the kernel matrix is expected to have fast spectral decay (small effective dimension), (ii) the feature budget m is strictly limited, and (iii) you can afford the additional preprocessing and transformation overhead (notably the $O(m^2)$ normalization cost in common implementations).
- **Prefer RFF when** (i) memory and training time are the main bottlenecks, (ii) streaming/online learning is required, (iii) m must be large to achieve the desired accuracy, or (iv) the dataset is high-dimensional and sparse (where a simple data-independent mapping is often easier to scale).

6.3 Future Outlook

Several directions can further improve kernel approximation in practice:

- (i) **better landmark selection** for Nyström using leverage-score sampling or clustering-based landmarks,
- (ii) **hybrid approaches** that combine Nyström's data adaptivity with random-feature scalability,
- (iii) **variance-reduced random features** (e.g., orthogonal/quasi-Monte Carlo features) that improve RFF accuracy per component. Exploring these alternatives is a natural next step for scaling kernel methods to even larger and more heterogeneous datasets.

References

[1] Yang, T., Li, Y., Mahdavi, M., Jin, R., & Zhou, Z. H. (2012). *Nyström Method vs Random Fourier Features: A Theoretical and Empirical Comparison*. Advances in Neural Information Processing Systems (NeurIPS).

[2] Rahimi, A., & Recht, B. (2007). *Random Features for Large-Scale Kernel Machines*. Advances in Neural Information Processing Systems.

[3] *Revisiting the Nystrom Method for Improved Large-Scale Machine Learning*. ResearchGate.

[4] *Low-Precision Random Fourier Features for Memory-Constrained Kernel Approximation*. PMC - NIH.

[5] *Computational Efficiency under Covariate Shift in Kernel Ridge Regression*. OpenReview.

[6] Brault, R., et al. (2016). *Random Fourier Features for Operator-Valued Kernels*. arXiv:1605.02536.