Motivation, Method and Algorithms

Motivation:
- Reducing \( l \) in \( Z : \mathbb{R}^m \to \mathbb{R}^l \) can cut down the training time in the learning tasks.
- Data-independent vector representations in RKS are allowed to be refined further without losing much precision.

Method:
- Choose to approximate the kernel matrix \( K \).
- Refine RKS by a fast data-dependent subspace embedding (FDSE).

Algorithms:

Algorithm 1: TEFM-G

Input: \( \mathbf{X}^T = [x_i \in \mathbb{R}^m]_{i=1}^{n} \), shift-invariant kernel function \( k(\cdot, \cdot) \), scalars \( \{q, d, l\} \).
Output: \( \mathbf{G} \in \mathbb{R}^{n \times m} \) with each row vector constructed by method as shown in Eq. (2) or Eq. (3).

Algorithm 2: TEFM-S

Input: \( \mathbf{X}^T = [x_i \in \mathbb{R}^m]_{i=1}^{n} \), shift-invariant kernel function \( k(\cdot, \cdot) \), scalars \( \{d, l\} \).
Output: \( \mathbf{G} \in \mathbb{R}^{n \times m} \).

Random Feature Map

Random feature map as an alternative:
- Map explicitly \( Z : \mathbb{R}^m \to \mathbb{R}^l \) with \( k(x_i, x_j) = \langle Z(x_i), Z(x_j) \rangle \).
- e.g., Random Kitchen Sinks (RKS) for shift-invariant kernels.

Kernel Matrix Approximation:

\[
\text{Theorem 1. Suppose we have a kernel matrix } K \in \mathbb{R}^{n \times n} \text{ based on shift-invariant functions and get features } G \in \mathbb{R}^{n \times l} \text{ via Algorithm 1 or 2. Then the following inequality holds with limited failure probability:}
\]

\[
\| K - GG^T \|_2 \leq O(n/l).
\]

Remark. Training on \( O(l) \) features generated by our two algorithms is faster than \( O(l^2) \) features by RKS (\( \| K - GG^T \|_2 \leq O(n/\sqrt{l}) \)) with the possibility of the approximation error being incurred almost on the same scale.

Impact on learning task:
- KRR and SVM can be unified into the optimization problem,

\[
F(w_G) = \min_w \frac{1}{2} \| w \|^2_2 + \frac{1}{n} \sum_{i=1}^{n} \ell(w^T Z(x_i), y_i),
\]

where \( \ell(\cdot) \) is a loss function. Then, training on \( Z(x_i) = G_i \) gets \( F(w_G) \), and training on \( K \) \( (i.e., Z(x_i) = \Phi(x_i)) \) gets \( F(w_K) \).

\text{Theorem 2. Suppose we get a kernel matrix } K \in \mathbb{R}^{n \times n} \text{ by operating shift-invariant functions on data } X^T = [x_i]_{i=1}^n \text{, and a feature matrix } G \in \mathbb{R}^{n \times l} \text{ by algorithm 1 or 2. Then the following inequality holds with limited failure probability:}

\[
F(w_G) \leq F(w_K) + O(1/l).
\]

Remark. The additional regularized training error as shown in the big-O notation is essentially incurred by the kernel matrix approximation error.