Multiresolution Kernel Approximation for Gaussian Process Regression

Yi Ding, Risi Kondor, and Jonathan Eskreis-Winkler

NIPS 2017, Long Beach, CA, USA

Table 1: Regression Results with k to be ≤ pseudo-inputs/\texttt{compAct}, SMSE and MNLP

<table>
<thead>
<tr>
<th>Method</th>
<th>k</th>
<th>Full</th>
<th>SOR</th>
<th>FITC</th>
<th>PITC</th>
<th>MKA</th>
<th>\texttt{compAct}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolving</td>
<td>16</td>
<td>0.36</td>
<td>0.32</td>
<td>0.30</td>
<td>0.31</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>noise</td>
<td>0.17</td>
<td>0.93</td>
<td>0.94</td>
<td>0.93</td>
<td>0.94</td>
<td>0.46</td>
<td>0.18</td>
</tr>
<tr>
<td>wine</td>
<td>32</td>
<td>0.59</td>
<td>0.33</td>
<td>0.87</td>
<td>0.84</td>
<td>0.03</td>
<td>0.71</td>
</tr>
<tr>
<td>pagblock</td>
<td>32</td>
<td>0.44</td>
<td>1.10</td>
<td>0.86</td>
<td>0.57</td>
<td>0.81</td>
<td>0.73</td>
</tr>
<tr>
<td>compAct</td>
<td>32</td>
<td>0.58</td>
<td>0.66</td>
<td>0.88</td>
<td>0.13</td>
<td>0.91</td>
<td>0.08</td>
</tr>
<tr>
<td>SOR</td>
<td>64</td>
<td>0.15</td>
<td>0.73</td>
<td>0.65</td>
<td>0.19</td>
<td>0.70</td>
<td>0.17</td>
</tr>
<tr>
<td>FITC</td>
<td>70</td>
<td>0.70</td>
<td>0.72</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>MKA</td>
<td>75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Motivation

- Gaussian process regression generally does not scale to beyond a few thousands data points without applying some kernel approximation method.
- Most approximations focus on the high eigenvalue part of the spectrum of the kernel matrix, \( K \), which leads to bad performance when the length scale of the kernel is small.
- We introduce Multiresolution Kernel Approximation (MKA), the first true broadbanded kernel approximation method.
- MKA is memory efficient, and a direct method, which means it also makes it easy to approximate \( K^{-1} \) and \( \text{vec}(K) \).

Global Low Rank Methods

Mathematically:

\[
K(x, x) = \sum_{i=1}^{n} \sum_{j=1}^{m} K_i(x_i, x_j) u_i u_j^T
\]

Assuming that \( x_i \sim x_j \) is a subset of the original point set \( (x_i, y_i) \), amounts to an approximation of the form \( K = K_i K_j^T \) with \( i = (1, \ldots, n) \). The canonical choice of \( C = C' = W \), where \( W = K_i u_i \) and \( W' \) denotes the Moore-Penrose pseudo-inverse of the resulting approximation \( K = K_i W_i W_i^T \) is known as the Nyström approximation.

Local and Hierarchical Low Rank Methods

(a) In a simple blocked low rank approximation the diagonal blocks are dense (gray), whereas the off-diagonal blocks are low rank. (b) In an HODLR matrix the low-rank off-diagonal blocks form a hierarchical structure leading to a much more compact representation. (c) \( n^2 \) matrices are a refinement of this idea.

Experiments on Real Data

The direct way of applying MKA to speed up GP regression is simply using \( K \) to approximate the augmented kernel matrix \( K = (K + \sigma^2 I) \) and then inverting this approximation. Note that the resulting \( K^{-1} \) never needs to be evaluated fully. In the main text instead, the matrix-product vector \( \mathbf{f} \cdot \mathbf{y} \) can be computed in ‘matrix-free’ form.

Assuming that the testing set \( (x_i, y_i) \) is known at training time, however, we can take an alternative approach, whereby instead of approximating \( K \) or \( K^{-1} \), we compute the MKA approximation of the joint training/test kernel \( \tilde{K}_{\text{test}} = K_{\text{train}} - K_{\text{train}} \tilde{K}_{\text{test}} K_{\text{test}} \)

Writing \( \tilde{K}_{\text{test}} \) in blocked form

\[
\tilde{K}_{\text{test}} = L^{-1} \cdot \tilde{D} \cdot L^{-T}
\]

and taking the Schur complement of \( D \) now recovers an alternative approximation \( K_{\text{test}} = A - BD^{-1}A' \) which is consistent with the off-diagonal block \( K' \) leading to our final MKA-GP formula \( \tilde{f} = K_{\text{test}}^T \mathbf{y} \) where \( \tilde{f} \eqdef \tilde{K}_{\text{test}} \mathbf{y} \).

While we identified this is somewhat more involved than naively estimating \( \tilde{K} \), assuming \( n \ll n \), the cost of inverting \( D \) is negligible, and the overall serial complexity of the algorithm remains \( (n^2 \rho) \).

Simulation: 1D Toy Data

(a) This is a simple benchmark example. (b) Then, according to truth (black circles), prediction mean (solid line curves) and one standard deviation in prediction uncertainty (dashed line curves). (c) For different training set sizes \( |\mathcal{D}_{\text{train}}| \) and full test set \( \mathcal{D}_{\text{test}} \).

Across the range of pseudo-inputs/\texttt{compAct} size considered, MKA’s performance is robust to \( \texttt{compAct} \), while low-rank based methods’ performance changes rapidly, which shows MKA’s ability to achieve good regression results even with a crucial compression level.

Conclusions

- Whether a learning problem is low rank or not depends on the nature of the data rather than just the spectral properties of the kernel matrix.
- MKA allows fast direct calculations of the inverse of the kernel matrix and its determinant, which are almost always the computational bottlenecks in GP problems.

Figure 1: Snelson’s 1D example: ground truth (black circles); prediction mean (solid line curves); one standard deviation in prediction uncertainty (dashed line curves).