Abstract

Low-rank approximation is commonly used to scale kernel-based algorithms to large-scale applications containing as many as several million instances. We introduce a new family of algorithms based on mixtures of Nyström approximations, ensemble Nyström algorithms, that yield more accurate kernel approximations than the standard Nyström method. We present extensive empirical results on data sets containing up to 1M points demonstrating the improvement over the standard Nyström approximation. Finally, we present a stability bound for Kernel Ridge Regression based on the norm of the kernel approximation to help determine the impact of the kernel approximation on the generalization error.

1 Introduction

The size of large-scale learning problems commonly found in modern tasks in computer vision, natural language processing, and systems design can exceed several million. Scaling standard kernel-based algorithms such as support vector machines (SVMs), kernel ridge regression (KRR), kernel principal component analysis (KPCA) to such magnitudes is a serious issue since even storing the kernel matrix can be prohibitive at this scale. One common solution for dealing with such large-scale problems consists of a low-rank approximation of the kernel matrix [5]. Other variants of these approximation techniques based on the Nyström method have also been recently presented and shown to be applicable to large-scale problems [1,2,4,6]. In this work, we introduce a new family of algorithms based on mixtures of Nyström approximations, ensemble Nyström algorithms, and show extensive experimental results demonstrating the improved performance of these algorithms in comparison to the standard Nyström method. Moreover, these ensemble algorithms naturally fit into distributed computing architectures such that their computational cost is roughly the same as that of the standard Nyström method. Finally, instead of merely focusing on the quality of the low-rank approximation, we present a novel stability bound that directly analyzes the effect of low-rank approximation in the kernel matrix on the hypothesis generated by Kernel Ridge Regression, a widely used kernel-based learning algorithm.

2 Algorithm

The Nyström approximation of an SPSD matrix $K$ is based on a sample of $m \ll n$ columns of $K$. Let $C$ denote the $n \times m$ matrix formed by these columns and $W$ the $m \times m$ matrix consisting of the intersection of these $m$ columns with the corresponding $m$ rows of $K$. For a uniform sampling of
columns, the rank-$k$ Nyström approximation $\widetilde{K}$ of $K$ equals $\widetilde{K} = CW_k^+C^T = K$, where $k \leq m$, $W_k$ is the best $k$-rank approximation of $W$, and $W_k^+$ is the pseudo-inverse of $W_k$.

Our novel ensemble Nyström algorithm treats each Nyström approximation generated for a sample of $m$ columns as an expert and combines $p \geq 1$ such experts to derive an improved hypothesis. We assume a fixed kernel function $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ that can be used to generate the entries of a kernel matrix $K$. The learner receives a sample $S$ of $mp$ columns randomly selected from matrix $K$ uniformly without replacement. $S$ is decomposed into $p$ subsamples $S_1, \ldots, S_p$. Each subsample $S_r$, $r \in [1, p]$, contains $m$ columns and is used to define a rank-$k$ Nyström approximation $\widetilde{K}_r$. The learner further receives a sample $V$ of $s$ columns used to determine the weight $\mu_r \in \mathbb{R}$ attributed to each expert $\widetilde{K}_r$. Thus, the general form of the approximation of $K$ generated by the ensemble Nyström algorithm is $\widetilde{K}^{ens} = \sum_{r=1}^p \mu_r \widetilde{K}_r$.

The most straightforward choice for weights, the uniform method, consists of assigning equal weight to each expert, $\mu_r = 1/p$, $r \in [1, p]$. Another method, the exponential weight method, consists of measuring the reconstruction error $\hat{e}_r$ of each expert $\widetilde{K}_r$ over the validation sample $V$ and defining the mixture weight as $\mu_r = \exp(-\eta \hat{e}_r)/Z$, where $\eta > 0$ is a parameter and $Z$ a normalization factor ensuring that the vector $\mu = (\mu_1, \ldots, \mu_p)$ belongs to the simplex $\Delta$ of $\mathbb{R}^p$: $\Delta = \{\mu \in \mathbb{R}^p : \mu \geq 0 \land \sum_{r=1}^p \mu_r = 1\}$. Alternatively, we can use the sample $V$ to train the mixture weights $\mu_r$ to optimize a regression objective function. Hence, in the ridge regression method we perform the following optimization: $\min_{\mu} \lambda \|\mu\|^2_2 + \|\sum_{r=1}^p \mu_r \widetilde{K}_r^V - K_V\|_F^2$, where $\lambda > 0$ and $K_V$ denotes the matrix formed by the columns of the samples $S$ and $V$.

The complexity of the ensemble Nyström algorithm is $O(pm^3 + pmkn + C_\mu)$, where $C_\mu$ is the cost of computing the mixture weights, $\mu$, used to combine the $p$ Nyström approximations. In general, the cubic term dominates the complexity since the mixture weights can be computed in constant time for the uniform method, in $O(psn)$ for the exponential weight method, or in $O(p^3 + pms)$ for the ridge regression method. Furthermore, although the ensemble Nyström algorithm requires $p$ times more space and cpu cycles than the standard Nyström method, these additional requirements are quite reasonable in practice. The space requirement is still manageable for even large-scale applications given that $p$ is typically $O(1)$ and $m$ is usually a very small percentage of $n$ (see Section 3 for further details). In terms of cpu requirements, we note that our algorithm can be easily parallelized, as all $p$ experts can be computed simultaneously. Thus, given a cluster of $p$ machines, the run time of this algorithm is nearly equal to that of the standard Nyström algorithm with $m$ samples.

### 3 Experiments

In this section we present experimental results for the ensemble Nyström method using the datasets listed in Table 1. We first compare the performance of various methods for calculating the mixture weights ($\mu_r$), and then show the effectiveness of our technique on a large-scale dataset. Throughout these experiments, we measure accuracy of a low-rank approximation ($\widetilde{K}$) by calculating percent error in Frobenius, $\% \text{error} = \|K - \widetilde{K}\|_F / \|K\|_F \times 100$.

Our first set of experiments involved the first five datasets in Table 1. We fixed the reduced rank to $k = 50$, and set the number of sampled columns to $m = 3\% n$.\footnote{Similar results (not shown) were observed for other values of $k$ and $m$ as well.} For the exponential and the ridge regression variants, we sampled an additional set of $s = 20$ columns and used an additional 20 columns ($s'$) as a hold-out set for choosing the optimal values of $\eta$ and $\lambda$. The number of
Figure 1: Percent error in Frobenius norm for ensemble Nyström method using uniform ('uni'), exponential ('exp'), ridge ('ridge') and optimal ('optimal') mixture weights as well as the best ('best w.l.') and mean ('mean w.l.') performance of the $p$ individual approximations.

Next, we present an empirical study of the effectiveness of the ensemble Nyström method on the SIFT-1M dataset containing 1 million data points. As is common practice with large-scale datasets, we worked on a cluster of several machines for this dataset. To fairly compare these techniques, we measured the quality of various approximation under a fixed time constraint. We set $k = 50$ and $p = 10$, based on results from other studies, and set the sizes of the validation and hold-out sets to $s = 2$ and $s' = 2$, respectively. The results, presented in Figure 2, clearly show that the ensemble Nyström method is the most effective technique given a fixed amount of time. Further, although the space requirements are $p = 10$ times greater for ensemble Nyström in comparison to standard Nyström, they are still quite reasonable. For instance, when working with 1M points, the ensemble Nyström method with ridge regression weights only required storage of approximately 1% of the columns of $K$ to achieve a percent error of 10%.

4 Kernel Stability of KRR

In this section we begin to address the following question: how does the kernel approximation affect the performance of the associated learning algorithm? To do so, we analyze the impact of kernel approximation on KRR, providing a bound on the hypothesis directly in terms of the quality of the kernel approximation. We consider the standard supervised learning setting where the learning algorithm receives a sample of $m$ labeled points $S = (x_1, y_1), \ldots, (x_m, y_m) \in (X \times Y)^m$, where
Figure 2: Large-scale performance comparison with SIFT-1M dataset. Given fixed computational time, ensemble Nyström with ridge weights tends to outperform other techniques.

$X$ is the input space and $Y$ the set of regression labels, $Y = \mathbb{R}$ with $|y| \leq M$. Our bounds hold for any SPSD approximation, $K'$, of the original kernel matrix $K$.

The dual optimization problem solved by KRR [3] is $\max_{\alpha \in \mathbb{R}^m} \lambda \alpha^\top \alpha + \alpha^\top K \alpha - 2 \alpha^\top y$, where $\lambda = m \lambda_0 > 0$ is the ridge parameter. The problem admits the closed form solution $\alpha = (K + \lambda I)^{-1} y$.

We denote by $h$ the hypothesis returned by kernel ridge regression when using the exact kernel matrix, and present the following bound (proof omitted for space reasons).

**Proposition 1.** Let $h'$ denote the hypothesis returned by kernel ridge regression when using the approximate kernel matrix $K' \in \mathbb{R}^{m \times m}$. Furthermore, define $\kappa > 0$ such that $K(x, x) \leq \kappa$ and $K'(x, x) \leq \kappa$ for all $x \in X$. This condition is verified with $\kappa = 1$ for Gaussian kernels for example. Then the following inequality holds for all $x \in X$,

$$|h'(x) - h(x)| \leq \frac{\kappa M}{\lambda_0 m} \|K' - K\|_2.$$  \hspace{1cm} (1)

**5 Conclusion**

We presented a novel family of ensemble Nyström algorithms for accurate low-rank approximations for large-scale applications. The consistent and significant performance improvement across a number of different data sets, along with the fact that these algorithms can be easily parallelized, suggests that these algorithms can benefit a variety of applications where kernel methods are used. Interestingly, the algorithmic solution we have proposed for scaling these kernel learning algorithms to larger scales is itself derived from the machine learning idea of ensemble methods. We also gave the first theoretical analysis linking the quality of low-rank approximations to the change in hypothesis value for an actual learning algorithm. This analysis helps determine for Kernel Ridge Regression the degree of approximation that can be tolerated without deviating radically from the solution obtained using the correct kernel matrix. In the future, we expect that similar bounds can be generated for other kernel algorithms.

**References**


