Scalable machine learning for massive datasets: Fast summation algorithms

Vikas Chandrakant Raykar
University of Maryland, CollegePark
{vikas}@cs.umd.edu

Abstract

Most state-of-the-art nonparametric machine learning algorithms have a computational complexity of either $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$, where N is the number of training examples. This has seriously restricted the use of massive data sets. The bottleneck computational primitive at the heart of various algorithms is the multiplication of a structured matrix with a vector, which we refer to as *matrix-vector product* (MVP) primitive. The goal of my thesis is to speedup up these MVP primitives by *fast approximate algorithms* that scale as $\mathcal{O}(N)$ and also provide *high accuracy guarantees*. I use ideas from computational physics, scientific computing, and computational geometry to design these algorithms. The proposed algorithms have been applied to various machine learning tasks.

Curse of non-parametric methods

During the past few decades is has become relatively easy to gather huge amount of data, apprehensively called-*massive data sets*. *Learning* is a principled method for distilling *predictive* models from the data. The *parametric approach* to learning assumes a functional form for the model to be learnt, and then estimates the unknown parameters. Once the model has been trained *the training examples can be discarded*. However, unless the form of the function is known a priori, assuming a certain form very often leads to erroneous inference. The *nonparametric methods*–also known as *memory based methods*–do not make any assumptions on the form of the underlying function. A price to be paid is that all the available *data has to be retained* while making the inference. Most of the current stat-of-the-art nonparametric machine learning algorithms have the computational complexity of either $\mathcal{O}(N^2)$ (for prediction) or $\mathcal{O}(N^3)$ (for training). This has seriously restricted the use of massive data sets. At the heart of various algorithms is the multiplication of a structured matrix with a vector, which we refer to as *matrix-vector product* (MVP) primitive. This MVP is the bottleneck contributing to the $\mathcal{O}(N^2)$ quadratic complexity. In my thesis I use ideas and techniques from computational physics (fast multipole methods), scientific computing (Krylov subspace methods), and computational geometry (kd-trees,clustering) to speed up *approximate* calculation of these primitives to $\mathcal{O}(N)$ and also provide *high accuracy guarantees*.

In most kernel based machine learning algorithms, Gaussian processes, and non-parametric statistics a key computationally intensive task is to compute a linear combination of local kernel functions centered on the training data, *i.e.*,

$$f(x) = \sum_{i=1}^{N} q_i k(x, x_i), \tag{1}$$

where $\{x_i \in \mathbb{R}^d, i = 1, \ldots, N\}$ are the N training data points, $\{q_i \in \mathbb{R}, i = 1, \ldots, N\}$ are the weights, $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the local kernel function, and $x \in \mathbb{R}^d$ is the test point at which $f(.)$ is to be computed. For kernel machines (e.g. regularized least squares, support vector machines, kernel regression) f is the regression/classification function. In case of Gaussian process regression f is the mean prediction. For non-parametric density estimation it is the kernel density estimate.. The computational complexity to evaluate (1) at a given test point is $\mathcal{O}(N)$. Training these models scales as $\mathcal{O}(N^3)$ since most involve solving the linear system of equation $(\mathbf{K} + \lambda \mathbf{I})\xi = \mathbf{y}$, where $\mathbf{K}$ is the $N \times N$ Gram matrix where $[\mathbf{K}]_{ij} = k(x_i, x_j)$. Recently, such non-parametric problems have been collectively referred to as *N-body problems in learning* (Gray & Moore 2001), in analogy with the gravitational N-body potential problems occurring in computational physics (Greengard 1994).

Fast approximate matrix-vector product

In general we need to evaluate (1) at M points $\{y_j \in \mathbb{R}^d, j = 1, \ldots, M\}$ leading to the quadratic $\mathcal{O}(MN)$ cost. The sum can be thought of as a *matrix-vector product* $f = \mathbf{K}q$, where $\mathbf{K}$ is a $M \times N$ matrix the entries of which are of the form $[\mathbf{K}]_{ij} = k(y_j, x_i)$ and $q = [q_1, \ldots, q_N]^T$ is a $N \times 1$ column vector. We develop fast ϵ-exact algorithms that compute the sum approximately in linear $\mathcal{O}(M + N)$ time. The algorithm is ϵ-exact, *i.e.*, for any given $\epsilon > 0$, $\widehat{f}$ is an $\epsilon - exact$ approximation to f if $\max_{y_j} \left[|\hat{f}(y_j) - f(y_j)|/Q \right] \leq \epsilon$ where $Q = \sum_{i=1}^{N} |q_i|$. The constant in $\mathcal{O}(M + N)$, depends on the desired *accuracy* ϵ, which however can be *arbitrary*. The fast algorithm is based on series expansion of the kernel and retaining only

Core MVP primitive and applications
Gaussian $G(y_j) = \sum_{i=1}^{N} q_i e^{-\|y_j - x_i\|^2/h^2}$ kernel density estimation, Gaussian process regression implicit surface fitting
Hermite $\times$ Gaussian $G(y_j) = \sum_{i=1}^{N} q_i H_r\left(\frac{y_j - x_i}{h_1}\right) e^{-(y_j - x_i)^2/h_2^2}$ optimal bandwidth estimation, projection pursuit
error function $G(y_j) = \sum_{i=1}^{N} q_i \text{ erfc}(y_j - x_i)$ ranking, collaborative filtering

Table 1: The fast summation algorithms designed and tasks to which they were applied.

the first few terms contributing to the desired accuracy. The algorithms are in the spirit of *fast multipole methods* used in computational physics (Greengard 1994).

Current thesis contributions

The thesis consists of two core contributions–(1) design of fast summation algorithms and (2) applying these fast primitives to certain large scale machine learning problems. Table 1 summarizes the current contributions. Below we present a brief summary of the current progress.

- ***Fast computation of sums of Gaussians*** The most commonly used kernel function is the *Gaussian kernel* $e^{-\|x-y\|^2/h^2}$, where h is called the *bandwidth* of the kernel. The fast Gauss transform proposed by (Greengard & Strain 1991) is a ϵ-exact approximation algorithm that reduces the computational complexity of the evaluation of the sum of N Gaussians at M points in d dimensions from $\mathcal{O}(MN)$ to $\mathcal{O}(M+N)$. However, the constant factor in $\mathcal{O}(M+N)$ grows exponentially with increasing dimensionality d, which makes the algorithm impractical for dimensions greater than three. We present a new algorithm where the constant factor is reduced to asymptotically polynomial order. As an example we show how the proposed method can be used for very fast multivariate kernel density estimation and fast Gaussian process regression. (Raykar *et al.* 2005; Raykar & Duraiswami 2007b)
- ***Fast optimal bandwidth estimation*** We propose an approximation algorithm for the univariate Gaussian kernel based density derivative estimation that reduces the computational complexity from $\mathcal{O}(MN)$ to linear $\mathcal{O}(M+N)$. We apply the density derivative evaluation procedure to estimate the optimal bandwidth for kernel density estimation, a process that is often intractable for large data sets. We also demonstrate that the proposed procedure can be extremely useful for speeding up projection pursuit techniques. (Raykar & Duraiswami 2005; 2006)
- ***Large scale preference learning*** Relying on an fast MVP for the error-function, we reduce the computational complexity of each iteration of a conjugate gradient algorithm for learning ranking functions from $\mathcal{O}(m^2)$, to $\mathcal{O}(m)$. Experiments indicate that the proposed algorithm is as accurate as the best available methods in terms of ranking accuracy is several orders of magnitude faster. The fast ranking procedure was applied to a collaborative filtering task. (Raykar & Duraiswami 2007a; Raykar, Duraiswami, & Krishnapuram 2007)

Future work

The following problems are among those that I wish to formulate well and solve in the course of this thesis.

- ***Core algorithms*** Development of these kind of fast approximate algorithms for more kernels–e.g., the Epanechnikov kernel for density estimation and the Matèrn class of kernels used in Gaussian process regression.
- ***Convergence issues*** In many applications these fast MVP primitives are embedded in a optimization routine–e.g., in ranking problem we embedded it in a conjugate-gradient procedure. A theoretical issue which we have barely touched upon concerns the convergence of these optimization routines when using approximate MVP primitives.
- ***Applications*** I would like to further explore different applications where these fast primitives could be useful.

A more ambitious task would be to explore if there are any deeper connections between structure in the data, computation, and inference. I am also planning on releasing the source code for all the algorithms under the LGPL.

References

Gray, A., and Moore, A. 2001. N-body problems in statistical learning. In *NIPS 2001*, 521–527.

Greengard, L., and Strain, J. 1991. The fast Gauss transform. *SIAM J. of Scien. and Stat. Comp.* 12(1):79–94.

Greengard, L. 1994. Fast algorithms for classical physics. *Science* 265(5174):909–914.

Raykar, V. C., and Duraiswami, R. 2005. Very fast optimal bandwidth selection for univariate kernel density estimation. Tech. Report CS-TR-4774, Univ. of Maryland, Collegepark.

Raykar, V. C., and Duraiswami, R. 2006. Fast optimal bandwidth selection for kernel density estimation. In *SIAM Int. Conf. on Data Mining*, 524–528.

Raykar, V. C., and Duraiswami, R. 2007a. Fast weighted summation of erfc functions. Tech. Report CS-TR-4848, Univ. of Maryland, Collegepark.

Raykar, V. C., and Duraiswami, R. 2007b. *Large Scale Kernel Machines*. MIT Press. chapter The Improved Fast Gauss Transform with applications to machine learning.

Raykar, V. C.; Yang, C.; Duraiswami, R.; and Gumerov, N. 2005. Fast computation of sums of Gaussians in high dimensions. Tech. Report CS-TR-4767, Univ. of Maryland, Collegepark.

Raykar, V. C.; Duraiswami, R.; and Krishnapuram, B. 2007. A fast algorithm for learning large scale preference relations. In *AISTATS 2007*.