tive, i.e., where the we have a strict inequality, and solve the resulting restricted quadratic program, for instance by conjugate gradient descent. We will encounter subset selection methods in Chapter 10.

6.5 Maximum Search Problems

Approximations

In several cases the task of finding an optimal function for estimation purposes means finding the best element from a finite set, or sometimes finding an optimal subset from a finite set of elements. These are discrete (sometimes combinatorial) optimization problems which are not so easily amenable to the techniques presented in the previous two sections. Furthermore, many commonly encountered problems are computationally expensive if solved exactly. Instead, by using probabilistic methods, it is possible to find *almost* optimal approximate solutions. These probabilistic methods are the topic of the present section.

6.5.1 Random Subset Selection

Consider the following problem: given a set of m functions, say $M := \{f_1, \ldots, f_m\}$, and some criterion $Q[f]$, find the function $\hat{f}$ that maximizes $Q[f]$. More formally,

$$\hat{f} := \operatorname*{argmax}_{f \in M} Q[f]. \tag{6.91}$$

Clearly, unless we have additional knowledge about the values $Q[f_i]$, we have to compute all terms $Q[f_i]$ if we want to solve (6.91) exactly. This will cost $O(m)$ operations. If m is large, which is often the case in practical applications, this operation is too expensive. In sparse greedy approximation problems (Section 10.2) or in Kernel Feature Analysis (Section 14.4), m can easily be of the order of 10^5 or larger (here, m is the number of training patterns). Hence we have to look for cheaper *approximate* solutions.

The key idea is to pick a random subset $M' \subset M$ that is sufficiently large, and take the maximum over M' as an approximation of the maximum over M. Provided the distribution of the values of $Q[f_i]$ is "well behaved", i.e., there exists not a small fraction of $Q[f_i]$ whose values are significantly smaller or larger than the average, we will obtain a solution that is close to the optimum with high probability. To formalize these ideas, we need the following result.

Lemma 6.31 (Maximum of Random Variables) *Denote by ξ, ξ' two independent random variables on $\mathbb{R}$ with corresponding distributions $\mathrm{P}_\xi, \mathrm{P}_{\xi'}$ and distribution functions $F_\xi, F_{\xi'}$. Then the random variable $\tilde{\xi} := \max(\xi, \xi')$ has the distribution function $F_{\tilde{\xi}} = F_\xi F_{\xi'}$.*

Proof Note that for a random variable, the distribution function $F(\xi_0)$ is given by

the probability $\mathrm{P}\{\xi \leq \xi_0\}$. Since ξ and ξ' are independent, we may write

$$F_{\bar{\xi}}(\bar{\xi}) = \mathrm{P}\left\{\max(\xi, \xi') \leq \bar{\xi}\right\} = \mathrm{P}\left\{\xi \leq \bar{\xi} \text{ and } \xi' \leq \bar{\xi}\right\} = \mathrm{P}\left\{\xi \leq \bar{\xi}\right\} \mathrm{P}\left\{\xi' \leq \bar{\xi}\right\} = F_{\xi}(\bar{\xi}) F_{\xi'}(\bar{\xi}), \tag{6.92}$$

which proves the claim. ∎

Distribution Over $\bar{\xi}$ is More Peaked

Repeated application of Lemma 6.31 leads to the following corollary.

Corollary 6.32 (Maximum Over Identical Random Variables) *Let $\xi_1, \ldots, \xi_{\tilde{m}}$ be $\tilde{m}$ independent and identically distributed (iid) random variables, with corresponding distribution function F_{ξ}. Then the random variable $\bar{\xi} := \max(\xi_1, \ldots, \xi_{\tilde{m}})$ has the distribution function $F_{\bar{\xi}}(\bar{\xi}) = \left(F_{\xi}(\bar{\xi})\right)^{\tilde{m}}$.*

In practice, the random variables ξ_i will be the values of $Q[f_i]$, where the f_i are drawn from the set M. If we draw them without replacement (i.e. none of the functions f_i appears twice), however, the values after each draw are dependent and we cannot apply Corollary 6.32 directly. Nonetheless, we can see that the maximum over draws *without* replacement will be larger than the maximum *with* replacement, since recurring observations can be understood as reducing the effective size of the set to be considered. Thus Corollary 6.32 gives us a *lower bound* on the value of the distribution function for draws without replacement. Moreover, for large m the difference between draws with and without replacement is small.

If the distribution of $Q[f_i]$ is known, we may use the distribution directly to determine the size $\tilde{m}$ of a subset to be used to find some $Q[f_i]$ that is almost as good as the solution to (6.91). In all other cases, we have to resort to assessing the *relative* quality of maxima over subsets. The following theorem tells us how.

Best Element of a Subset

Theorem 6.33 (Ranks on Random Subsets) *Denote by $M := \{x_1, \ldots, x_m\} \subset \mathbb{R}$ a set of cardinality m, and by $\tilde{M} \subset M$ a random subset of size $\tilde{m}$. Then the probability that* $\max \tilde{M}$ *is greater equal than n elements of M is at least $1 - \left(\frac{n}{m}\right)^{\tilde{m}}$.*

Proof We prove this by assuming the converse, namely that $\max \tilde{M}$ is smaller than $(m - n)$ elements of M. For $\tilde{m} = 1$ we know that this probability is $\frac{n}{m}$, since there are n elements to choose from. For $\tilde{m} > 1$, the probability is the one of choosing $\tilde{m}$ elements out of a subset M_{low} of n elements, rather than all m elements. Therefore we have that

$$\mathrm{P}(\tilde{M} \subset M_{\text{low}}) = \frac{\binom{n}{\tilde{m}}}{\binom{m}{\tilde{m}}} = \frac{n}{m} \cdot \frac{n-1}{m-1} \cdot \ldots \cdot \frac{n-\tilde{m}+1}{m-\tilde{m}+1} < \left(\frac{n}{m}\right)^{\tilde{m}}.$$

Consequently the probability that the maximum over $\tilde{M}$ will be larger than n elements of M is given by $1 - \mathrm{P}(\tilde{M} \subset M_{\text{low}}) \geq 1 - \left(\frac{n}{m}\right)^{\tilde{m}}$. ∎

The practical consequence is that we may use $1 - \left(\frac{n}{m}\right)^{\tilde{m}}$ to compute the required size of a random subset to achieve the desired degree of approximation. If we want to obtain results in the $\frac{n}{m}$ percentile range with $1 - \eta$ confidence, we must

solve for $\tilde{m} = \frac{\log(1-\eta)}{\ln n/m}$. To give a numerical example, if we desire values that are better than 95% of all other estimates with $1 - 0.05$ probability, then $\kappa = 59$ samples are sufficient. This $(95\%, 95\%, 59)$ rule is very useful in practice.[10] A similar method was used to speed up the process of boosting classifiers in the MadaBoost algorithm [143]. Furthermore, one could think whether it might not be useful to recycle old observations rather than computing all 59 values from scratch. If this can be done cheaply, and under some additional independence assumptions, subset selection methods can be improved further. For details see [424] who use the method in the context of memory management for operating systems.

6.5.2 Random Evaluation

Quite often, the evaluation of the term $Q[f]$ itself is rather time consuming, especially if $Q[f]$ is the sum of many (m, for instance) iid random variables. Again, we can speed up matters considerably by using probabilistic methods. The key idea is that averages over independent random variables are concentrated, which is to say that averages over subsets do not differ too much from averages over the whole set.

Approximating Sums by Partial Sums

Hoeffding's Theorem (Section 5.2) quantifies the size of the deviations between the expectation of a sum of random variables and their values at individual trials. We will use this to bound deviations between averages over sets and subsets. All we have to do is translate Theorem 5.1 into a statement regarding sample averages over different sample sizes. This can be readily constructed as follows:

Corollary 6.34 (Deviation Bounds for Empirical Means [508]) *Suppose $\xi_1, \ldots, \xi_m$ are iid bounded random variables, falling into the interval $[a, a+b]$ with probability one. Denote their average by $Q_m = \frac{1}{m}\sum_i \xi_i$. Furthermore, denote by $\xi_{s(1)}, \ldots, \xi_{s(\tilde{m})}$ with $\tilde{m} < m$ a subset of the same random variables (with $s : \{1, \ldots, \tilde{m}\} \to \{1, \ldots, m\}$ being an injective map, i.e. $s(i) = s(j)$ only if $i = j$), and $Q_{\tilde{m}} = \frac{1}{\tilde{m}}\sum_i \xi_{s(i)}$. Then for any $\varepsilon > 0$,*

Deviation of Subsets

$$\left.\begin{array}{l} \mathrm{P}\{Q_m - Q_{\tilde{m}} \geq \varepsilon\} \\ \mathrm{P}\{Q_{\tilde{m}} - Q_m \geq \varepsilon\} \end{array}\right\} \leq \exp\left(-\frac{2m\tilde{m}\varepsilon^2}{(m-\tilde{m})b^2}\right) = \exp\left(-2m\frac{\varepsilon^2}{b^2}\frac{\frac{\tilde{m}}{m}}{1-\frac{\tilde{m}}{m}}\right) \tag{6.93}$$

Proof By construction $\mathbf{E}[Q_m - Q_{\tilde{m}}] = 0$, since Q_m and $Q_{\tilde{m}}$ are both averages over sums of random variables drawn from the same distribution. Hence we only have to rewrite $Q_m - Q_{\tilde{m}}$ as an average over (different) random variables to apply Hoeffding's bound. Since all Q_i are identically distributed, we may pick the first $\tilde{m}$ random variables, without loss of generality. In other words, we assume that

10. During World War I tanks were often numbered in continuous increasing order. Unfortunately this "feature" allowed the enemy to estimate the number of tanks. How?

$s(i) = i$ for $i = 1, \ldots, \tilde{m}$. Then

$$Q_m - Q_{\tilde{m}} = \frac{1}{m}\sum_{i=1}^{m} \xi_i - \frac{1}{\tilde{m}}\sum_{i=1}^{\tilde{m}} \xi_i = \frac{1}{m}\sum_{i=1}^{\tilde{m}} \left(1 - \frac{m}{\tilde{m}}\right)\xi_i + \frac{1}{m}\sum_{i=\tilde{m}+1}^{m} \xi_i. \tag{6.94}$$

Thus we may split up $Q_m - Q_{\tilde{m}}$ into a sum of $\tilde{m}$ random variables with range $b_i = (\frac{m}{\tilde{m}} - 1)b$, and $m - \tilde{m}$ random variables with range $b_i = b$. We obtain

$$\sum_{i=1}^{m} b_i^2 = b^2\tilde{m}\left(\frac{m}{\tilde{m}} - 1\right)^2 + (m - \tilde{m})b^2 = b^2(m - \tilde{m})\frac{m}{\tilde{m}}. \tag{6.95}$$

Substituting this into (5.7) and noting that $Q_m - Q_{\tilde{m}} - \mathbf{E}[Q_m - Q_{\tilde{m}}] = Q_m - Q_{\tilde{m}}$ completes the proof. ∎

For small $\frac{\tilde{m}}{m}$ the rhs in (6.93) reduces to $\exp\left(-\frac{2\tilde{m}\varepsilon^2}{b^2}\right)$. In other words, deviations on the subsample $\tilde{m}$ dominate the overall deviation of $Q_m - Q_{\tilde{m}}$ from 0. This allows us to compute a cutoff criterion for evaluating Q_m by computing only a subset of its terms.

Cutoff Criterion

We need only solve (6.93) for $\frac{\tilde{m}}{m}$. Hence, in order to ensure that $Q_{\tilde{m}}$ is within ε of Q_m with probability $1 - \eta$, we have to take a fraction $\frac{\tilde{m}}{m}$ of samples that satisfies

$$\frac{\frac{\tilde{m}}{m}}{1 - \frac{\tilde{m}}{m}} = \frac{b^2(\ln 2 - \ln \eta)}{2m\varepsilon^2} =: c, \text{ and therefore } \frac{\tilde{m}}{m} = \frac{c}{1 + c}. \tag{6.96}$$

The fraction $\frac{\tilde{m}}{m}$ can be small for large m, which is exactly the case where we need methods to speed up evaluation.

6.5.3 Greedy Optimization Strategies

Quite often the overall goal is not necessarily to find the single best element x_i from a set X to solve a problem, but to find a good subset $\tilde{X} \subset X$ of size $\tilde{m}$ according to some quality criterion $Q[\tilde{X}]$. Problems of this type include approximating a matrix by a subset of its rows and columns (Section 10.2), finding approximate solutions to Kernel Fisher Discriminant Analysis (Chapter 15) and finding a sparse solution to the problem of Gaussian Process Regression (Section 16.3.4). These all have a common structure:

Applications

(i) Finding an optimal set $\tilde{X} \subset X$ is quite often a combinatorial problem, or it even may be NP-hard, since it means selecting $\tilde{m} = |\tilde{X}|$ elements from a set of $m = |X|$ elements. There are $\binom{m}{\tilde{m}}$ different choices, which clearly prevents an exhaustive search over all of them. Additionally, the size of $\tilde{m}$ is often not known beforehand. Hence we need a fast approximate algorithm.

(ii) The evaluation of $Q[\tilde{X} \cup \{x_i\}]$ is inexpensive, provided $Q[\tilde{X}]$ has been computed before. This indicates that an iterative algorithm can be useful.

(iii) The value of $Q[X]$, or equivalently how well we would do by taking the whole set X, can be bounded efficiently by using $Q[\tilde{X}]$ (or some by-products of the computation of $Q[\tilde{M}]$) without actually computing $Q[X]$.

Algorithm 6.6 Sparse Greedy Algorithm

Require: Set of functions X, Precision ϵ, Criterion $Q[\cdot]$
  Set $\tilde{X} = \emptyset$
  repeat
    Choose random subset X' of size m' from $X \backslash \tilde{X}$.
    Pick $\hat{x} = \text{argmax}_{x \in X'} Q[X' \cup \{x\}]$
    $X' = X' \cup \{\hat{x}\}$
    If needed, (re)compute bound on $Q[X]$.
  until $Q[\tilde{X}] + \epsilon \geq$ Bound on $Q[X]$
Output: $\tilde{X}, Q[\tilde{X}]$

(iv) The set of functions X is typically very large (i.e. more than 10^5 elements), yet the individual improvements by f_i via $Q[\tilde{X} \cup \{x_i\}]$ do not differ too much, meaning that specific x_i for which $Q[\tilde{X} \cup \{x_i\}]$ deviate by a large amount from the rest of $Q[\tilde{X} \cup \{x_i\}]$ do not exist.

Iterative Enlargement of $\tilde{X}$

In this case we may use a sparse greedy algorithm to find near optimal solutions among the remaining $X \backslash \tilde{X}$ elements. This combines the idea of an iterative enlargement of $\tilde{X}$ by one more element at a time (which is feasible since we can compute $Q[\tilde{X} \cup \{f_i\}]$ cheaply) with the idea that we need not consider all f_i as possible candidates for the enlargement. This uses the reasoning in Section 6.5.1 combined with the fact that the distribution of the improvements is not too long tailed (cf. (iv)). The overall strategy is described in Algorithm 6.6.

Problems 6.9 and 6.10 contain more examples of sparse greedy algorithms.

6.6 Summary

This chapter gave an overview of different optimization methods, which form the basic toolbox for solving the problems arising in learning with kernels. The main focus was on convex and differentiable problems, hence the overview of properties of convex sets and functions defined on them.

The key insights in Section 6.1 are that *convex sets* can be defined by *level sets of convex functions* and that convex optimization problems have *one global minimum*. Furthermore, the fact that the solutions of convex maximization over polyhedral sets can be found on the vertices will prove useful in some unsupervised learning applications (Section 14.4).

Basic tools for unconstrained problems (Section 6.2) include interval cutting methods, the Newton method, Conjugate Gradient descent, and Predictor-Corrector methods. These techniques are often used as building blocks to solve more advanced constrained optimization problems.

Since constrained minimization is a fairly complex topic, we only presented a selection of fundamental results, such as necessary and sufficient conditions in the general case of nonlinear programming. The KKT conditions for differentiable