



33	5	Prox-Gradient Methods	29
34	6	Accelerating Gradient Methods	32
35	6.1	Heavy-Ball Method	32
36	6.2	Conjugate Gradient	33
37	6.3	Nesterov's Accelerated Gradient: Weakly Convex Case	34
38	6.4	Nesterov's Accelerated Gradient: Strongly Convex Case	36
39	6.5	Lower Bounds on Rates	39
40	7	Newton Methods	40
41	7.1	Basic Newton's Method	40
42	7.2	Newton's Method for Convex Functions	42
43	7.3	Newton Methods for Nonconvex Functions	43
44	7.4	A Cubic Regularization Approach	45
45	8	Conclusions	47

## 1. Introduction

In this article, we consider algorithms for solving smooth optimization problems, possibly with simple constraints or structured nonsmooth regularizers. One such canonical formulation is

$$(1.0.1) \quad \min_{x \in \mathbb{R}^n} f(x),$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  has at least Lipschitz continuous gradients. Additional assumptions about  $f$ , such as convexity and Lipschitz continuity of the Hessian, are introduced as needed. Another formulation we consider is

$$(1.0.2) \quad \min_{x \in \mathbb{R}^n} f(x) + \lambda\psi(x),$$

where  $f$  is as in (1.0.1),  $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$  is a function that is usually convex and usually nonsmooth, and  $\lambda \geq 0$  is a regularization parameter.<sup>1</sup> We refer to (1.0.2) as a *regularized* minimization problem because the presence of the term involving  $\psi$  induces certain structural properties on the solution, that make it more desirable or plausible in the context of the application. We describe iterative algorithms that generate a sequence  $\{x^k\}_{k=0,1,2,\dots}$  of points that, in the case of convex objective functions, converges to the set of solutions. (Some algorithms also generate other “auxiliary” sequences of iterates.)

We are motivated to study problems of the forms (1.0.1) and (1.0.2) by their ubiquity in data analysis applications. Accordingly, Section 2 describes some canonical problems in data analysis and their formulation as optimization problems. After some preliminaries in Section 3, we describe in Section 4 algorithms that take step based on the gradients  $\nabla f(x^k)$ . Extensions of these methods to

<sup>1</sup>A set  $S$  is said to be *convex* if for any pair of points  $z', z'' \in S$ , we have that  $\alpha z' + (1 - \alpha)z'' \in S$  for all  $\alpha \in [0, 1]$ . A function  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex if  $\phi(\alpha z' + (1 - \alpha)z'') \leq \alpha\phi(z') + (1 - \alpha)\phi(z'')$  for all  $z', z''$  in the (convex) domain of  $\phi$  and all  $\alpha \in [0, 1]$ .

68 the case (1.0.2) of regularized objectives are described in Section 5. Section 6 de-  
 69 scribes accelerated gradient methods, which achieve better worst-case complexity  
 70 than basic gradient methods, while still only using first-derivative information.  
 71 We discuss Newton’s method in Section 7, outlining variants that can guarantee  
 72 convergence to points that approximately satisfy second-order conditions for a  
 73 local minimizer of a smooth nonconvex function.

74 **1.1. Omissions** Our approach throughout is to give a concise description of  
 75 some of the most important algorithmic tools for smooth nonlinear optimization  
 76 and regularized optimization, along with the basic convergence theory for each.  
 77 (In any given context, we mean by “smooth” that the function is differentiable as  
 78 many times as is necessary for the discussion to make sense.) In most cases, the  
 79 theory is elementary enough to include here in its entirety. In the few remaining  
 80 cases, we provide citations to works in which complete proofs can be found.

81 Although we allow nonsmoothness in the regularization term in (1.0.2), we do  
 82 not cover subgradient methods or mirror descent explicitly in this chapter. We  
 83 also do not discuss stochastic gradient methods, a class of methods that is central  
 84 to modern machine learning. All these topics are discussed in the contribution of  
 85 John Duchi to the current volume [22]. Other omissions include the following.

- 86 • Coordinate descent methods; see [47] for a recent review.
- 87 • Augmented Lagrangian methods, including alternating direction meth-  
 88 ods of multipliers (ADMM) [23]. The review [5] remains a good reference  
 89 for the latter topic, especially as it applies to problems from data analysis.
- 90 • Semidefinite programming (see [43, 45]) and conic optimization (see [6]).
- 91 • Methods tailored specifically to linear or quadratic programming, such as  
 92 the simplex method or interior-point methods (see [46] for a discussion of  
 93 the latter).
- 94 • Quasi-Newton methods, which modify Newton’s method by approxim-  
 95 ating the Hessian or its inverse, thus attaining attractive theoretical and  
 96 practical performance without using any second-derivative information.  
 97 For a discussion of these methods, see [36, Chapter 6]. One important  
 98 method of this class, which is useful in data analysis and many other  
 99 large-scale problems, is the limited-memory method L-BFGS [30]; see also  
 100 [36, Section 7.2].

101 **1.2. Notation** Our notational conventions in this chapter are as follows. We  
 102 use upper-case Roman characters ( $A$ ,  $L$ ,  $R$ , and so on) for matrices and lower-  
 103 case Roman ( $x$ ,  $v$ ,  $u$ , and so on) for vectors. (Vectors are assumed to be *column*  
 104 vectors.) Transposes are indicated by a superscript “ $T$ .” Elements of matrices and  
 105 vectors are indicated by subscripts, for example,  $A_{ij}$  and  $x_j$ . Iteration numbers are  
 106 indicated by superscripts, for example,  $x^k$ . We denote the set of real numbers by  
 107  $\mathbb{R}$ , so that  $\mathbb{R}^n$  denotes the Euclidean space of dimension  $n$ . The set of symmetric  
 108 real  $n \times n$  matrices is denoted by  $\text{SIR}^{n \times n}$ . Real scalars are usually denoted by

109 Greek characters, for example,  $\alpha$ ,  $\beta$ , and so on, though in deference to convention,  
 110 we sometimes use Roman capitals (for example,  $L$  for the Lipschitz constant of  
 111 a gradient). Where vector norms appear, the type of norm in use is indicated  
 112 by a subscript (for example  $\|x\|_1$ ), except that when no subscript appears, the  
 113 Euclidean norm  $\|\cdot\|_2$  is assumed. Matrix norms are defined where first used.

## 114 2. Optimization Formulations of Data Analysis Problems

115 In this section, we describe briefly some representative problems in data anal-  
 116 ysis and machine learning, emphasizing their formulation as optimization prob-  
 117 lems. Our list is by no means exhaustive. In many cases, there are a number of  
 118 different ways to formulate a given application as an optimization problem. We  
 119 do not try to describe all of them. But our list here gives a flavor of the interface  
 120 between data analysis and optimization.

121 **2.1. Setup** Practical data sets are often extremely messy. Data may be misla-  
 122 beled, noisy, incomplete, or otherwise corrupted. Much of the hard work in data  
 123 analysis is done by professionals, familiar with the underlying applications, who  
 124 “clean” the data and prepare it for analysis, while being careful not to change the  
 125 essential properties that they wish to discern from the analysis. Dasu and John-  
 126 son [19] claim out that “80% of data analysis is spent on the process of cleaning  
 127 and preparing the data.” We do not discuss this aspect of the process, focusing in-  
 128 stead on the part of the data analysis pipeline in which the problem is formulated  
 129 and solved.

130 The data set in a typical analysis problem consists of  $m$  objects:

$$131 \quad (2.1.1) \quad D := \{(a_j, y_j), j = 1, 2, \dots, m\},$$

132 where  $a_j$  is a vector (or matrix) of *features* and  $y_j$  is an *observation* or *label*. (Each  
 133 pair  $(a_j, y_j)$  has the same size and shape for all  $j = 1, 2, \dots, m$ .) The analysis  
 134 task then consists of discovering a function  $\phi$  such that  $\phi(a_j) \approx y_j$  holds for  
 135 most  $j = 1, 2, \dots, m$ . The process of discovering the mapping  $\phi$  is often called  
 136 “learning” or “training.”

137 The function  $\phi$  is often defined in terms of a vector or matrix of parameters,  
 138 which we denote by  $x$  or  $X$ . (Other notation also appears below.) With these  
 139 parametrizations, the problem of identifying  $\phi$  becomes a data-fitting problem:  
 140 “Find the parameters  $x$  defining  $\phi$  such that  $\phi(a_j) \approx y_j$ ,  $j = 1, 2, \dots, m$  in some  
 141 optimal sense.” Once we come up with a definition of the term “optimal,” we  
 142 have an optimization problem. Many such optimization formulations have objec-  
 143 tive functions of the “summation” type

$$144 \quad (2.1.2) \quad L_D(x) := \sum_{j=1}^m \ell(a_j, y_j; x),$$

145 where the  $j$ th term  $\ell(a_j, y_j; x)$  is a measure of the mismatch between  $\phi(a_j)$  and  
 146  $y_j$ , and  $x$  is the vector of parameters that determines  $\phi$ .

147 One use of  $\phi$  is to make predictions about future data items. Given another  
 148 previously unseen item of data  $\hat{a}$  of the same type as  $a_j$ ,  $j = 1, 2, \dots, m$ , we  
 149 predict that the label  $\hat{y}$  associated with  $\hat{a}$  would be  $\phi(\hat{a})$ . The mapping may also  
 150 expose other structure and properties in the data set. For example, it may reveal  
 151 that only a small fraction of the features in  $a_j$  are needed to reliably predict the  
 152 label  $y_j$ . (This is known as *feature selection*.) The function  $\phi$  or its parameter  $x$   
 153 may also reveal important structure in the data. For example,  $X$  could reveal a  
 154 low-dimensional subspace that contains most of the  $a_j$ , or  $X$  could reveal a matrix  
 155 with particular structure (low-rank, sparse) such that observations of  $X$  prompted  
 156 by the feature vectors  $a_j$  yield results close to  $y_j$ .

157 Examples of labels  $y_j$  include the following.

- 158 • A real number, leading to a *regression* problem.
- 159 • A label, say  $y_j \in \{1, 2, \dots, M\}$  indicating that  $a_j$  belongs to one of  $M$   
 160 classes. This is a *classification* problem. We have  $M = 2$  for binary classifi-  
 161 cation and  $M > 2$  for multiclass classification.
- 162 • Null. Some problems only have feature vectors  $a_j$  and no labels. In this  
 163 case, the data analysis task may consist of grouping the  $a_j$  into clusters  
 164 (where the vectors within each cluster are deemed to be functionally sim-  
 165 ilar), or identifying a low-dimensional subspace (or a collection of low-  
 166 dimensional subspaces) that approximately contains the  $a_j$ . Such prob-  
 167 lems require the labels  $y_j$  to be learned, alongside the function  $\phi$ . For  
 168 example, in a clustering problem,  $y_j$  could represent the cluster to which  
 169  $a_j$  is assigned.

170 Even after cleaning and preparation, the setup above may contain many com-  
 171 plications that need to be dealt with in formulating the problem in rigorous math-  
 172 ematical terms. The quantities  $(a_j, y_j)$  may contain noise, or may be otherwise  
 173 corrupted. We would like the mapping  $\phi$  to be robust to such errors. There may  
 174 be *missing data*: parts of the vectors  $a_j$  may be missing, or we may not know all  
 175 the labels  $y_j$ . The data may be arriving in *streaming* fashion rather than being  
 176 available all at once. In this case, we would learn  $\phi$  in an *online* fashion.

177 One particular consideration is that we wish to avoid *overfitting* the model to  
 178 the data set  $D$  in (2.1.1). The particular data set  $D$  available to us can often be  
 179 thought of as a finite sample drawn from some underlying larger (often infinite)  
 180 collection of data, and we wish the function  $\phi$  to perform well on the unobserved  
 181 data points as well as the observed subset  $D$ . In other words, we want  $\phi$  to  
 182 be not too sensitive to the particular sample  $D$  that is used to define empirical  
 183 objective functions such as (2.1.2). The optimization formulation can be modified  
 184 in various ways to achieve this goal, by the inclusion of constraints or penalty  
 185 terms that limit some measure of “complexity” of the function (such techniques  
 186 are called *generalization* or *regularization*). Another approach is to terminate the  
 187 optimization algorithm early, the rationale being that overfitting occurs mainly in  
 188 the later stages of the optimization process.

189 **2.2. Least Squares** Probably the oldest and best-known data analysis problem is  
 190 linear least squares. Here, the data points  $(a_j, y_j)$  lie in  $\mathbb{R}^n \times \mathbb{R}$ , and we solve

$$191 \quad (2.2.1) \quad \min_x \frac{1}{2m} \sum_{j=1}^m (a_j^\top x - y_j)^2 = \frac{1}{2m} \|Ax - y\|_2^2,$$

192 where  $A$  is the matrix whose rows are  $a_j^\top$ ,  $j = 1, 2, \dots, m$  and  $y = (y_1, y_2, \dots, y_m)^\top$ .  
 193 In the terminology above, the function  $\phi$  is defined by  $\phi(a) := a^\top x$ . (We could  
 194 also introduce a nonzero intercept by adding an extra parameter  $\beta \in \mathbb{R}$  and  
 195 defining  $\phi(a) := a^\top x + \beta$ .) This formulation can be motivated statistically, as a  
 196 maximum-likelihood estimate of  $x$  when the observations  $y_j$  are exact but for  
 197 i.i.d. Gaussian noise. Randomized linear algebra methods for large-scale in-  
 198 stances of this problem are discussed in Section 5 of the lectures of Drineas and  
 199 Mahoney [20] in this volume.

200 Various modifications of (2.2.1) impose desirable structure on  $x$  and hence on  
 201  $\phi$ . For example, Tikhonov regularization with a squared  $\ell_2$ -norm, which is

$$202 \quad \min_x \frac{1}{2m} \|Ax - y\|_2^2 + \lambda \|x\|_2^2, \quad \text{for some parameter } \lambda > 0,$$

203 yields a solution  $x$  with less sensitivity to perturbations in the data  $(a_j, y_j)$ . The  
 204 LASSO formulation

$$205 \quad (2.2.2) \quad \min_x \frac{1}{2m} \|Ax - y\|_2^2 + \lambda \|x\|_1$$

206 tends to yield solutions  $x$  that are sparse, that is, containing relatively few nonzero  
 207 components [42]. This formulation performs feature selection: The locations of  
 208 the nonzero components in  $x$  reveal those components of  $a_j$  that are instrumental  
 209 in determining the observation  $y_j$ . Besides its statistical appeal — predictors that  
 210 depend on few features are potentially simpler and more comprehensible than  
 211 those depending on many features — feature selection has practical appeal in  
 212 making predictions about future data. Rather than gathering all components of a  
 213 new data vector  $\hat{a}$ , we need to find only the “selected” features, since only these  
 214 are needed to make a prediction. The LASSO formulation (2.2.2) is an important  
 215 prototype for many problems in data analysis, in that it involves a regularization  
 216 term  $\lambda \|x\|_1$  that is nonsmooth and convex, but with relatively simple structure  
 217 that can potentially be exploited by algorithms.

218 **2.3. Matrix Completion** Matrix completion is in one sense a natural extension  
 219 of least-squares to problems in which the data  $a_j$  are naturally represented as  
 220 matrices rather than vectors. Changing notation slightly, we suppose that each  
 221  $A_j$  is an  $n \times p$  matrix, and we seek another  $n \times p$  matrix  $X$  that solves

$$222 \quad (2.3.1) \quad \min_x \frac{1}{2m} \sum_{j=1}^m (\langle A_j, X \rangle - y_j)^2,$$

223 where  $\langle A, B \rangle := \text{trace}(A^\top B)$ . Here we can think of the  $A_j$  as “probing” the un-  
 224 known matrix  $X$ . Commonly considered types of observations are random linear

225 combinations (where the elements of  $A_j$  are selected i.i.d. from some distribution)  
 226 or single-element observations (in which each  $A_j$  has 1 in a single location and  
 227 zeros elsewhere). A regularized version of (2.3.1), leading to solutions  $X$  that are  
 228 low-rank, is

$$229 \quad (2.3.2) \quad \min_X \frac{1}{2m} \sum_{j=1}^m (\langle A_j, X \rangle - y_j)^2 + \lambda \|X\|_*,$$

230 where  $\|X\|_*$  is the nuclear norm, which is the sum of singular values of  $X$  [39].  
 231 The nuclear norm plays a role analogous to the  $\ell_1$  norm in (2.2.2). Although the  
 232 nuclear norm is a somewhat complex nonsmooth function, it is at least convex, so  
 233 that the formulation (2.3.2) is also convex. This formulation can be shown to yield  
 234 a statistically valid solution when the true  $X$  is low-rank and the observation ma-  
 235 trices  $A_j$  satisfy a “restricted isometry” property, commonly satisfied by random  
 236 matrices, but not by matrices with just one nonzero element. The formulation is  
 237 also valid in a different context, in which the true  $X$  is incoherent (roughly speak-  
 238 ing, it does not have a few elements that are much larger than the others), and  
 239 the observations  $A_j$  are of single elements [10].

240 In another form of regularization, the matrix  $X$  is represented explicitly as a  
 241 product of two “thin” matrices  $L$  and  $R$ , where  $L \in \mathbb{R}^{n \times r}$  and  $R \in \mathbb{R}^{p \times r}$ , with  
 242  $r \ll \min(n, p)$ . We set  $X = LR^T$  in (2.3.1) and solve

$$243 \quad (2.3.3) \quad \min_{L,R} \frac{1}{2m} \sum_{j=1}^m (\langle A_j, LR^T \rangle - y_j)^2.$$

244 In this formulation, the rank  $r$  is “hard-wired” into the definition of  $X$ , so there is  
 245 no need to include a regularizing term. This formulation is also typically much  
 246 more compact than (2.3.2); the total number of elements in  $(L, R)$  is  $(n + p)r$ ,  
 247 which is much less than  $np$ . A disadvantage is that it is nonconvex. An active  
 248 line of current research, pioneered in [9] and also drawing on statistical sources,  
 249 shows that the nonconvexity is benign in many situations, and that under certain  
 250 assumptions on the data  $(A_j, y_j)$ ,  $j = 1, 2, \dots, m$  and careful choice of algorithmic  
 251 strategy, good solutions can be obtained from the formulation (2.3.3). A clue to  
 252 this good behavior is that although this formulation is nonconvex, it is in some  
 253 sense an approximation to a tractable problem: If we have a complete observation  
 254 of  $X$ , then a rank- $r$  approximation can be found by performing a singular value  
 255 decomposition of  $X$ , and defining  $L$  and  $R$  in terms of the  $r$  leading left and right  
 256 singular vectors.

257 **2.4. Nonnegative Matrix Factorization** Some applications in computer vision,  
 258 chemometrics, and document clustering require us to find factors  $L$  and  $R$  like  
 259 those in (2.3.3) in which all elements are nonnegative. If the full matrix  $Y \in \mathbb{R}^{n \times p}$   
 260 is observed, this problem has the form

$$261 \quad \min_{L,R} \|LR^T - Y\|_F^2, \quad \text{subject to } L \geq 0, R \geq 0.$$

262 **2.5. Sparse Inverse Covariance Estimation** In this problem, the labels  $y_j$  are  
 263 null, and the vectors  $a_j \in \mathbb{R}^n$  are viewed as independent observations of a ran-  
 264 dom vector  $A \in \mathbb{R}^n$ , which has zero mean. The sample covariance matrix con-  
 265 structed from these observations is

$$266 \quad S = \frac{1}{m-1} \sum_{j=1}^m a_j a_j^T.$$

267 The element  $S_{il}$  is an estimate of the covariance between the  $i$ th and  $l$ th elements  
 268 of the random variable vector  $A$ . Our interest is in calculating an estimate  $X$  of  
 269 the *inverse* covariance matrix that is *sparse*. The structure of  $X$  yields important  
 270 information about  $A$ . In particular, if  $X_{il} = 0$ , we can conclude that the  $i$  and  
 271  $l$  components of  $A$  are *conditionally independent*. (That is, they are independent  
 272 given knowledge of the values of the other  $n-2$  components of  $A$ .) Stated an-  
 273 other way, the nonzero locations in  $X$  indicate the arcs in the dependency graph  
 274 whose nodes correspond to the  $n$  components of  $A$ .

275 One optimization formulation that has been proposed for estimating the in-  
 276 verse sparse covariance matrix  $X$  is the following:

$$277 \quad (2.5.1) \quad \min_{X \in \mathbb{S}\mathbb{R}^{n \times n}, X \succeq 0} \langle S, X \rangle - \log \det(X) + \lambda \|X\|_1,$$

278 where  $\mathbb{S}\mathbb{R}^{n \times n}$  is the set of  $n \times n$  symmetric matrices,  $X \succeq 0$  indicates that  $X$  is  
 279 positive definite, and  $\|X\|_1 := \sum_{i,l=1}^n |X_{il}|$  (see [17, 25]).

280 **2.6. Sparse Principal Components** The setup for this problem is similar to the  
 281 previous section, in that we have a sample covariance matrix  $S$  that is estimated  
 282 from a number of observations of some underlying random vector. The *princi-*  
 283 *pal components* of this matrix are the eigenvectors corresponding to the largest  
 284 eigenvalues. It is often of interest to find *sparse* principal components, approxi-  
 285 mations to the leading eigenvectors that also contain few nonzeros. An explicit  
 286 optimization formulation of this problem is

$$287 \quad (2.6.1) \quad \max_{v \in \mathbb{R}^n} v^T S v \quad \text{s.t.} \quad \|v\|_2 = 1, \quad \|v\|_0 \leq k,$$

288 where  $\|\cdot\|_0$  indicates the cardinality of  $v$  (that is, the number of nonzeros in  $v$ )  
 289 and  $k$  is a user-defined parameter indicating a bound on the cardinality of  $v$ . The  
 290 problem (2.6.1) is NP-hard, so exact formulations (for example, as a quadratic  
 291 program with binary variables) are intractable. We consider instead a relaxation,  
 292 due to [18], which replaces  $vv^T$  by a positive semidefinite proxy  $M \in \mathbb{S}\mathbb{R}^{n \times n}$ :

$$293 \quad (2.6.2) \quad \max_{M \in \mathbb{S}\mathbb{R}^{n \times n}} \langle S, M \rangle \quad \text{s.t.} \quad M \succeq 0, \quad \langle I, M \rangle = 1, \quad \|M\|_1 \leq \rho,$$

294 for some parameter  $\rho > 0$  that can be adjusted to attain the desired sparsity. This  
 295 formulation is a convex optimization problem, in fact, a semidefinite program-  
 296 ming problem.

297 This formulation can be generalized to find the leading  $r > 1$  sparse principal  
 298 components. Ideally, we would obtain these from a matrix  $V \in \mathbb{R}^{n \times r}$  whose

299 columns are mutually orthogonal and have at most  $k$  nonzeros each. We can  
 300 write a convex relaxation of this problem, once again a semidefinite program, as

$$301 \quad (2.6.3) \quad \max_{M \in \mathbb{S}\mathbb{R}^{n \times n}} \langle S, M \rangle \quad \text{s.t.} \quad 0 \preceq M \preceq I, \langle I, M \rangle = 1, \|M\|_1 \leq \rho.$$

302 A more compact (but nonconvex) formulation is

$$303 \quad \max_{F \in \mathbb{R}^{n \times r}} \langle S, FF^T \rangle \quad \text{s.t.} \quad \|F\|_2 \leq 1, \|F\|_{2,1} \leq \bar{R},$$

304 where  $\|F\|_{2,1} := \sum_{i=1}^n \|F_i\|_2$  [15]. The latter regularization term is often called  
 305 a “group-sparse” or “group-LASSO” regularizer. (An early use of this type of  
 306 regularizer was described in [44].)

307 **2.7. Sparse Plus Low-Rank Matrix Decomposition** Another useful paradigm  
 308 is to decompose a partly or fully observed  $n \times p$  matrix  $Y$  into the sum of a  
 309 sparse matrix and a low-rank matrix. A convex formulation of the fully-observed  
 310 problem is

$$311 \quad \min_{M,S} \|M\|_* + \lambda \|S\|_1 \quad \text{s.t.} \quad Y = M + S,$$

where  $\|S\|_1 := \sum_{i=1}^n \sum_{j=1}^p |S_{ij}|$  [11, 14]. Compact, nonconvex formulations that  
 allow noise in the observations include the following:

$$\min_{L,R,S} \frac{1}{2} \|LR^T + S - Y\|_F^2 \quad (\text{fully observed})$$

$$\min_{L,R,S} \frac{1}{2} \|P_\Phi(LR^T + S - Y)\|_F^2 \quad (\text{partially observed}),$$

312 where  $\Phi$  represents the locations of the observed entries of  $Y$  and  $P_\Phi$  is projection  
 313 onto this set [15, 48].

314 One application of these formulations is to robust PCA, where the low-rank  
 315 part represents principal components and the sparse part represents “outlier”  
 316 observations. Another application is to foreground-background separation in  
 317 video processing. Here, each column of  $Y$  represents the pixels in one frame of  
 318 video, whereas each row of  $Y$  shows the evolution of one pixel over time.

319 **2.8. Subspace Identification** In this application, the  $a_j \in \mathbb{R}^n$ ,  $j = 1, 2, \dots, m$  are  
 320 vectors that lie (approximately) in a low-dimensional subspace. The aim is to  
 321 identify this subspace, expressed as the column subspace of a matrix  $X \in \mathbb{R}^{n \times r}$ .  
 322 If the  $a_j$  are fully observed, an obvious way to solve this problem is to perform  
 323 a singular value decomposition of the  $n \times m$  matrix  $A = [a_j]_{j=1}^m$ , and take  $X$  to  
 324 be the leading  $r$  right singular vectors. In interesting variants of this problem,  
 325 however, the vectors  $a_j$  may be arriving in streaming fashion and may be only  
 326 partly observed, for example in indices  $\Phi_j \subset \{1, 2, \dots, n\}$ . We would thus need to  
 327 identify a matrix  $X$  and vectors  $s_j \in \mathbb{R}^r$  such that

$$328 \quad P_{\Phi_j}(a_j - Xs_j) \approx 0, \quad j = 1, 2, \dots, m.$$

329 The algorithm for identifying  $X$ , described in [1], is a manifold-projection scheme  
 330 that takes steps in incremental fashion for each  $a_j$  in turn. Its validity relies on

331 incoherence of the matrix  $X$  with respect to the principal axes, that is, the matrix  
 332  $X$  should not have a few elements that are much larger than the others. A local  
 333 convergence analysis of this method is given in [2].

**2.9. Support Vector Machines** Classification via support vector machines (SVM)  
 is a classical paradigm in machine learning. This problem takes as input data  
 $(a_j, y_j)$  with  $a_j \in \mathbb{R}^n$  and  $y_j \in \{-1, 1\}$ , and seeks a vector  $x \in \mathbb{R}^n$  and a scalar  
 $\beta \in \mathbb{R}$  such that

$$(2.9.1a) \quad a_j^\top x - \beta \geq 1 \quad \text{when } y_j = +1;$$

$$(2.9.1b) \quad a_j^\top x - \beta \leq -1 \quad \text{when } y_j = -1.$$

334 Any pair  $(x, \beta)$  that satisfies these conditions defines a *separating hyperplane* in  
 335  $\mathbb{R}^n$ , that separates the “positive” cases  $\{a_j \mid y_j = +1\}$  from the “negative” cases  
 336  $\{a_j \mid y_j = -1\}$ . (In the language of Section 2.1, we could define the function  
 337  $\phi$  as  $\phi(a_j) = \text{sign}(a_j^\top x - \beta)$ .) Among all separating hyperplanes, the one that  
 338 minimizes  $\|x\|^2$  is the one that maximizes the *margin* between the two classes,  
 339 that is, the hyperplane whose distance to the nearest point  $a_j$  of either class is  
 340 greatest.

341 We can formulate the problem of finding a separating hyperplane as an opti-  
 342 mization problem by defining an objective with the summation form (2.1.2):

$$343 \quad (2.9.2) \quad H(x, \beta) = \frac{1}{m} \sum_{j=1}^m \max(1 - y_j(a_j^\top x - \beta), 0).$$

344 Note that the  $j$ th term in this summation is zero if the conditions (2.9.1) are  
 345 satisfied, and positive otherwise. Even if no pair  $(x, \beta)$  exists with  $H(x, \beta) = 0$ ,  
 346 the pair  $(x, \beta)$  that minimizes (2.1.2) will be the one that comes as close as possible  
 347 to satisfying (2.9.1), in a suitable sense. A term  $\lambda \|x\|_2^2$ , where  $\lambda$  is a small positive  
 348 parameter, is often added to (2.9.2), yielding the following regularized version:

$$349 \quad (2.9.3) \quad H(x, \beta) = \frac{1}{m} \sum_{j=1}^m \max(1 - y_j(a_j^\top x - \beta), 0) + \frac{1}{2} \lambda \|x\|_2^2.$$

350 If  $\lambda$  is sufficiently small (but positive), and if separating hyperplanes exist, the  
 351 pair  $(x, \beta)$  that minimizes (2.9.3) is the maximum-margin separating hyperplane.  
 352 The maximum-margin property is consistent with the goals of generalizability  
 353 and robustness. For example, if the observed data  $(a_j, y_j)$  is drawn from an  
 354 underlying “cloud” of positive and negative cases, the maximum-margin solution  
 355 usually does a reasonable job of separating other empirical data samples drawn  
 356 from the same clouds, whereas a hyperplane that passes close by several of the  
 357 observed data points may not do as well (see Figure 2.9.4).

358 The problem of minimizing (2.9.3) can be written as a convex quadratic pro-  
 359 gram — having a convex quadratic objective and linear constraints — by intro-  
 360 ducing variables  $s_j$ ,  $j = 1, 2, \dots, m$  to represent the residual terms. Then,

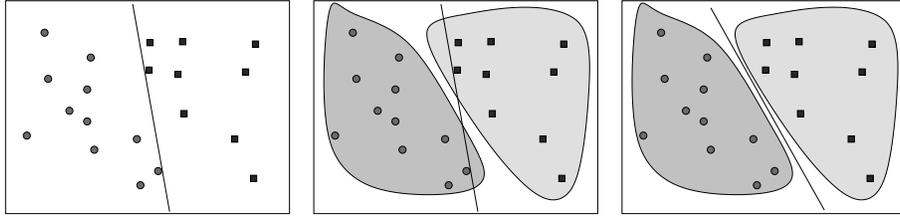


FIGURE 2.9.4. Linear support vector machine classification, with one class represented by circles and the other by squares. One possible choice of separating hyperplane is shown at left. If the observed data is an empirical sample drawn from a cloud of underlying data points, this plane does not do well in separating the two clouds (middle). The maximum-margin separating hyperplane does better (right).

$$(2.9.5a) \quad \min_{x, \beta, s} \frac{1}{m} \mathbf{1}^T s + \frac{1}{2} \lambda \|x\|_2^2,$$

$$(2.9.5b) \quad \text{subject to } s_j \geq 1 - y_j (a_j^T x - \beta), \quad s_j \geq 0, \quad j = 1, 2, \dots, m,$$

361 where  $\mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^m$ .

362 Often it is not possible to find a hyperplane that separates the positive and  
 363 negative cases well enough to be useful as a classifier. One solution is to trans-  
 364 form all of the raw data vectors  $a_j$  by a mapping  $\zeta$  into a higher-dimensional  
 365 Euclidean space, then perform the support-vector-machine classification on the  
 366 vectors  $\zeta(a_j)$ ,  $j = 1, 2, \dots, m$ .

The conditions (2.9.1) would thus be replaced by

$$(2.9.6a) \quad \zeta(a_j)^T x - \beta \geq 1 \quad \text{when } y_j = +1;$$

$$(2.9.6b) \quad \zeta(a_j)^T x - \beta \leq -1 \quad \text{when } y_j = -1,$$

367 leading to the following analog of (2.9.3):

$$368 \quad (2.9.7) \quad H(x, \beta) = \frac{1}{m} \sum_{j=1}^m \max(1 - y_j (\zeta(a_j)^T x - \beta), 0) + \frac{1}{2} \lambda \|x\|_2^2.$$

369 When transformed back to  $\mathbb{R}^m$ , the surface  $\{a | \zeta(a)^T x - \beta = 0\}$  is nonlinear and  
 370 possibly disconnected, and is often a much more powerful classifier than the  
 371 hyperplanes resulting from (2.9.3).

372 We can formulate (2.9.7) as a convex quadratic program in exactly the same  
 373 manner as we derived (2.9.5) from (2.9.3). By taking the dual of this quadratic  
 374 program, we obtain another convex quadratic program, in  $m$  variables:

$$375 \quad (2.9.8) \quad \min_{\alpha \in \mathbb{R}^m} \frac{1}{2} \alpha^T Q \alpha - \mathbf{1}^T \alpha \quad \text{subject to } 0 \leq \alpha \leq \frac{1}{\lambda} \mathbf{1}, \quad y^T \alpha = 0,$$

376 where

$$377 \quad Q_{kl} = y_k y_l \zeta(a_k)^T \zeta(a_l), \quad y = (y_1, y_2, \dots, y_m)^T, \quad \mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^m.$$

378 Interestingly, problem (2.9.8) can be formulated and solved without any explicit  
 379 knowledge or definition of the mapping  $\zeta$ . We need only a technique to define the  
 380 elements of  $Q$ . This can be done with the use of a *kernel function*  $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ ,  
 381 where  $K(a_k, a_l)$  replaces  $\zeta(a_k)^\top \zeta(a_l)$  [4, 16]. This is the so-called “kernel trick.”  
 382 (The kernel function  $K$  can also be used to construct a classification function  
 383  $\phi$  from the solution of (2.9.8).) A particularly popular choice of kernel is the  
 384 Gaussian kernel:

$$385 \quad K(a_k, a_l) := \exp(-\|a_k - a_l\|^2 / (2\sigma)),$$

386 where  $\sigma$  is a positive parameter.

387 **2.10. Logistic Regression** Logistic regression can be viewed as a variant of bi-  
 388 nary support-vector machine classification, in which rather than the classification  
 389 function  $\phi$  giving a unqualified prediction of the class in which a new data vector  
 390  $a$  lies, it returns an estimate of the *odds* of  $a$  belonging to one class or the other.  
 391 We seek an “odds function”  $p$  parametrized by a vector  $x \in \mathbb{R}^n$  as follows:

$$392 \quad (2.10.1) \quad p(a; x) := (1 + \exp(a^\top x))^{-1},$$

and aim to choose the parameter  $x$  so that

$$(2.10.2a) \quad p(a_j; x) \approx 1 \quad \text{when } y_j = +1;$$

$$(2.10.2b) \quad p(a_j; x) \approx 0 \quad \text{when } y_j = -1.$$

393 (Note the similarity to (2.9.1).) The optimal value of  $x$  can be found by maximizing  
 394 a log-likelihood function:

$$395 \quad (2.10.3) \quad L(x) := \frac{1}{m} \left[ \sum_{j:y_j=-1} \log(1 - p(a_j; x)) + \sum_{j:y_j=1} \log p(a_j; x) \right].$$

396 We can perform feature selection using this model by introducing a regularizer  
 397  $\lambda \|x\|_1$ , as follows:

$$398 \quad (2.10.4) \quad \max_x \frac{1}{m} \left[ \sum_{j:y_j=-1} \log(1 - p(a_j; x)) + \sum_{j:y_j=1} \log p(a_j; x) \right] - \lambda \|x\|_1,$$

399 where  $\lambda > 0$  is a regularization parameter. (Note that we *subtract* rather than add  
 400 the regularization term  $\lambda \|x\|_1$  to the objective, because this problem is formulated  
 401 as a maximization rather than a minimization.) As we see later, this term has  
 402 the effect of producing a solution in which few components of  $x$  are nonzero,  
 403 making it possible to evaluate  $p(a; x)$  by knowing only those components of  $a$   
 404 that correspond to the nonzeros in  $x$ .

405 An important extension of this technique is to *multiclass* (or *multinomial*) lo-  
 406 gistic regression, in which the data vectors  $a_j$  belong to more than two classes.  
 407 Such applications are common in modern data analysis. For example, in a speech  
 408 recognition system, the  $M$  classes could each represent a *phoneme* of speech, one  
 409 of the potentially thousands of distinct elementary sounds that can be uttered by

410 humans in a few tens of milliseconds. A multinomial logistic regression problem  
 411 requires a distinct odds function  $p_k$  for each class  $k \in \{1, 2, \dots, M\}$ . These func-  
 412 tions are parametrized by vectors  $x_{[k]} \in \mathbb{R}^n$ ,  $k = 1, 2, \dots, M$ , defined as follows:  
 413

$$414 \quad (2.10.5) \quad p_k(\mathbf{a}; \mathbf{X}) := \frac{\exp(\mathbf{a}^\top x_{[k]})}{\sum_{l=1}^M \exp(\mathbf{a}^\top x_{[l]})}, \quad k = 1, 2, \dots, M,$$

415 where we define  $\mathbf{X} := \{x_{[k]} \mid k = 1, 2, \dots, M\}$ . Note that for all  $\mathbf{a}$  and for all  
 416  $k = 1, 2, \dots, M$ , we have  $p_k(\mathbf{a}) \in (0, 1)$  and also  $\sum_{k=1}^M p_k(\mathbf{a}) = 1$ . The operation  
 417 in (2.10.5) is referred to as a “softmax” on the quantities  $\{\mathbf{a}^\top x_{[l]} \mid l = 1, 2, \dots, M\}$ .  
 418 If one of these inner products dominates the others, that is,  $\mathbf{a}^\top x_{[k]} \gg \mathbf{a}^\top x_{[l]}$  for  
 419 all  $l \neq k$ , the formula (2.10.5) will yield  $p_k(\mathbf{a}; \mathbf{X}) \approx 1$  and  $p_l(\mathbf{a}; \mathbf{X}) \approx 0$  for all  $l \neq k$ .

420 In the setting of multiclass logistic regression, the labels  $y_j$  are vectors in  $\mathbb{R}^M$ ,  
 421 whose elements are defined as follows:

$$422 \quad (2.10.6) \quad y_{jk} = \begin{cases} 1 & \text{when } a_j \text{ belongs to class } k, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly to (2.10.2), we seek to define the vectors  $x_{[k]}$  so that

$$(2.10.7a) \quad p_k(\mathbf{a}; \mathbf{X}) \approx 1 \quad \text{when } y_{jk} = 1$$

$$(2.10.7b) \quad p_k(\mathbf{a}; \mathbf{X}) \approx 0 \quad \text{when } y_{jk} = 0.$$

423 The problem of finding values of  $x_{[k]}$  that satisfy these conditions can again be  
 424 formulated as one of maximizing a log-likelihood:

$$425 \quad (2.10.8) \quad L(\mathbf{X}) := \frac{1}{m} \sum_{j=1}^m \left[ \sum_{\ell=1}^M y_{j\ell} (x_{[\ell]}^\top a_j) - \log \left( \sum_{\ell=1}^M \exp(x_{[\ell]}^\top a_j) \right) \right].$$

426 “Group-sparse” regularization terms can be included in this formulation to se-  
 427 lect a set of features in the vectors  $a_j$ , common to each class, that distinguish  
 428 effectively between the classes.

429 **2.11. Deep Learning** Deep neural networks are often designed to perform the  
 430 same function as multiclass logistic regression, that is, to classify a data vector  $\mathbf{a}$   
 431 into one of  $M$  possible classes, where  $M \geq 2$  is large in some key applications.  
 432 The difference is that the data vector  $\mathbf{a}$  undergoes a series of structured transfor-  
 433 mations before being passed through a multiclass logistic regression classifier of  
 434 the type described in the previous subsection.

435 The simple neural network shown in Figure 2.11.1 illustrates the basic ideas.  
 436 In this figure, the data vector  $a_j$  enters at the bottom of the network, each node in  
 437 the bottom layer corresponding to one component of  $a_j$ . The vector then moves  
 438 upward through the network, undergoing a structured nonlinear transformation  
 439 as it moves from one layer to the next. A typical form of this transformation,  
 440 which converts the vector  $a_j^{l-1}$  at layer  $l-1$  to input vector  $a_j^l$  at layer  $l$ , is

$$441 \quad a_j^l = \sigma(W^l a_j^{l-1} + g^l), \quad l = 1, 2, \dots, D,$$

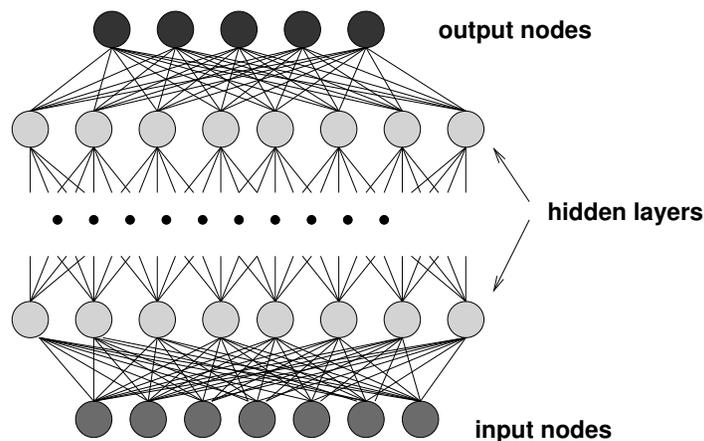


FIGURE 2.11.1. A deep neural network, showing connections between adjacent layers.

442 where  $W^l$  is a matrix of dimension  $|\alpha_j^l| \times |\alpha_j^{l-1}|$  and  $g^l$  is a vector of length  $|\alpha_j^l|$ ,  $\sigma$   
 443 is a *componentwise* nonlinear transformation, and  $D$  is the number of *hidden layers*,  
 444 defined as the layers situated strictly between the bottom and top layers. Each  
 445 arc in Figure 2.11.1 represents one of the elements of a transformation matrix  $W^l$ .  
 446 We define  $\alpha_j^0$  to be the “raw” input vector  $\alpha_j$ , and let  $\alpha_j^D$  be the vector formed  
 447 by the nodes at the topmost hidden layer in Figure 2.11.1. Typical forms of the  
 448 function  $\sigma$  include the following, acting identically on each component  $t \in \mathbb{R}$  of  
 449 its input vector:

- 450 • Logistic function:  $t \rightarrow 1/(1 + e^{-t})$ ;
- 451 • Hinge loss:  $t \rightarrow \max(t, 0)$ ;
- 452 • Bernoulli: a random function that outputs 1 with probability  $1/(1 + e^{-t})$   
 453 and 0 otherwise.

454 Each node in the top layer corresponds to a particular class, and the output of  
 455 each node corresponds to the odds of the input vector belonging to each class. As  
 456 mentioned, the “softmax” operator is typically used to convert the transformed  
 457 input vector in the second-top layer (layer  $D$ ) to a set of odds at the top layer. As-  
 458 sociated with each input vector  $\alpha_j$  are labels  $y_{jk}$ , defined as in (2.10.6) to indicate  
 459 which of the  $M$  classes that  $\alpha_j$  belongs to.

460 The parameters in this neural network are the matrix-vector pairs  $(W^l, g^l)$ ,  
 461  $l = 1, 2, \dots, D$  that transform the input vector  $\alpha_j$  into its form  $\alpha_j^D$  at the topmost  
 462 hidden layer, together with the parameters  $X$  of the multiclass logistic regression  
 463 operation that takes place at the very top stage, where  $X$  is defined exactly as  
 464 in the discussion of Section 2.10. We aim to choose all these parameters so that  
 465 the network does a good job on classifying the training data correctly. Using the  
 466 notation  $w$  for the hidden layer transformations, that is,

$$467 \quad (2.11.2) \quad w := (W^1, g^1, W^2, g^2, \dots, W^D, g^D),$$

468 and defining  $X := \{x_{[k]} \mid k = 1, 2, \dots, M\}$  as in Section 2.10, we can write the loss  
 469 function for deep learning as follows:

$$470 \quad (2.11.3) \quad L(w, X) := \frac{1}{m} \sum_{j=1}^m \left[ \sum_{\ell=1}^M y_{j\ell} (x_{[\ell]}^T a_j^D(w)) - \log \left( \sum_{\ell=1}^M \exp(x_{[\ell]}^T a_j^D(w)) \right) \right].$$

471 Note that this is exactly the function (2.10.8) applied to the output of the top  
 472 hidden layer  $a_j^D(w)$ . We write  $a_j^D(w)$  to make explicit the dependence of  $a_j^D$   
 473 on the parameters  $w$  of (2.11.2), as well as on the input vector  $a_j$ . (We can view  
 474 multiclass logistic regression (2.10.8) as a special case of deep learning in which  
 475 there are no hidden layers, so that  $D = 0$ ,  $w$  is null, and  $a_j^D = a_j$ ,  $j = 1, 2, \dots, m$ .)

476 Neural networks in use for particular applications (in image recognition and  
 477 speech recognition, for example, where they have been very successful) include  
 478 many variants on the basic design above. These include restricted connectivity  
 479 between layers (that is, enforcing structure on the matrices  $W^l$ ,  $l = 1, 2, \dots, D$ ),  
 480 layer arrangements that are more complex than the linear layout illustrated in  
 481 Figure 2.11.1, with outputs coming from different levels, connections across non-  
 482 adjacent layers, different componentwise transformations  $\sigma$  at different layers,  
 483 and so on. Deep neural networks for practical applications are highly engineered  
 484 objects.

485 The loss function (2.11.3) shares with many other applications the “summation”  
 486 form (2.1.2), but it has several features that set it apart from the other applications  
 487 discussed above. First, and possibly most important, it is *nonconvex* in the param-  
 488 eters  $w$ . There is reason to believe that the “landscape” of  $L$  is complex, with the  
 489 global minimizer being exceedingly difficult to find. Second, the total number  
 490 of parameters in  $(w, X)$  is usually very large. The most popular algorithms for  
 491 minimizing (2.11.3) are of stochastic gradient type, which like most optimization  
 492 methods come with no guarantee for finding the minimizer of a nonconvex func-  
 493 tion. Effective training of deep learning classifiers typically requires a great deal  
 494 of data and computation power. Huge clusters of powerful computers, often us-  
 495 ing multicore processors, GPUs, and even specially architected processing units,  
 496 are devoted to this task. Efficiency also requires many heuristics in the formula-  
 497 tion and the algorithm (for example, in the choice of regularization functions and  
 498 in the steplengths for stochastic gradient).

### 499 3. Preliminaries

500 We discuss here some foundations for the analysis of subsequent sections.  
 501 These include useful facts about smooth and nonsmooth convex functions, Tay-  
 502 lor’s theorem and some of its consequences, optimality conditions, and proximal  
 503 operators.

504 In the discussion of this section, our basic assumption is that  $f$  is a mapping  
 505 from  $\mathbb{R}^n$  to  $\mathbb{R} \cup \{+\infty\}$ , continuous on its effective domain  $D := \{x \mid f(x) < \infty\}$ .  
 506 Further assumptions of  $f$  are introduced as needed.

507 **3.1. Solutions** Consider the problem of minimizing  $f$  (1.0.1). We have the fol-  
 508 lowing terminology:

- 509 •  $x^*$  is a *local minimizer* of  $f$  if there is a neighborhood  $N$  of  $x^*$  such that  
 510  $f(x) \geq f(x^*)$  for all  $x \in N$ .
- 511 •  $x^*$  is a *global minimizer* of  $f$  if  $f(x) \geq f(x^*)$  for all  $x \in \mathbb{R}^n$ .
- 512 •  $x^*$  is a *strict local minimizer* if it is a local minimizer on some neighborhood  
 513  $N$  and in addition  $f(x) > f(x^*)$  for all  $x \in N$  with  $x \neq x^*$ .
- 514 •  $x^*$  is an *isolated local minimizer* if there is a neighborhood  $N$  of  $x^*$  such that  
 515  $f(x) \geq f(x^*)$  for all  $x \in N$  and in addition,  $N$  contains no local minimizers  
 516 other than  $x^*$ .

517 **3.2. Convexity and Subgradients** A convex set  $\Omega \subset \mathbb{R}^n$  has the property that

$$518 \quad (3.2.1) \quad x, y \in \Omega \Rightarrow (1 - \alpha)x + \alpha y \in \Omega \text{ for all } \alpha \in [0, 1].$$

519 We usually deal with *closed* convex sets in this article. For a convex set  $\Omega \subset \mathbb{R}^n$   
 520 we define the *indicator function*  $I_\Omega(x)$  as follows:

$$521 \quad I_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega \\ +\infty & \text{otherwise.} \end{cases}$$

522 Indicator functions are useful devices for deriving optimality conditions for con-  
 523 strained problems, and even for developing algorithms. The constrained opti-  
 524 mization problem

$$525 \quad (3.2.2) \quad \min_{x \in \Omega} f(x)$$

526 can be restated equivalently as follows:

$$527 \quad (3.2.3) \quad \min f(x) + I_\Omega(x).$$

528 We noted already that a convex function  $\phi : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  has the following  
 529 defining property:

$$530 \quad (3.2.4) \quad \phi((1 - \alpha)x + \alpha y) \leq (1 - \alpha)\phi(x) + \alpha\phi(y), \quad \text{for all } x, y \in \mathbb{R}^n \text{ and all } \alpha \in [0, 1].$$

531 The concepts of “minimizer” are simpler in the case of convex objective func-  
 532 tions than in the general case. In particular, the distinction between “local” and  
 533 “global” minimizers disappears. For  $f$  convex in (1.0.1), we have the following.

- 534 (a) Any local minimizer of (1.0.1) is also a global minimizer.
- 535 (b) The set of global minimizers of (1.0.1) is a convex set.

536 If there exists a value  $\gamma > 0$  such that

$$537 \quad (3.2.5) \quad \phi((1 - \alpha)x + \alpha y) \leq (1 - \alpha)\phi(x) + \alpha\phi(y) - \frac{1}{2}\gamma\alpha(1 - \alpha)\|x - y\|_2^2$$

538 for all  $x$  and  $y$  in the domain of  $\phi$  and  $\alpha \in [0, 1]$ , we say that  $\phi$  is *strongly convex*  
 539 *with modulus of convexity*  $\gamma$ .

540 We summarize some definitions and results about subgradients of convex func-  
 541 tions here. For a more extensive discussion, see [22].

542 **Definition 3.2.6.** A vector  $v \in \mathbb{R}^n$  is a *subgradient* of  $f$  at a point  $x$  if

$$543 \quad f(x + d) \geq f(x) + v^T d. \quad \text{for all } d \in \mathbb{R}^n.$$

544 The *subdifferential*, denoted  $\partial f(x)$ , is the set of all subgradients of  $f$  at  $x$ .

545 Subdifferentials satisfy a *monotonicity* property, as we show now.

546 **Lemma 3.2.7.** If  $a \in \partial f(x)$  and  $b \in \partial f(y)$ , we have  $(a - b)^T(x - y) \geq 0$ .

547 *Proof.* From the convexity of  $f$  and the definitions of  $a$  and  $b$ , we deduce that  
 548  $f(y) \geq f(x) + a^T(y - x)$  and  $f(x) \geq f(y) + b^T(x - y)$ . The result follows by adding  
 549 these two inequalities.  $\square$

550 We can easily characterize a minimum in terms of the subdifferential.

551 **Theorem 3.2.8.** The point  $x^*$  is the minimizer of a convex function  $f$  if and only if  
 552  $0 \in \partial f(x^*)$ .

553 *Proof.* Suppose that  $0 \in \partial f(x^*)$ , we have by substituting  $x = x^*$  and  $v = 0$  into  
 554 Definition 3.2.6 that  $f(x^* + d) \geq f(x^*)$  for all  $d \in \mathbb{R}^n$ , which implies that  $x^*$  is a  
 555 minimizer of  $f$ .

556 The converse follows trivially by showing that  $v = 0$  satisfies Definition 3.2.6  
 557 when  $x^*$  is a minimizer.  $\square$

558 The subdifferential is the generalization to nonsmooth convex functions of the  
 559 concept of derivative of a smooth function.

560 **Theorem 3.2.9.** If  $f$  is convex and differentiable at  $x$ , then  $\partial f(x) = \{\nabla f(x)\}$ .

561 A converse of this result is also true. Specifically, if the subdifferential of a  
 562 convex function  $f$  at  $x$  contains a single subgradient, then  $f$  is differentiable with  
 563 gradient equal to this subgradient (see [40, Theorem 25.1]).

564 **3.3. Taylor's Theorem** Taylor's theorem is a foundational result for optimization  
 565 of smooth nonlinear functions. It shows how smooth functions can be approxi-  
 566 mated locally by low-order (linear or quadratic) functions.

567 **Theorem 3.3.1.** Given a continuously differentiable function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , and given  
 568  $x, p \in \mathbb{R}^n$ , we have that

$$(3.3.2) \quad f(x + p) = f(x) + \int_0^1 \nabla f(x + \xi p)^T p \, d\xi,$$

$$(3.3.3) \quad f(x + p) = f(x) + \nabla f(x + \xi p)^T p, \quad \text{some } \xi \in (0, 1).$$

569 If  $f$  is twice continuously differentiable, we have

$$(3.3.4) \quad \nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 f(x + \xi p) p \, d\xi,$$

$$(3.3.5) \quad f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + \xi p) p, \quad \text{for some } \xi \in (0, 1).$$

570 We can derive an important consequence of this theorem when  $f$  is *Lipschitz*  
571 continuously differentiable with constant  $L$ , that is,

$$572 \quad (3.3.6) \quad \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \text{for all } x, y \in \mathbb{R}^n.$$

573 We have by setting  $y = x + p$  in (3.3.2) and subtracting the term  $\nabla f(x)^\top(y - x)$   
574 from both sides that

$$575 \quad f(y) - f(x) - \nabla f(x)^\top(y - x) = \int_0^1 [\nabla f(x + \xi(y - x)) - \nabla f(x)]^\top(y - x) \, d\xi.$$

By using (3.3.6), we have

$$\begin{aligned} [\nabla f(x + \xi(y - x)) - \nabla f(x)]^\top(y - x) &\leq \|\nabla f(x + \xi(y - x)) - \nabla f(x)\| \|y - x\| \\ &\leq L\xi \|y - x\|^2. \end{aligned}$$

576 By substituting this bound into the previous integral, we obtain

$$577 \quad (3.3.7) \quad f(y) - f(x) - \nabla f(x)^\top(y - x) \leq \frac{L}{2} \|y - x\|^2.$$

578 For the remainder of Section 3.3, we assume that  $f$  is continuously differ-  
579 entiable and also *convex*. The definition of convexity (3.2.4) and the fact that  
580  $\partial f(x) = \{\nabla f(x)\}$  implies that

$$581 \quad (3.3.8) \quad f(y) \geq f(x) + \nabla f(x)^\top(y - x), \quad \text{for all } x, y \in \mathbb{R}^n.$$

582 We defined “strong convexity with modulus  $\gamma$ ” in (3.2.5). When  $f$  is differentiable,  
583 we have the following equivalent definition, obtained by rearranging (3.2.5) and  
584 letting  $\alpha \downarrow 0$ .

$$585 \quad (3.3.9) \quad f(y) \geq f(x) + \nabla f(x)^\top(y - x) + \frac{\gamma}{2} \|y - x\|^2.$$

586 By combining this expression with (3.3.7), we have the following result.

587 **Lemma 3.3.10.** *Given convex  $f$  satisfying (3.2.5), with  $\nabla f$  uniformly Lipschitz continu-*  
588 *ous with constant  $L$ , we have for any  $x, y$  that*

$$589 \quad (3.3.11) \quad \frac{\gamma}{2} \|y - x\|^2 \leq f(y) - f(x) - \nabla f(x)^\top(y - x) \leq \frac{L}{2} \|y - x\|^2.$$

590 For later convenience, we define a *condition number*  $\kappa$  as follows:

$$591 \quad (3.3.12) \quad \kappa := \frac{L}{\gamma}.$$

592 When  $f$  is *twice* continuously differentiable, we can characterize the constants  $\gamma$   
593 and  $L$  in terms of the eigenvalues of the Hessian  $\nabla^2 f(x)$ . Specifically, we can show  
594 that (3.3.11) is equivalent to

$$595 \quad (3.3.13) \quad \gamma I \preceq \nabla^2 f(x) \preceq LI, \quad \text{for all } x.$$

596 When  $f$  is strictly convex and quadratic,  $\kappa$  defined in (3.3.12) is the condition  
597 number of the (constant) Hessian, in the usual sense of linear algebra.

598 Strongly convex functions have unique minimizers, as we now show.

599 **Theorem 3.3.14.** *Let  $f$  be differentiable and strongly convex with modulus  $\gamma > 0$ . Then*  
600 *the minimizer  $x^*$  of  $f$  exists and is unique.*

601 *Proof.* We show first that for any point  $x^0$ , the level set  $\{x \mid f(x) \leq f(x^0)\}$  is closed  
 602 and bounded, and hence compact. Suppose for contradiction that there is a se-  
 603 quence  $\{x^\ell\}$  such that  $\|x^\ell\| \rightarrow \infty$  and

$$604 \quad (3.3.15) \quad f(x^\ell) \leq f(x^0).$$

605 By strong convexity of  $f$ , we have for some  $\gamma > 0$  that

$$606 \quad f(x^\ell) \geq f(x^0) + \nabla f(x^0)^\top (x^\ell - x^0) + \frac{\gamma}{2} \|x^\ell - x^0\|^2.$$

607 By rearranging slightly, and using (3.3.15), we obtain

$$608 \quad \frac{\gamma}{2} \|x^\ell - x^0\|^2 \leq -\nabla f(x^0)^\top (x^\ell - x^0) \leq \|\nabla f(x^0)\| \|x^\ell - x^0\|.$$

609 By dividing both sides by  $(\gamma/2)\|x^\ell - x^0\|$ , we obtain  $\|x^\ell - x^0\| \leq (2/\gamma)\|\nabla f(x^0)\|$   
 610 for all  $\ell$ , which contradicts unboundedness of  $\{x^\ell\}$ . Thus, the level set is bounded.  
 611 Since it is also closed (by continuity of  $f$ ), it is compact.

612 Since  $f$  is continuous, it attains its minimum on the compact level set, which is  
 613 also the solution of  $\min_x f(x)$ , and we denote it by  $x^*$ . Suppose for contradiction  
 614 that the minimizer is not unique, so that we have two points  $x_1^*$  and  $x_2^*$  that  
 615 minimize  $f$ . Obviously, these points must attain equal objective values, so that  
 616  $f(x_1^*) = f(x_2^*) = f^*$  for some  $f^*$ . By taking (3.2.5) and setting  $\phi = f^*$ ,  $x = x_1^*$ ,  
 617  $y = x_2^*$ , and  $\alpha = 1/2$ , we obtain

$$618 \quad f((x_1^* + x_2^*)/2) \leq \frac{1}{2}(f(x_1^*) + f(x_2^*)) - \frac{1}{8}\gamma\|x_1^* - x_2^*\|^2 < f^*,$$

619 so the point  $(x_1^* + x_2^*)/2$  has a smaller function value than both  $x_1^*$  and  $x_2^*$ , contra-  
 620 dicting our assumption that  $x_1^*$  and  $x_2^*$  are both minimizers. Hence, the minimizer  
 621  $x^*$  is unique.  $\square$

622 **3.4. Optimality Conditions for Smooth Functions** We consider the case of a  
 623 smooth (twice continuously differentiable) function  $f$  that is not necessarily con-  
 624 vex. Before designing algorithms to find a minimizer of  $f$ , we need to identify  
 625 properties of  $f$  and its derivatives at a point  $\bar{x}$  that tell us whether or not  $\bar{x}$  is a  
 626 minimizer, of one of the types described in Subsection 3.1. We call such properties  
 627 *optimality conditions*.

628 A *first-order necessary* condition for optimality is that  $\nabla f(\bar{x}) = 0$ . More precisely,  
 629 if  $\bar{x}$  is a local minimizer, then  $\nabla f(\bar{x}) = 0$ . We can prove this by using Taylor's  
 630 theorem. Supposing for contradiction that  $\nabla f(\bar{x}) \neq 0$ , we can show by setting  
 631  $x = \bar{x}$  and  $p = -\alpha \nabla f(\bar{x})$  for  $\alpha > 0$  in (3.3.3) that  $f(\bar{x} - \alpha \nabla f(\bar{x})) < f(\bar{x})$  for all  
 632  $\alpha > 0$  sufficiently small. Thus *any* neighborhood of  $\bar{x}$  will contain points  $x$  with a  
 633  $f(x) < f(\bar{x})$ , so  $\bar{x}$  cannot be a local minimizer.

634 If  $f$  is convex, as well as smooth, the condition  $\nabla f(\bar{x}) = 0$  is *sufficient* for  $\bar{x}$  to be  
 635 a *global* solution. This claim follows immediately from Theorems 3.2.8 and 3.2.9.

636 A *second-order necessary* condition for  $\bar{x}$  to be a local solution is that  $\nabla f(\bar{x}) = 0$   
 637 and  $\nabla^2 f(\bar{x})$  is positive semidefinite. The proof is by an argument similar to that  
 638 of the first-order necessary condition, but using the second-order Taylor series ex-  
 639 pansion (3.3.5) instead of (3.3.3). A *second-order sufficient* condition is that  $\nabla f(\bar{x}) = 0$

640 and  $\nabla^2 f(\bar{x})$  is positive definite. This condition guarantees that  $\bar{x}$  is a *strict* local  
 641 minimizer, that is, there is a neighborhood of  $\bar{x}$  such that  $\bar{x}$  has a strictly smaller  
 642 function value than all other points in this neighborhood. Again, the proof makes  
 643 use of (3.3.5).

644 We call  $\bar{x}$  a *stationary point* for smooth  $f$  if it satisfies the first-order necessary  
 645 condition  $\nabla f(\bar{x}) = 0$ . Stationary points are not necessarily local minimizers. In  
 646 fact, local *maximizers* satisfy the same condition. More interestingly, stationary  
 647 points can be *saddle points*. These are points for which there exist directions  $u$   
 648 and  $v$  such that  $f(\bar{x} + \alpha u) < f(\bar{x})$  and  $f(\bar{x} + \alpha v) > f(\bar{x})$  for all positive  $\alpha$  suffi-  
 649 ciently small. When the Hessian  $\nabla^2 f(\bar{x})$  has both strictly positive and strictly  
 650 negative eigenvalues, it follows from (3.3.5) that  $\bar{x}$  is a saddle point. When  $\nabla^2 f(\bar{x})$   
 651 is positive semidefinite or negative semidefinite, second derivatives alone are in-  
 652 sufficient to classify  $\bar{x}$ ; higher-order derivative information is needed.

653 **3.5. Proximal Operators and the Moreau Envelope** Here we present some anal-  
 654 ysis for analyzing the convergence of algorithms for the regularized problem  
 655 (1.0.2), where the objective is the sum of a smooth function and a convex (usually  
 656 nonsmooth) function.

657 We start with a formal definition.

658 **Definition 3.5.1.** For a closed proper convex function  $h$  and a positive scalar  $\lambda$ ,  
 659 the *Moreau envelope* is

$$660 \quad (3.5.2) \quad M_{\lambda,h}(x) := \inf_u \left\{ h(u) + \frac{1}{2\lambda} \|u - x\|^2 \right\} = \frac{1}{\lambda} \inf_u \left\{ \lambda h(u) + \frac{1}{2} \|u - x\|^2 \right\}.$$

661 The proximal operator of the function  $\lambda h$  is the value of  $u$  that achieves the infi-  
 662 mum in (3.5.2), that is,

$$663 \quad (3.5.3) \quad \text{prox}_{\lambda h}(x) := \arg \min_u \left\{ \lambda h(u) + \frac{1}{2} \|u - x\|^2 \right\}.$$

664 From optimality properties for (3.5.3) (see Theorem 3.2.8), we have

$$665 \quad (3.5.4) \quad 0 \in \lambda \partial h(\text{prox}_{\lambda h}(x)) + (\text{prox}_{\lambda h}(x) - x).$$

666 The Moreau envelope can be viewed as a kind of smoothing or regularization  
 667 of the function  $h$ . It has a finite value for all  $x$ , even when  $h$  takes on infinite  
 668 values for some  $x \in \mathbb{R}^n$ . In fact, it is differentiable everywhere, with gradient

$$669 \quad \nabla M_{\lambda,h}(x) = \frac{1}{\lambda} (x - \text{prox}_{\lambda h}(x)).$$

670 Moreover,  $x^*$  is a minimizer of  $h$  if and only if it is a minimizer of  $M_{\lambda,h}$ .

671 The proximal operator satisfies a nonexpansiveness property. From the opti-  
 672 mality conditions (3.5.4) at two points  $x$  and  $y$ , we have

$$673 \quad x - \text{prox}_{\lambda h}(x) \in \lambda \partial h(\text{prox}_{\lambda h}(x)), \quad y - \text{prox}_{\lambda h}(y) \in \lambda \partial h(\text{prox}_{\lambda h}(y)).$$

674 By applying monotonicity (Lemma 3.2.7), we have

$$675 \quad (1/\lambda)((x - \text{prox}_{\lambda h}(x)) - (y - \text{prox}_{\lambda h}(y)))^\top (\text{prox}_{\lambda h}(x) - \text{prox}_{\lambda h}(y)) \geq 0,$$

Rearranging this and applying the Cauchy-Schwartz inequality yields

$$\begin{aligned} \|\text{prox}_{\lambda h}(x) - \text{prox}_{\lambda h}(y)\|^2 &\leq (x - y)^\top (\text{prox}_{\lambda h}(x) - \text{prox}_{\lambda h}(y)) \\ &\leq \|x - y\| \|\text{prox}_{\lambda h}(x) - \text{prox}_{\lambda h}(y)\|, \end{aligned}$$

676 from which we obtain  $\|\text{prox}_{\lambda h}(x) - \text{prox}_{\lambda h}(y)\| \leq \|x - y\|$ , as claimed.

677 We list the prox operator for several instances of  $h$  that are common in data  
678 analysis applications. These definitions are useful in implementing the prox-  
679 gradient algorithms of Section 5.

- 680 •  $h(x) = 0$  for all  $x$ , for which we have  $\text{prox}_{\lambda h}(x) = 0$ . (This observation is  
681 useful in proving that the prox-gradient method reduces to the familiar  
682 steepest descent method when the objective contains no regularization  
683 term.)
- 684 •  $h(x) = I_\Omega(x)$ , the indicator function for a closed convex set  $\Omega$ . In this  
685 case, we have for any  $\lambda > 0$  that

$$686 \quad \text{prox}_{\lambda I_\Omega}(x) = \arg \min_u \left\{ \lambda I_\Omega(u) + \frac{1}{2} \|u - x\|^2 \right\} = \arg \min_{u \in \Omega} \frac{1}{2} \|u - x\|^2,$$

687 which is simply the projection of  $x$  onto the set  $\Omega$ .

- 688 •  $h(x) = \|x\|_1$ . By substituting into definition (3.5.3) we see that the mini-  
689 mization separates into its  $n$  separate components, and that the  $i$ th com-  
690 ponent of  $\text{prox}_{\lambda \|\cdot\|_1}(x)$  is

$$691 \quad \left[ \text{prox}_{\lambda \|\cdot\|_1}(x) \right]_i = \arg \min_{u_i} \left\{ \lambda |u_i| + \frac{1}{2} (u_i - x_i)^2 \right\}.$$

692 We can thus verify that

$$693 \quad (3.5.5) \quad \left[ \text{prox}_{\lambda \|\cdot\|_1}(x) \right]_i = \begin{cases} x_i - \lambda & \text{if } x_i > \lambda; \\ 0 & \text{if } x_i \in [-\lambda, \lambda]; \\ x_i + \lambda & \text{if } x_i < -\lambda, \end{cases}$$

694 an operation that is known as *soft-thresholding*.

- 695 •  $h(x) = \|x\|_0$ , where  $\|x\|_0$  denotes the *cardinality* of the vector  $x$ , its number  
696 of nonzero components. Although this  $h$  is not a convex function (as  
697 we can see by considering convex combinations of the vectors  $(0, 1)^\top$  and  
698  $(1, 0)^\top$  in  $\mathbb{R}^2$ ), its proximal operator is well defined, and is known as *hard*  
699 *thresholding*:

$$700 \quad \left[ \text{prox}_{\lambda \|\cdot\|_0}(x) \right]_i = \begin{cases} x_i & \text{if } |x_i| \geq \sqrt{2\lambda}; \\ 0 & \text{if } |x_i| < \sqrt{2\lambda}. \end{cases}$$

701 As in (3.5.5), the definition (3.5.3) separates into  $n$  individual components.

702 **3.6. Convergence Rates** An important measure for evaluating algorithms is the  
703 rate of convergence to zero of some measure of error. For smooth  $f$ , we may be  
704 interested in how rapidly the sequence of gradient norms  $\{\|\nabla f(x^k)\|\}$  converges  
705 to zero. For nonsmooth convex  $f$ , a measure of interest may be convergence to

706 zero of  $\{\text{dist}(0, \partial f(x^k))\}$  (the sequence of distances from 0 to the subdifferential  
 707  $\partial f(x^k)$ ). Other error measures for which we may be able to prove convergence  
 708 rates include  $\|x^k - x^*\|$  (where  $x^*$  is a solution) and  $f(x^k) - f^*$  (where  $f^*$  is the  
 709 optimal value of the objective function  $f$ ). For generality, we denote by  $\{\phi_k\}$  the  
 710 sequence of nonnegative scalars whose rate of convergence to 0 we wish to find.

711 We say that *linear* convergence holds if there is some  $\sigma \in (0, 1)$  such that

$$712 \quad (3.6.1) \quad \phi_{k+1}/\phi_k \leq 1 - \sigma, \quad \text{for all } k \text{ sufficiently large.}$$

713 (This property is sometimes also called *geometric* or *exponential* convergence, but  
 714 the term *linear* is standard in the optimization literature, so we use it here.) It  
 715 follows from (3.6.1) that there is some positive constant  $C$  such that

$$716 \quad (3.6.2) \quad \phi_k \leq C(1 - \sigma)^k, \quad k = 1, 2, \dots$$

717 While (3.6.1) implies (3.6.2), the converse does not hold. The sequence

$$718 \quad \phi_k = \begin{cases} 2^{-k} & k \text{ even} \\ 0 & k \text{ odd,} \end{cases}$$

719 satisfies (3.6.2) with  $C = 1$  and  $\sigma = .5$ , but does not satisfy (3.6.1). To distinguish  
 720 between these two slightly different definitions, (3.6.1) is sometimes called *Q-*  
 721 *linear* while (3.6.2) is called *R-linear*.

*Sublinear* convergence is, as its name suggests, slower than linear. Several  
 varieties of sublinear convergence are encountered in optimization algorithms  
 for data analysis, including the following

$$(3.6.3a) \quad \phi_k \leq C/\sqrt{k}, \quad k = 1, 2, \dots,$$

$$(3.6.3b) \quad \phi_k \leq C/k, \quad k = 1, 2, \dots,$$

$$(3.6.3c) \quad \phi_k \leq C/k^2, \quad k = 1, 2, \dots,$$

722 where in each case,  $C$  is some positive constant.

723 Superlinear convergence occurs when the constant  $\sigma \in (0, 1)$  in (3.6.1) can be  
 724 chosen arbitrarily close to 1. Specifically, we say that the sequence  $\{\phi_k\}$  converges  
 725 *Q-superlinearly* to 0 if

$$726 \quad (3.6.4) \quad \lim_{k \rightarrow \infty} \phi_{k+1}/\phi_k = 0.$$

727 *Q-Quadratic* convergence occurs when

$$728 \quad (3.6.5) \quad \phi_{k+1}/\phi_k^2 \leq C, \quad k = 1, 2, \dots,$$

729 for some sufficiently large  $C$ . We say that the convergence is *R-superlinear* if  
 730 there is a *Q-superlinearly* convergent sequence  $\{\nu_k\}$  that dominates  $\{\phi_k\}$  (that is,  
 731  $0 \leq \phi_k \leq \nu_k$  for all  $k$ ). *R-quadratic* convergence is defined similarly. Quadratic  
 732 and superlinear rates are associated with higher-order methods, such as Newton  
 733 and quasi-Newton methods.

734 When a convergence rate applies *globally*, from any reasonable starting point,  
 735 it can be used to derive a complexity bound for the algorithm, which takes the

736 form of a bound on the number of iterations  $K$  required to reduce  $\phi_k$  below  
 737 some specified tolerance  $\epsilon$ . For a sequence satisfying the R-linear convergence  
 738 condition (3.6.2) a sufficient condition for  $\phi_K \leq \epsilon$  is  $C(1 - \sigma)^K \leq \epsilon$ . By using the  
 739 estimate  $\log(1 - \sigma) \leq -\sigma$  for all  $\sigma \in (0, 1)$ , we have that

$$740 \quad C(1 - \sigma)^K \leq \epsilon \Leftrightarrow K \log(1 - \sigma) \leq \log(\epsilon/C) \Leftrightarrow K \geq \log(C/\epsilon)/\sigma.$$

741 It follows that for linearly convergent algorithms, the number of iterations re-  
 742 quired to converge to a tolerance  $\epsilon$  depends logarithmically on  $1/\epsilon$  and inversely  
 743 on the rate constant  $\sigma$ . For an algorithm that satisfies the sublinear rate (3.6.3a), a  
 744 sufficient condition for  $\phi_K \leq \epsilon$  is  $C/\sqrt{K} \leq \epsilon$ , which is equivalent to  $K \geq (C/\epsilon)^2$ ,  
 745 so the complexity is  $O(1/\epsilon^2)$ . Similar analyses for (3.6.3b) reveal complexity of  
 746  $O(1/\epsilon)$ , while for (3.6.3c), we have complexity  $O(1/\sqrt{\epsilon})$ .

747 For quadratically convergent methods, the complexity is doubly logarithmic  
 748 in  $\epsilon$  (that is,  $O(\log \log(1/\epsilon))$ ). Once the algorithm enters a neighborhood of qua-  
 749 dratic convergence, just a few additional iterations are required for convergence  
 750 to a solution of high accuracy.

## 751 4. Gradient Methods

752 We consider here iterative methods for solving the unconstrained smooth prob-  
 753 lem (1.0.1) that make use of the gradient  $\nabla f$  (see also, [22] which describes sub-  
 754 gradient methods for nonsmooth convex functions.) We consider mostly methods  
 755 that generate an iteration sequence  $\{x^k\}$  via the formula

$$756 \quad (4.0.1) \quad x^{k+1} = x^k + \alpha_k d^k,$$

757 where  $d^k$  is the search direction and  $\alpha_k$  is a steplength.

758 We consider the steepest descent method, which searches along the negative  
 759 gradient direction  $d^k = -\nabla f(x^k)$ , proving convergence results for nonconvex  
 760 functions, convex functions, and strongly convex functions. In Subsection 4.5, we  
 761 consider methods that use more general descent directions  $d^k$ , proving conver-  
 762 gence of methods that make careful choices of the line search parameter  $\alpha_k$  at  
 763 each iteration. In Subsection 4.6, we consider the conditional gradient method for  
 764 minimization of a smooth function  $f$  over a compact set.

765 **4.1. Steepest Descent** The simplest stepsize protocol is the short-step variant  
 766 of steepest descent. We assume here that  $f$  is differentiable, with gradient  $\nabla f$   
 767 satisfying the Lipschitz continuity condition (3.3.6) with constant  $L$ . We choose  
 768 the search direction  $d^k = -\nabla f(x^k)$  in (4.0.1), and set the steplength  $\alpha_k$  to be the  
 769 constant  $1/L$ , to obtain the iteration

$$770 \quad (4.1.1) \quad x^{k+1} = x^k - \frac{1}{L} \nabla f(x^k), \quad k = 0, 1, 2, \dots$$

771 To estimate the amount of decrease in  $f$  obtained at each iterate of this method,  
 772 we use Taylor's theorem. From (3.3.7), we have

$$773 \quad (4.1.2) \quad f(x + \alpha d) \leq f(x) + \alpha \nabla f(x)^T d + \alpha^2 \frac{L}{2} \|d\|^2,$$

774 For  $x = x^k$  and  $d = -\nabla f(x^k)$ , the value of  $\alpha$  that minimizes the expression on the  
 775 right-hand side is  $\alpha = 1/L$ . By substituting these values, we obtain

$$776 \quad (4.1.3) \quad f(x^{k+1}) = f(x^k - (1/L)\nabla f(x^k)) \leq f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|^2.$$

777 This expression is one of the foundational inequalities in the analysis of optimiza-  
 778 tion methods. Depending on the assumptions about  $f$ , we can derive a variety of  
 779 different convergence rates from this basic inequality.

780 **4.2. General Case** We consider first a function  $f$  that is Lipschitz continuously  
 781 differentiable and bounded below, but that need not necessarily be convex. Using  
 782 (4.1.3) alone, we can prove a sublinear convergence result for the steepest descent  
 783 method.

784 **Theorem 4.2.1.** *Suppose that  $f$  is Lipschitz continuously differentiable, satisfying (3.3.6),  
 785 and that  $f$  is bounded below by a constant  $\bar{f}$ . Then for the steepest descent method with  
 786 constant steplength  $\alpha_k \equiv 1/L$ , applied from a starting point  $x^0$ , we have for any integer  
 787  $T \geq 1$  that*

$$788 \quad \min_{0 \leq k \leq T-1} \|\nabla f(x^k)\| \leq \sqrt{\frac{2L[f(x^0) - f(x^T)]}{T}} \leq \sqrt{\frac{2L[f(x^0) - \bar{f}]}{T}}.$$

789 *Proof.* Rearranging (4.1.3) and summing over the first  $T - 1$  iterates, we have

$$790 \quad (4.2.2) \quad \sum_{k=0}^{T-1} \|\nabla f(x^k)\|^2 \leq 2L \sum_{k=0}^{T-1} [f(x^k) - f(x^{k+1})] = 2L[f(x^0) - f(x^T)].$$

791 (Note the telescoping sum.) Since  $f$  is bounded below by  $\bar{f}$ , the right-hand side is  
 792 bounded above by the constant  $2L[f(x^0) - \bar{f}]$ . We also have that

$$793 \quad \min_{0 \leq k \leq T-1} \|\nabla f(x^k)\| = \sqrt{\min_{0 \leq k \leq T-1} \|\nabla f(x^k)\|^2} \leq \sqrt{\frac{1}{T} \sum_{k=0}^{T-1} \|\nabla f(x^k)\|^2}.$$

794 The result is obtained by combining this bound with (4.2.2).  $\square$

795 This result shows that within the first  $T - 1$  steps of steepest descent, at least  
 796 one of the iterates has gradient norm less than  $\sqrt{2L[f(x^0) - \bar{f}]/T}$ , which repre-  
 797 sents sublinear convergence of type (3.6.3a). It follows too from (4.2.2) that for  $f$   
 798 bounded below, any accumulation point of the sequence  $\{x^k\}$  is stationary.

799 **4.3. Convex Case** When  $f$  is also convex, we have the following stronger result  
 800 for the steepest descent method.

801 **Theorem 4.3.1.** *Suppose that  $f$  is convex and Lipschitz continuously differentiable, sat-  
 802 isfying (3.3.6), and that (1.0.1) has a solution  $x^*$ . Then the steepest descent method with  
 803 stepsize  $\alpha_k \equiv 1/L$  generates a sequence  $\{x^k\}_{k=0}^{\infty}$  that satisfies*

$$804 \quad (4.3.2) \quad f(x^T) - f^* \leq \frac{L}{2T} \|x^0 - x^*\|^2.$$

*Proof.* By convexity of  $f$ , we have  $f(x^*) \geq f(x^k) + \nabla f(x^k)^\top (x^* - x^k)$ , so by substituting into (4.1.3), we obtain for  $k = 0, 1, 2, \dots$  that

$$\begin{aligned} f(x^{k+1}) &\leq f(x^*) + \nabla f(x^k)^\top (x^k - x^*) - \frac{1}{2L} \|\nabla f(x^k)\|^2 \\ &= f(x^*) + \frac{L}{2} \left( \|x^k - x^*\|^2 - \left\| x^k - x^* - \frac{1}{L} \nabla f(x^k) \right\|^2 \right) \\ &= f(x^*) + \frac{L}{2} \left( \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 \right). \end{aligned}$$

By summing over  $k = 0, 1, 2, \dots, T-1$ , and noting the telescoping sum, we have

$$\begin{aligned} \sum_{k=0}^{T-1} (f(x^{k+1}) - f^*) &\leq \frac{L}{2} \sum_{k=0}^{T-1} \left( \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 \right) \\ &= \frac{L}{2} \left( \|x^0 - x^*\|^2 - \|x^T - x^*\|^2 \right) \\ &\leq \frac{L}{2} \|x^0 - x^*\|^2. \end{aligned}$$

805 Since  $\{f(x^k)\}$  is a nonincreasing sequence, we have, as required,

$$806 \quad f(x^T) - f(x^*) \leq \frac{1}{T} \sum_{k=0}^{T-1} (f(x^{k+1}) - f^*) \leq \frac{L}{2T} \|x^0 - x^*\|^2. \quad \square$$

807 **4.4. Strongly Convex Case** Recall that the definition (3.3.9) of strong convexity  
808 shows that  $f$  can be bounded below by a quadratic with Hessian  $\gamma I$ . A strongly  
809 convex  $f$  with  $L$ -Lipschitz gradients is also bounded *above* by a similar quadratic  
810 (see (3.3.7)) differing only in the quadratic term, which becomes  $LI$ . From this  
811 “sandwich” effect, we derive a linear convergence rate for the gradient method,  
812 stated formally in the following theorem.

813 **Theorem 4.4.1.** *Suppose that  $f$  is Lipschitz continuously differentiable, satisfying (3.3.6),*  
814 *and strongly convex, satisfying (3.2.5) with modulus of convexity  $\gamma$ . Then  $f$  has a unique*  
815 *minimizer  $x^*$ , and the steepest descent method with stepsize  $\alpha_k \equiv 1/L$  generates a se-*  
816 *quence  $\{x^k\}_{k=0}^\infty$  that satisfies*

$$817 \quad f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\gamma}{L}\right) (f(x^k) - f(x^*)), \quad k = 0, 1, 2, \dots$$

*Proof.* Existence of the unique minimizer  $x^*$  follows from Theorem 3.3.14. Minimizing both sides of the inequality (3.3.9) with respect to  $y$ , we find that the minimizer on the left side is attained at  $y = x^*$ , while on the right side it is attained at  $x - \nabla f(x)/\gamma$ . Plugging these optimal values into (3.3.9), we obtain

$$\begin{aligned} \min_y f(y) &\geq \min_y f(x) + \nabla f(x)^\top (y - x) + \frac{\gamma}{2} \|y - x\|^2 \\ \Rightarrow f(x^*) &\geq f(x) - \nabla f(x)^\top \left( \frac{1}{\gamma} \nabla f(x) \right) + \frac{\gamma}{2} \left\| \frac{1}{\gamma} \nabla f(x) \right\|^2 \\ \Rightarrow f(x^*) &\geq f(x) - \frac{1}{2\gamma} \|\nabla f(x)\|^2. \end{aligned}$$

818 By rearrangement, we obtain

$$819 \quad (4.4.2) \quad \|\nabla f(x)\|^2 \geq 2\gamma[f(x) - f(x^*)].$$

820 By substituting (4.4.2) into our basic inequality (4.1.3), we obtain

$$821 \quad f(x^{k+1}) = f\left(x^k - \frac{1}{L}\nabla f(x^k)\right) \leq f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|^2 \leq f(x^k) - \frac{\gamma}{L}(f(x^k) - f^*).$$

822 Subtracting  $f^*$  from both sides of this inequality yields the result.  $\square$

823 Note that After  $T$  steps, we have

$$824 \quad (4.4.3) \quad f(x^T) - f^* \leq \left(1 - \frac{\gamma}{L}\right)^T (f(x^0) - f^*),$$

825 which is convergence of type (3.6.2) with constant  $\sigma = \gamma/L$ .

826 **4.5. General Case: Line-Search Methods** Returning to the case in which  $f$  has  
 827 Lipschitz continuous gradients but is possibly nonconvex, we consider algorithms  
 828 that take steps of the form (4.0.1), where  $d^k$  is a *descent direction*, that is, it  
 829 makes a positive inner product with the negative gradient  $-\nabla f(x^k)$ , so that  
 830  $\nabla f(x^k)^T d^k < 0$ . This condition ensures that  $f(x^k + \alpha d^k) < f(x^k)$  for sufficiently  
 831 small positive values of step length  $\alpha$  — we obtain improvement in  $f$  by taking  
 832 small steps along  $d^k$ . (This claim follows from (3.3.3).) *Line-search methods* are  
 833 built around this fundamental observation. By introducing additional conditions  
 834 on  $d^k$  and  $\alpha_k$ , that can be verified in practice with reasonable effort, we can estab-  
 835 lish a bound on decrease similar to (4.1.3) on each iteration, and thus a conclusion  
 836 similar to that of Theorem 4.2.1.

837 We assume that  $d^k$  satisfies the following for some  $\eta > 0$ :

$$838 \quad (4.5.1) \quad \nabla f(x^k)^T d^k \leq -\eta \|\nabla f(x^k)\| \|d^k\|.$$

For the steplength  $\alpha_k$ , we assume the following *weak Wolfe* conditions hold, for some constants  $c_1$  and  $c_2$  with  $0 < c_1 < c_2 < 1$ :

$$(4.5.2a) \quad f(x^k + \alpha_k d^k) \leq f(x^k) + c_1 \alpha_k \nabla f(x^k)^T d^k$$

$$(4.5.2b) \quad \nabla f(x^k + \alpha_k d^k)^T d^k \geq c_2 \nabla f(x^k)^T d^k.$$

839 Condition (4.5.2a) is called “sufficient decrease;” it ensures descent at each step of  
 840 at least a small fraction  $c_1$  of the amount promised by the first-order Taylor-series  
 841 expansion (3.3.3). Condition (4.5.2b) ensures that the directional derivative of  $f$   
 842 along the search direction  $d^k$  is significantly less negative at the chosen steplength  
 843  $\alpha_k$  than at  $\alpha = 0$ . This condition ensures that the step is “not too short.” It can  
 844 be shown that it is always possible to find  $\alpha_k$  that satisfies both conditions (4.5.2)  
 845 simultaneously.

846 Line-search procedures, which are specialized optimization procedures for  
 847 minimizing functions of one variable, have been devised to find such values effi-  
 848 ciently; see [36, Chapter 3] for details.

849 For line-search methods of this type, we have the following generalization of  
 850 Theorem 4.2.1.

851 **Theorem 4.5.3.** *Suppose that  $f$  is Lipschitz continuously differentiable, satisfying (3.3.6),*  
 852 *and that  $f$  is bounded below by a constant  $\bar{f}$ . Consider the method that takes steps of the*  
 853 *form (4.0.1), where  $d^k$  satisfies (4.5.1) for some  $\eta > 0$  and the conditions (4.5.2) hold at*  
 854 *all  $k$ , for some constants  $c_1$  and  $c_2$  with  $0 < c_1 < c_2 < 1$ . Then for any integer  $T \geq 1$ ,*  
 855 *we have*

$$856 \quad \min_{0 \leq k \leq T-1} \|\nabla f(x^k)\| \leq \sqrt{\frac{L}{\eta^2 c_1 (1 - c_2)}} \sqrt{\frac{f(x^0) - \bar{f}}{T}}.$$

857 *Proof.* By combining the Lipschitz property (3.3.6) with (4.5.2b), we have

$$858 \quad -(1 - c_2) \nabla f(x^k)^\top d^k \leq [\nabla f(x^k + \alpha_k d^k) - \nabla f(x^k)]^\top d^k \leq L \alpha_k \|d^k\|^2.$$

859 By comparing the first and last terms in these inequalities, we obtain the following  
 860 lower bound on  $\alpha_k$ :

$$861 \quad \alpha_k \geq -\frac{(1 - c_2) \nabla f(x^k)^\top d^k}{L \|d^k\|^2}.$$

862 By substituting this bound into (4.5.2a), and using (4.5.1) and the step definition  
 863 (4.0.1), we obtain

$$864 \quad \begin{aligned} f(x^{k+1}) &= f(x^k + \alpha_k d^k) \leq f(x^k) + c_1 \alpha_k \nabla f(x^k)^\top d^k \\ &\leq f(x^k) - \frac{c_1(1 - c_2)}{L} \frac{(\nabla f(x^k)^\top d^k)^2}{\|d^k\|^2} \\ &\leq f(x^k) - \frac{c_1(1 - c_2)}{L} \eta^2 \|\nabla f(x^k)\|^2, \end{aligned}$$

865 which by rearrangement yields

$$866 \quad (4.5.5) \quad \|\nabla f(x^k)\|^2 \leq \frac{L}{c_1(1 - c_2)\eta^2} (f(x^k) - f(x^{k+1})).$$

867 The result now follows as in the proof of Theorem 4.2.1.  $\square$

868 It follows by taking limits on both sides of (4.5.5) that

$$869 \quad (4.5.6) \quad \lim_{k \rightarrow \infty} \|\nabla f(x^k)\| = 0,$$

870 and therefore all accumulation points  $\bar{x}$  of the sequence  $\{x^k\}$  generated by the  
 871 algorithm (4.0.1) have  $\nabla f(\bar{x}) = 0$ . In the case of  $f$  convex, this condition guarantees  
 872 that  $\bar{x}$  is a solution of (1.0.1). When  $f$  is nonconvex,  $\bar{x}$  may be a local minimum,  
 873 but it may also be a saddle point or a local maximum.

874 The paper [29] uses the stable manifold theorem to show that line-search gra-  
 875 dient methods are highly unlikely to converge to stationary points  $\bar{x}$  at which  
 876 some eigenvalues of the Hessian  $\nabla^2 f(\bar{x})$  are negative. Although it is easy to con-  
 877 struct examples for which such bad behavior occurs, it requires special choices of  
 878 starting point  $x^0$ . Possibly the most obvious example is where  $f(x_1, x_2) = x_1^2 - x_2^2$   
 879 starting from  $x^0 = (1, 0)^\top$ , where  $d^k = -\nabla f(x^k)$  at each  $k$ . For this example, all  
 880 iterates have  $x_2^k = 0$  and, under appropriate conditions, converge to the saddle  
 881 point  $\bar{x} = 0$ . Any starting point with  $x_2^0 \neq 0$  cannot converge to 0, in fact, it is easy  
 882 to see that  $x_2^k$  diverges away from 0.

883 **4.6. Conditional Gradient Method** The conditional gradient approach, often  
 884 known as “Frank-Wolfe” after the authors who devised it [24], is a method for  
 885 convex nonlinear optimization over compact convex sets. This is the problem

$$886 \quad (4.6.1) \quad \min_{x \in \Omega} f(x),$$

887 (see earlier discussion around (3.2.2)), where  $\Omega$  is a compact convex set and  $f$   
 888 is a convex function whose gradient is Lipschitz continuously differentiable in a  
 889 neighborhood of  $\Omega$ , with Lipschitz constant  $L$ . We assume that  $\Omega$  has diameter  
 890  $D$ , that is,  $\|x - y\| \leq D$  for all  $x, y \in \Omega$ .

The conditional gradient method replaces the objective in (4.6.1) at each iteration by a linear Taylor-series approximation around the current iterate  $x^k$ , and minimizes this linear objective over the original constraint set  $\Omega$ . It then takes a step from  $x^k$  towards the minimizer of this linearized subproblem. The full method is as follows:

$$(4.6.2a) \quad v^k := \arg \min_{v \in \Omega} v^T \nabla f(x^k);$$

$$(4.6.2b) \quad x^{k+1} := x^k + \alpha_k (v^k - x^k), \quad \alpha_k := \frac{2}{k+2}.$$

891 The method has a sublinear convergence rate, as we show below, and indeed  
 892 requires many iterations in practice to obtain an accurate solution. Despite this  
 893 feature, it makes sense in many interesting applications, because the subproblems  
 894 (4.6.2a) can be solved very cheaply in some settings, and because highly accurate  
 895 solutions are not required in some applications.

896 We have the following result for sublinear convergence of the conditional gradient method.  
 897

898 **Theorem 4.6.3.** *Under the conditions above, where  $L$  is the Lipschitz constant for  $\nabla f$  on  
 899 an open neighborhood of  $\Omega$  and  $D$  is the diameter of  $\Omega$ , the conditional gradient method  
 900 (4.6.2) applied to (4.6.1) satisfies*

$$901 \quad (4.6.4) \quad f(x^k) - f(x^*) \leq \frac{2LD^2}{k+2}, \quad k = 1, 2, \dots,$$

902 where  $x^*$  is any solution of (4.6.1).

903 *Proof.* Setting  $x = x^k$  and  $y = x^{k+1} = x^k + \alpha_k (v^k - x^k)$  in (3.3.7), we have

$$904 \quad (4.6.5) \quad \begin{aligned} f(x^{k+1}) &\leq f(x^k) + \alpha_k \nabla f(x^k)^T (v^k - x^k) + \frac{1}{2} \alpha_k^2 L \|v^k - x^k\|^2 \\ &\leq f(x^k) + \alpha_k \nabla f(x^k)^T (v^k - x^k) + \frac{1}{2} \alpha_k^2 LD^2, \end{aligned}$$

905 where the second inequality comes from the definition of  $D$ . For the first-order  
 906 term, we have since  $v^k$  solves (4.6.2a) and  $x^*$  is feasible for (4.6.2a) that

$$907 \quad \nabla f(x^k)^T (v^k - x^k) \leq \nabla f(x^k)^T (x^* - x^k) \leq f(x^*) - f(x^k).$$

908 By substituting in (4.6.5) and subtracting  $f(x^*)$  from both sides, we obtain

$$909 \quad (4.6.6) \quad f(x^{k+1}) - f(x^*) \leq (1 - \alpha_k)[f(x^k) - f(x^*)] + \frac{1}{2} \alpha_k^2 LD^2.$$

910 We now apply an inductive argument. For  $k = 0$ , we have  $\alpha_0 = 1$  and

$$911 \quad f(x^1) - f(x^*) \leq \frac{1}{2}LD^2 < \frac{2}{3}LD^2,$$

so that (4.6.4) holds in this case. Supposing that (4.6.4) holds for some value of  $k$ , we aim to show that it holds for  $k + 1$  too. We have

$$\begin{aligned} & f(x^{k+1}) - f(x^*) \\ & \leq \left(1 - \frac{2}{k+2}\right) [f(x^k) - f(x^*)] + \frac{1}{2} \frac{4}{(k+2)^2} LD^2 \quad \text{from (4.6.6), (4.6.2b)} \\ & \leq LD^2 \left[ \frac{2k}{(k+2)^2} + \frac{2}{(k+2)^2} \right] \quad \text{from (4.6.4)} \\ & = 2LD^2 \frac{(k+1)}{(k+2)^2} \\ & = 2LD^2 \frac{k+1}{k+2} \frac{1}{k+2} \\ & \leq 2LD^2 \frac{k+2}{k+3} \frac{1}{k+2} = \frac{2LD^2}{k+3}, \end{aligned}$$

912 as required. □

## 913 5. Prox-Gradient Methods

914 We now describe an elementary but powerful approach for solving the regu-  
915 larized optimization problem

$$916 \quad (5.0.1) \quad \min_{x \in \mathbb{R}^n} \phi(x) := f(x) + \lambda\psi(x),$$

917 where  $f$  is a smooth convex function,  $\psi$  is a convex regularization function (known  
918 simply as the “regularizer”), and  $\lambda \geq 0$  is a regularization parameter. The tech-  
919 nique we describe here is a natural extension of the steepest-descent approach,  
920 in that it reduces to the steepest-descent method analyzed in Theorems 4.3.1 and  
921 4.4.1 applied to  $f$  when the regularization term is not present ( $\lambda = 0$ ). It is useful  
922 when the regularizer  $\psi$  has a simple structure that is easy to account for explicitly,  
923 as is true for many regularizers that arise in data analysis, such as the  $\ell_1$  function  
924 ( $\psi(x) = \|x\|_1$ ) of the indicator function for a simple set  $\Omega$  ( $\psi(x) = I_\Omega(x)$ ), such  
925 as a box  $\Omega = [l_1, u_1] \otimes [l_2, u_2] \otimes \dots \otimes [l_n, u_n]$ . For such regularizers, the proximal  
926 operators can be computed explicitly and efficiently.<sup>2</sup>

927 Each step of the algorithm is defined as follows:

$$928 \quad (5.0.2) \quad x^{k+1} := \text{prox}_{\alpha_k \lambda \psi}(x^k - \alpha_k \nabla f(x^k)),$$

929 for some steplength  $\alpha_k > 0$ , and the prox operator defined in (3.5.3). By substitut-  
930 ing into this definition, we can verify that  $x^{k+1}$  is the solution of an approximation  
931 to the objective  $\phi$  of (5.0.1), namely:

$$932 \quad (5.0.3) \quad x^{k+1} := \arg \min_z \nabla f(x^k)^\top (z - x^k) + \frac{1}{2\alpha_k} \|z - x^k\|^2 + \lambda\psi(z).$$

<sup>2</sup>For the analysis of this section I am indebted to class notes of L. Vandenberghe, from 2013-14.

933 One way to verify this equivalence is to note that the objective in (5.0.3) can be  
 934 written as

$$935 \quad \frac{1}{\alpha_k} \left\{ \frac{1}{2} \left\| z - (x^k - \alpha_k \nabla f(x^k)) \right\|^2 + \alpha_k \lambda \psi(x) \right\},$$

936 (modulo a term  $\alpha_k \|\nabla f(x^k)\|^2$  that does not involve  $z$ ). The subproblem objective  
 937 in (5.0.3) consists of a linear term  $\nabla f(x^k)^\top (z - x^k)$  (the first-order term in a Taylor-  
 938 series expansion), a proximality term  $\frac{1}{2\alpha_k} \|z - x^k\|^2$  that becomes more strict as  
 939  $\alpha_k \downarrow 0$ , and the regularization term  $\lambda \psi(x)$  in unaltered form. When  $\lambda = 0$ , we  
 940 have  $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$ , so the iteration (5.0.2) (or (5.0.3)) reduces to the  
 941 usual steepest-descent approach discussed in Section 4 in this case. It is useful  
 942 to continue thinking of  $\alpha_k$  as playing the role of a line-search parameter, though  
 943 here the line search is expressed implicitly through a proximal term.

944 We will demonstrate convergence of the method (5.0.2) at a sublinear rate, for  
 945 functions  $f$  whose gradients satisfy a Lipschitz continuity property with Lipschitz  
 946 constant  $L$  (see (3.3.6)), and for the constant steplength choice  $\alpha_k = 1/L$ . The proof  
 947 makes use of a “gradient map” defined by

$$948 \quad (5.0.4) \quad G_\alpha(x) := \frac{1}{\alpha} \left( x - \text{prox}_{\alpha\lambda\psi}(x - \alpha\nabla f(x)) \right).$$

949 By comparing with (5.0.2), we see that this map defines the step taken at iteration  
 950  $k$ :

$$951 \quad (5.0.5) \quad x^{k+1} = x^k - \alpha_k G_{\alpha_k}(x^k) \Leftrightarrow G_{\alpha_k}(x^k) = \frac{1}{\alpha_k} (x^k - x^{k+1}).$$

952 The following technical lemma reveals some useful properties of  $G_\alpha(x)$ .

953 **Lemma 5.0.6.** *Suppose that in problem (5.0.1),  $\psi$  is a closed convex function and that  $f$   
 954 is convex with Lipschitz continuous gradient on  $\mathbb{R}^n$ , with Lipschitz constant  $L$ . Then for  
 955 the definition (5.0.4) with  $\alpha > 0$ , the following claims are true.*

- 956 (a)  $G_\alpha(x) \in \nabla f(x) + \lambda \partial \psi(x - \alpha G_\alpha(x))$ .  
 957 (b) For any  $z$ , and any  $\alpha \in (0, 1/L]$ , we have that

$$958 \quad \phi(x - \alpha G_\alpha(x)) \leq \phi(z) + G_\alpha(x)^\top (x - z) - \frac{\alpha}{2} \|G_\alpha(x)\|^2.$$

959 *Proof.* For part (a), we use the optimality property (3.5.4) of the prox operator,  
 960 and make the following substitutions:  $x - \alpha \nabla f(x)$  for “ $x$ ”,  $\alpha \lambda$  for “ $\lambda$ ”, and  $\psi$  for  
 961 “ $h$ ” to obtain

$$962 \quad 0 \in \alpha \lambda \partial \psi(\text{prox}_{\alpha\lambda\psi}(x - \alpha \nabla f(x))) + (\text{prox}_{\alpha\lambda\psi}(x - \alpha \nabla f(x)) - (x - \alpha \nabla f(x))).$$

963 We make the substitution  $\text{prox}_{\alpha\lambda\psi}(x - \alpha \nabla f(x)) = x - \alpha G_\alpha(x)$ , using definition  
 964 (5.0.4), to obtain

$$965 \quad 0 \in \alpha \lambda \partial \psi(x - \alpha G_\alpha(x)) - \alpha (G_\alpha(x) - \nabla f(x)),$$

966 and the result follows when we divide by  $\alpha$ .

967 For (b), we start with the following consequence of Lipschitz continuity of  $\nabla f$ ,  
 968 from Lemma 3.3.10:

$$969 \quad f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2} \|y - x\|^2.$$

970 By setting  $y = x - \alpha G_\alpha(x)$ , for any  $\alpha \in (0, 1/L]$ , we have

$$\begin{aligned} 971 \quad (5.0.7) \quad f(x - \alpha G_\alpha(x)) &\leq f(x) - \alpha G_\alpha(x)^\top \nabla f(x) + \frac{L\alpha^2}{2} \|G_\alpha(x)\|^2 \\ &\leq f(x) - \alpha G_\alpha(x)^\top \nabla f(x) + \frac{\alpha}{2} \|G_\alpha(x)\|^2. \end{aligned}$$

972 (The second inequality uses  $\alpha \in (0, 1/L]$ .) We also have by convexity of  $f$  and  $\psi$   
973 that for any  $z$  and any  $v \in \partial\psi(x - \alpha G_\alpha(x))$  the following are true:

$$\begin{aligned} 974 \quad (5.0.8) \quad f(z) &\geq f(x) + \nabla f(x)^\top (z - x), \\ \psi(z) &\geq \psi(x - \alpha G_\alpha(x)) + v^\top (z - (x - \alpha G_\alpha(x))). \end{aligned}$$

975 From part (a) that  $v = (G_\alpha(x) - \nabla f(x))/\lambda \in \partial\psi(x - \alpha G_\alpha(x))$ . Making this choice  
976 of  $v$  in (5.0.8) and using (5.0.7) we have for any  $\alpha \in (0, 1/L]$  that

$$\begin{aligned} &\phi(x - \alpha G_\alpha(x)) \\ &= f(x - \alpha G_\alpha(x)) + \lambda\psi(x - \alpha G_\alpha(x)) \\ &\leq f(x) - \alpha G_\alpha(x)^\top \nabla f(x) + \frac{\alpha}{2} \|G_\alpha(x)\|^2 + \lambda\psi(x - \alpha G_\alpha(x)) \quad (\text{from (5.0.7)}) \\ &\leq f(z) + \nabla f(x)^\top (x - z) - \alpha G_\alpha(x)^\top \nabla f(x) + \frac{\alpha}{2} \|G_\alpha(x)\|^2 \\ &\quad + \lambda\psi(z) + (G_\alpha(x) - \nabla f(x))^\top (x - \alpha G_\alpha(x) - z) \quad (\text{from (5.0.8)}) \\ &= f(z) + \lambda\psi(z) + G_\alpha(x)^\top (x - z) - \frac{\alpha}{2} \|G_\alpha(x)\|^2, \end{aligned}$$

977 where the last equality follows from cancellation of several terms in the previous  
978 line. Thus (b) is proved.  $\square$

979 **Theorem 5.0.9.** *Suppose that in problem (5.0.1),  $\psi$  is a closed convex function and that  $f$   
980 is convex with Lipschitz continuous gradient on  $\mathbb{R}^n$ , with Lipschitz constant  $L$ . Suppose  
981 that (5.0.1) attains a minimizer  $x^*$  (not necessarily unique) with optimal objective value  
982  $\phi^*$ . Then if  $\alpha_k = 1/L$  for all  $k$  in (5.0.2), we have*

$$983 \quad \phi(x^k) - \phi^* \leq \frac{L\|x^0 - x^*\|^2}{2k}, \quad k = 1, 2, \dots$$

984 *Proof.* Since  $\alpha_k = 1/L$  satisfies the conditions of Lemma 5.0.6, we can use part (b)  
985 of this result to show that the sequence  $\{\phi(x^k)\}$  is decreasing and that the distance  
986 to the optimum  $x^*$  also decreases at each iteration. Setting  $x = z = x^k$  and  $\alpha = \alpha_k$   
987 in Lemma 5.0.6, and recalling (5.0.5), we have

$$988 \quad \phi(x^{k+1}) = \phi(x^k - \alpha_k G_{\alpha_k}(x^k)) \leq \phi(x^k) - \frac{\alpha_k}{2} \|G_{\alpha_k}(x^k)\|^2,$$

989 justifying the first claim. For the second claim, we have by setting  $x = x^k$ ,  $\alpha = \alpha_k$ ,  
990 and  $z = x^*$  in Lemma 5.0.6 that

$$\begin{aligned} 991 \quad (5.0.10) \quad 0 &\leq \phi(x^{k+1}) - \phi^* = \phi(x^k - \alpha_k G_{\alpha_k}(x^k)) - \phi^* \\ &\leq G_{\alpha_k}(x^k)^\top (x^k - x^*) - \frac{\alpha_k}{2} \|G_{\alpha_k}(x^k)\|^2 \\ &= \frac{1}{2\alpha_k} \left( \|x^k - x^*\|^2 - \|x^k - x^* - \alpha_k G_{\alpha_k}(x^k)\|^2 \right) \\ &= \frac{1}{2\alpha_k} \left( \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 \right), \end{aligned}$$

992 from which  $\|x^{k+1} - x^*\| \leq \|x^k - x^*\|$  follows.

993 By setting  $\alpha_k = 1/L$  in (5.0.10), and summing over  $k = 0, 1, 2, \dots, K-1$ , we  
 994 obtain from a telescoping sum on the right-hand side that

$$995 \quad \sum_{k=0}^{K-1} (\phi(x^{k+1}) - \phi^*) \leq \frac{L}{2} \left( \|x^0 - x^*\|^2 - \|x^K - x^*\|^2 \right) \leq \frac{L}{2} \|x^0 - x^*\|^2.$$

996 By monotonicity of  $\{\phi(x^k)\}$ , we have

$$997 \quad K(\phi(x^K) - \phi^*) \leq \sum_{k=0}^{K-1} (\phi(x^{k+1}) - \phi^*).$$

998 The result follows immediately by combining these last two expressions.  $\square$

## 999 6. Accelerating Gradient Methods

1000 We showed in Section 4 that the basic steepest descent method for solving  
 1001 (1.0.1) for smooth  $f$  converges sublinearly at a  $1/k$  rate when  $f$  is convex, and  
 1002 linearly at a rate of  $(1 - \gamma/L)$  when  $f$  is strongly convex, satisfying (3.3.13) for  
 1003 positive  $\gamma$  and  $L$ . We show in this section that by using the gradient information  
 1004 in a more clever way, faster convergence rates can be attained.

1005 The key idea is *momentum*. In iteration  $k$  of a momentum method, we tend to  
 1006 continue moving along the *previous* search direction at each iteration, making a  
 1007 small adjustment toward the negative gradient  $-\nabla f$  evaluated at  $x^k$  or a nearby  
 1008 point. (Steepest descent simply uses  $-\nabla f(x^k)$  as the search direction.) Although  
 1009 not obvious at first, there is some intuition behind the momentum idea. The step  
 1010 taken at the previous iterate  $x^{k-1}$  was based on negative gradient information  
 1011 at that iteration, along with the search direction from the iteration prior to that  
 1012 one, namely,  $x^{k-2}$ . By continuing this line of reasoning backwards, we see that  
 1013 the previous step is a linear combination of all the gradient information that we  
 1014 have encountered at all iterates so far, going back to the initial iterate  $x^0$ . If this  
 1015 information is aggregated properly, it can produce a richer overall picture of the  
 1016 function than the latest negative gradient alone, and thus has the potential to  
 1017 yield better convergence.

1018 Sure enough, several intricate methods that use the momentum idea have been  
 1019 proposed, and have been widely successful. These methods are often called *accel-*  
 1020 *erated gradient methods*. A major contributor in this area is Yuri Nesterov, dating to  
 1021 his seminal contribution in 1983 [33] and explicated further in his book [34] and  
 1022 other publications. Another key contribution is [3], which derived an accelerated  
 1023 method for the regularized case (1.0.2).

1024 **6.1. Heavy-Ball Method** Possibly the most elementary method of momentum  
 1025 type is the *heavy-ball* method of Polyak [37]; see also [38]. Each iteration of this  
 1026 method has the form

$$1027 \quad (6.1.1) \quad x^{k+1} = x^k - \alpha_k \nabla f(x^k) + \beta_k (x^k - x^{k-1}),$$

1028 where  $\alpha_k$  and  $\beta_k$  are positive scalars. That is, a momentum term  $\beta_k(x^k - x^{k-1})$   
 1029 is added to the usual steepest descent update. Although this method can be ap-  
 1030 plied to any smooth convex  $f$  (and even to nonconvex functions), the convergence  
 1031 analysis is most straightforward for the special case of strongly convex *quadratic*  
 1032 functions (see [38]). (This analysis also suggests appropriate values for the step  
 1033 lengths  $\alpha_k$  and  $\beta_k$ .) Consider the function

$$1034 \quad (6.1.2) \quad \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} x^\top A x - b^\top x,$$

1035 where the (constant) Hessian  $A$  has eigenvalues in the range  $[\gamma, L]$ , with  $0 < \gamma \leq L$ .  
 1036 For the following constant choices of steplength parameters:

$$1037 \quad \alpha_k = \alpha := \frac{4}{(\sqrt{L} + \sqrt{\gamma})^2}, \quad \beta_k = \beta := \frac{\sqrt{L} - \sqrt{\gamma}}{\sqrt{L} + \sqrt{\gamma}},$$

1038 it can be shown that  $\|x^k - x^*\| \leq C\beta^k$ , for some (possibly large) constant  $C$ . We  
 1039 can use (3.3.7) to translate this into a bound on the function error, as follows:

$$1040 \quad f(x^k) - f(x^*) \leq \frac{L}{2} \|x^k - x^*\|^2 \leq \frac{LC^2}{2} \beta^{2k},$$

1041 allowing a direct comparison with the rate (4.4.3) for the steepest descent method.  
 1042 If we suppose that  $L \gg \gamma$ , we have

$$1043 \quad \beta \approx 1 - 2\sqrt{\frac{\gamma}{L}},$$

1044 so that we achieve approximate convergence  $f(x^k) - f(x^*) \leq \epsilon$  (for small posi-  
 1045 tive  $\epsilon$ ) in  $O(\sqrt{L/\gamma} \log(1/\epsilon))$  iterations, compared with  $O((L/\gamma) \log(1/\epsilon))$  for  
 1046 steepest descent — a significant improvement.

1047 The heavy-ball method is fundamental, but several points should be noted.  
 1048 First, the analysis for convex quadratic  $f$  is based on linear algebra arguments,  
 1049 and does not generalize to general strongly convex nonlinear functions. Second,  
 1050 the method requires knowledge of  $\gamma$  and  $L$ , for the purposes of defining param-  
 1051 eters  $\alpha$  and  $\beta$ . Third, it is not a descent method; we usually have  $f(x^{k+1}) > f(x^k)$   
 1052 for many  $k$ . These properties are not specific to the heavy-ball method — some  
 1053 of them are shared by other methods that use momentum.

1054 **6.2. Conjugate Gradient** The conjugate gradient method for solving linear sys-  
 1055 tems  $Ax = b$  (or, equivalently, minimizing the convex quadratic (6.1.2)) where  $A$   
 1056 is symmetric positive definite, is one of the most important algorithms in compu-  
 1057 tational science. Though invented earlier than the other algorithms discussed in  
 1058 this section (see [27]) and motivated in a different way, conjugate gradient clearly  
 1059 makes use of momentum. Its steps have the form

$$1060 \quad (6.2.1) \quad x^{k+1} = x^k + \alpha_k p^k, \quad \text{where } p^k = -\nabla f(x^k) + \xi_k p^{k-1},$$

1061 for some choices of  $\alpha_k$  and  $\xi_k$ , which is identical to (6.1.1) when we define  $\beta_k$  ap-  
 1062 propriately. For convex, strongly quadratic problems (6.1.2), conjugate gradient  
 1063 has excellent properties. It does not require prior knowledge of the range  $[\gamma, L]$  of

1064 the eigenvalue spectrum of  $A$ , choosing the steplengths  $\alpha_k$  and  $\xi_k$  in an adaptive  
 1065 fashion. (In fact,  $\alpha_k$  is chosen to be the exact minimizer along the search direction  
 1066  $p_k$ .) The main arithmetic operation per iteration is one matrix-vector multiplica-  
 1067 tion involving  $A$ , the same cost as a gradient evaluation for  $f$  in (6.1.2). Most  
 1068 importantly, there is a rich convergence theory, that characterizes convergence in  
 1069 terms of the properties of the full spectrum of  $A$  (not just its extreme elements),  
 1070 showing in particular that good approximate solutions can be obtained quickly  
 1071 if the eigenvalues are clustered. Convergence to an exact solution of (6.1.2) in at  
 1072 most  $n$  iterations is guaranteed (provided, naturally, that the arithmetic is carried  
 1073 out exactly).

1074 There has been much work over the years on extending the conjugate gradi-  
 1075 ent method to general smooth functions  $f$ . Few of the theoretical properties for  
 1076 the quadratic case carry over to the nonlinear setting, though several results are  
 1077 known; see [36, Chapter 5], for example. Such “nonlinear” conjugate gradient  
 1078 methods vary in the accuracy with which they perform the line search for  $\alpha_k$  in  
 1079 (6.2.1) and — more fundamentally — in the choice of  $\xi_k$ . The latter is done in  
 1080 a way that ensures that each search direction  $p^k$  is a descent direction. In some  
 1081 methods,  $\xi_k$  is set to zero on some iterations, which causes the method to take  
 1082 a steepest descent step, effectively “restarting” the conjugate gradient method at  
 1083 the latest iterate.

1084 Despite these qualifications, nonlinear conjugate gradient is quite commonly  
 1085 used in practice, because of its minimal storage requirements and the fact that  
 1086 it requires only one gradient evaluation per iteration. Its popularity has been  
 1087 eclipsed in recent years by the limited-memory quasi-Newton method L-BFGS  
 1088 [30], [36, Section 7.2], which requires more storage (though still  $O(n)$ ) and is  
 1089 similarly economical and easy to implement.

1090 **6.3. Nesterov’s Accelerated Gradient: Weakly Convex Case** We now describe  
 1091 Nesterov’s method for (1.0.1) and prove its convergence — sublinear at a  $1/k^2$   
 1092 rate — for the case of  $f$  convex with Lipschitz continuous gradients satisfying  
 1093 (3.3.6). Each iteration of this method has the form

$$1094 \quad (6.3.1) \quad x^{k+1} = x^k - \alpha_k \nabla f \left( x^k + \beta_k (x^k - x^{k-1}) \right) + \beta_k (x^k - x^{k-1}),$$

for choices of the parameters  $\alpha_k$  and  $\beta_k$  to be defined. Note immediately the  
 similarity to the heavy-ball formula (6.1.1). The only difference is that the extrap-  
 olation step  $x^k \rightarrow x^k + \beta_k (x^k - x^{k-1})$  is taken before evaluation of the gradient  
 $\nabla f$  in (6.3.1), whereas in (6.1.1) the gradient is simply evaluated at  $x^k$ . It is con-  
 venient for purposes of analysis (and implementation) to introduce an auxiliary  
 sequence  $\{y^k\}$ , fix  $\alpha_k \equiv 1/L$ , and rewrite the update (6.3.1) as follows:

$$(6.3.2a) \quad x^{k+1} = y^k - \frac{1}{L} \nabla f(y^k),$$

$$(6.3.2b) \quad y^{k+1} = x^{k+1} + \beta_{k+1} (x^{k+1} - x^k), \quad k = 0, 1, 2, \dots,$$

1095 where we initialize at an arbitrary  $y^0$  and set  $x^0 = y^0$ . We define  $\beta_k$  with reference  
1096 to another scalar sequence  $\lambda_k$  in the following manner:

$$1097 \quad (6.3.3) \quad \lambda_0 = 0, \quad \lambda_{k+1} = \frac{1}{2} \left( 1 + \sqrt{1 + 4\lambda_k^2} \right), \quad \beta_k = \frac{\lambda_k - 1}{\lambda_{k+1}}.$$

1098 Since  $\lambda_k \geq 1$  for  $k = 1, 2, \dots$ , we have  $\beta_{k+1} \geq 0$  for  $k = 0, 1, 2, \dots$ . It also follows  
1099 from the definition of  $\lambda_{k+1}$  that

$$1100 \quad (6.3.4) \quad \lambda_{k+1}^2 - \lambda_{k+1} = \lambda_k^2.$$

1101 We have the following result for convergence of Nesterov's scheme on general  
1102 convex functions. We prove it using an argument from [3], as reformulated in  
1103 [7, Section 3.7]. The analysis is famously technical, and intuition is hard to come  
1104 by. Some recent progress has been made in deriving algorithms similar to (6.3.2)  
1105 that have a plausible geometric or algebraic motivation; see [8, 21].

1106 **Theorem 6.3.5.** *Suppose that  $f$  in (1.0.1) is convex, with  $\nabla f$  Lipschitz continuously*  
1107 *differentiable with constant  $L$  (as in (3.3.6)) and that the minimum of  $f$  is attained at  $x^*$ ,*  
1108 *with  $f^* := f(x^*)$ . Then the method defined by (6.3.2), (6.3.3) with  $x^0 = y^0$  yields an*  
1109 *iteration sequence  $\{x^k\}$  with the following property:*

$$1110 \quad f(x^T) - f^* \leq \frac{2L\|x^0 - x^*\|^2}{(T+1)^2}, \quad T = 1, 2, \dots$$

1111 *Proof.* From convexity of  $f$  and (3.3.7), we have for any  $x$  and  $y$  that

$$\begin{aligned} & f(y - \nabla f(y)/L) - f(x) \\ & \leq f(y - \nabla f(y)/L) - f(y) + \nabla f(y)^T(y - x) \\ 1112 \quad (6.3.6) \quad & \leq \nabla f(y)^T(y - \nabla f(y)/L - y) + \frac{L}{2}\|y - \nabla f(y)/L - y\|^2 + \nabla f(y)^T(y - x) \\ & = -\frac{1}{2L}\|\nabla f(y)\|^2 + \nabla f(y)^T(y - x). \end{aligned}$$

1113 Setting  $y = y^k$  and  $x = x^k$  in this bound, we obtain

$$\begin{aligned} & f(x^{k+1}) - f(x^k) = f(y^k - \nabla f(y^k)/L) - f(x^k) \\ 1114 \quad (6.3.7) \quad & \leq -\frac{1}{2L}\|\nabla f(y^k)\|^2 + \nabla f(y^k)^T(y^k - x^k) \\ & = -\frac{L}{2}\|x^{k+1} - y^k\|^2 - L(x^{k+1} - y^k)^T(y^k - x^k). \end{aligned}$$

1115 We now set  $y = y^k$  and  $x = x^*$  in (6.3.6), and use (6.3.2a) to obtain

$$1116 \quad (6.3.8) \quad f(x^{k+1}) - f(x^*) \leq -\frac{L}{2}\|x^{k+1} - y^k\|^2 - L(x^{k+1} - y^k)^T(y^k - x^*).$$

Introducing notation  $\delta_k := f(x^k) - f(x^*)$ , we multiply (6.3.7) by  $\lambda_{k+1} - 1$  and add it to (6.3.8) to obtain

$$\begin{aligned} & (\lambda_{k+1} - 1)(\delta_{k+1} - \delta_k) + \delta_{k+1} \\ & \leq -\frac{L}{2}\lambda_{k+1}\|x^{k+1} - y^k\|^2 - L(x^{k+1} - y^k)^T(\lambda_{k+1}y^k - (\lambda_{k+1} - 1)x^k - x^*). \end{aligned}$$

We multiply this bound by  $\lambda_{k+1}$ , and use (6.3.4) to obtain

$$(6.3.9) \quad \begin{aligned} & \lambda_{k+1}^2 \delta_{k+1} - \lambda_k^2 \delta_k \\ & \leq -\frac{L}{2} \left[ \|\lambda_{k+1}(x^{k+1} - y^k)\|^2 + 2\lambda_{k+1}(x^{k+1} - y^k)^\top (\lambda_{k+1}y^k - (\lambda_{k+1} - 1)x^k - x^*) \right] \\ & = -\frac{L}{2} \left[ \|\lambda_{k+1}x^{k+1} - (\lambda_{k+1} - 1)x^k - x^*\|^2 - \|\lambda_{k+1}y^k - (\lambda_{k+1} - 1)x^k - x^*\|^2 \right], \end{aligned}$$

where in the final equality we used the identity  $\|a\|^2 + 2a^\top b = \|a + b\|^2 - \|b\|^2$ . By multiplying (6.3.2b) by  $\lambda_{k+2}$ , and using  $\lambda_{k+2}\beta_{k+1} = \lambda_{k+1} - 1$  from (6.3.3), we have

$$\begin{aligned} \lambda_{k+2}y^{k+1} &= \lambda_{k+2}x^{k+1} + \lambda_{k+2}\beta_{k+1}(x^{k+1} - x^k) \\ &= \lambda_{k+2}x^{k+1} + (\lambda_{k+1} - 1)(x^{k+1} - x^k). \end{aligned}$$

1117 By rearranging this equality, we have

$$1118 \quad \lambda_{k+1}x^{k+1} - (\lambda_{k+1} - 1)x^k = \lambda_{k+2}y^{k+1} - (\lambda_{k+2} - 1)x^{k+1}.$$

1119 By substituting into the first term on the right-hand side of (6.3.9), and using the  
1120 definition

$$1121 \quad (6.3.10) \quad u^k := \lambda_{k+1}y^k - (\lambda_{k+1} - 1)x^k - x^*,$$

1122 we obtain

$$1123 \quad \lambda_{k+1}^2 \delta_{k+1} - \lambda_k^2 \delta_k \leq -\frac{L}{2} (\|u^{k+1}\|^2 - \|u^k\|^2).$$

1124 By summing both sides of this inequality over  $k = 0, 1, \dots, T-1$ , and using  $\lambda_0 = 0$ ,  
1125 we obtain

$$1126 \quad \lambda_T^2 \delta_T \leq \frac{L}{2} (\|u^0\|^2 - \|u^T\|^2) \leq \frac{L}{2} \|x^0 - x^*\|^2,$$

1127 so that

$$1128 \quad (6.3.11) \quad \delta_T = f(x^T) - f(x^*) \leq \frac{L\|x^0 - x^*\|^2}{2\lambda_T^2}.$$

1129 A simple induction confirms that  $\lambda_k \geq (k+1)/2$  for  $k = 1, 2, \dots$ , and the claim of  
1130 the theorem follows by substituting this bound into (6.3.11).  $\square$

1131 **6.4. Nesterov's Accelerated Gradient: Strongly Convex Case** We turn now to  
1132 Nesterov's approach for smooth strongly convex functions, which satisfy (3.2.5)  
1133 with  $\gamma > 0$ . Again, we follow the proof in [7, Section 3.7], which is based on  
1134 the analysis in [34]. The method uses the same update formula (6.3.2) as in the  
1135 weakly convex case, and the same initialization, but with a different choice of  
1136  $\beta_{k+1}$ , namely:

$$1137 \quad (6.4.1) \quad \beta_{k+1} \equiv \frac{\sqrt{L} - \sqrt{\gamma}}{\sqrt{L} + \sqrt{\gamma}} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}.$$

1138 The condition measure  $\kappa$  is defined in (3.3.12). We prove the following conver-  
1139 gence result.

1140 **Theorem 6.4.2.** *Suppose that  $f$  is such that  $\nabla f$  is Lipschitz continuously differentiable  
1141 with constant  $L$ , and that it is strongly convex with modulus of convexity  $\gamma$  and unique*

1142 *minimizer*  $x^*$ . Then the method (6.3.2), (6.4.1) with starting point  $x^0 = y^0$  satisfies

$$1143 \quad f(x^T) - f(x^*) \leq \frac{L+\gamma}{2} \|x^0 - x^*\|^2 \left(1 - \frac{1}{\sqrt{\kappa}}\right)^T, \quad T = 1, 2, \dots$$

*Proof.* The proof makes use of a family of strongly convex functions  $\Phi_k(z)$  defined inductively as follows:

$$(6.4.3a) \quad \Phi_0(z) = f(y^0) + \frac{\gamma}{2} \|z - y^0\|^2,$$

$$(6.4.3b) \quad \Phi_{k+1}(z) = (1 - 1/\sqrt{\kappa})\Phi_k(z) + \frac{1}{\sqrt{\kappa}} \left( f(y^k) + \nabla f(y^k)^\top (z - y^k) + \frac{\gamma}{2} \|z - y^k\|^2 \right).$$

1144 Each  $\Phi_k(\cdot)$  is a quadratic, and an inductive argument shows that  $\nabla^2 \Phi_k(z) = \gamma I$   
1145 for all  $k$  and all  $z$ . Thus, each  $\Phi_k$  has the form

$$1146 \quad (6.4.4) \quad \Phi_k(z) = \Phi_k^* + \frac{\gamma}{2} \|z - v^k\|^2, \quad k = 0, 1, 2, \dots,$$

1147 where  $v^k$  is the minimizer of  $\Phi_k(\cdot)$  and  $\Phi_k^*$  is its optimal value. (From (6.4.3a),  
1148 we have  $v^0 = y^0$ .) We note too that  $\Phi_k$  becomes a tighter *overapproximation* to  $f$   
1149 as  $k \rightarrow \infty$ . To show this, we use (3.3.9) to replace the final term in parentheses in  
1150 (6.4.3b) by  $f(z)$ , then subtract  $f(z)$  from both sides of (6.4.3b) to obtain

$$1151 \quad (6.4.5) \quad \Phi_{k+1}(z) - f(z) \leq (1 - 1/\sqrt{\kappa})(\Phi_k(z) - f(z)).$$

1152 In the remainder of the proof, we show that the following bound holds:

$$1153 \quad (6.4.6) \quad f(x^k) \leq \min_z \Phi_k(z) = \Phi_k^*, \quad k = 0, 1, 2, \dots$$

1154 The upper bound in Lemma 3.3.10 for  $x = x^*$  gives  $f(z) - f(x^*) \leq (L/2)\|z - x^*\|^2$ .

1155 By combining this bound with (6.4.5) and (6.4.6), we have

$$\begin{aligned} f(x^k) - f(x^*) &\leq \Phi_k^* - f(x^*) \\ &\leq \Phi_k(x^*) - f(x^*) \\ &\leq (1 - 1/\sqrt{\kappa})^k (\Phi_0(x^*) - f(x^*)) \\ 1156 \quad (6.4.7) \quad &\leq (1 - 1/\sqrt{\kappa})^k [( \Phi_0(x^*) - f(x^0) ) + (f(x^0) - f(x^*))] \\ &\leq (1 - 1/\sqrt{\kappa})^k \frac{\gamma + L}{2} \|x^0 - x^*\|^2. \end{aligned}$$

The proof is completed by establishing (6.4.6), by induction on  $k$ . Since  $x^0 = y^0$ , it holds by definition at  $k = 0$ . By using step formula (6.3.2a), the convexity property (3.3.8) (with  $x = y^k$ ), and the inductive hypothesis, we have

$$\begin{aligned} (6.4.8) \quad f(x^{k+1}) &\leq f(y^k) - \frac{1}{2L} \|\nabla f(y^k)\|^2 \\ &= (1 - 1/\sqrt{\kappa})f(x^k) + (1 - 1/\sqrt{\kappa})(f(y^k) - f(x^k)) + f(y^k)/\sqrt{\kappa} - \frac{1}{2L} \|\nabla f(y^k)\|^2 \\ &\leq (1 - 1/\sqrt{\kappa})\Phi_k^* + (1 - 1/\sqrt{\kappa})\nabla f(y^k)^\top (y^k - x^k) + f(y^k)/\sqrt{\kappa} - \frac{1}{2L} \|\nabla f(y^k)\|^2. \end{aligned}$$

1157 Thus the claim is established (and the theorem is proved) if we can show that the  
1158 right-hand side in (6.4.8) is bounded above by  $\Phi_{k+1}^*$ .

1159 Recalling the observation (6.4.4), we have by taking derivatives of both sides of  
1160 (6.4.3b) with respect to  $z$  that

$$1161 \quad (6.4.9) \quad \nabla \Phi_{k+1}(z) = \gamma(1 - 1/\sqrt{\kappa})(z - v^k) + \nabla f(y^k)/\sqrt{\kappa} + \gamma(z - y^k)/\sqrt{\kappa}.$$

1162 Since  $v^{k+1}$  is the minimizer of  $\Phi_{k+1}$  we can set  $\nabla \Phi_{k+1}(v^{k+1}) = 0$  in (6.4.9) to  
1163 obtain

$$1164 \quad (6.4.10) \quad v^{k+1} = (1 - 1/\sqrt{\kappa})v^k + y^k/\sqrt{\kappa} - \nabla f(y^k)/(\gamma\sqrt{\kappa}).$$

1165 By subtracting  $y^k$  from both sides of this expression, and taking  $\|\cdot\|^2$  of both  
1166 sides, we obtain

$$1167 \quad (6.4.11) \quad \begin{aligned} \|v^{k+1} - y^k\|^2 &= (1 - 1/\sqrt{\kappa})^2 \|y^k - v^k\|^2 + \|\nabla f(y^k)\|^2/(\gamma^2\kappa) \\ &\quad - 2(1 - 1/\sqrt{\kappa})/(\gamma\sqrt{\kappa}) \nabla f(y^k)^\top (v^k - y^k). \end{aligned}$$

1168 By evaluating  $\Phi_{k+1}$  at  $z = y^k$ , using both (6.4.4) and (6.4.3b), we obtain

$$1169 \quad (6.4.12) \quad \begin{aligned} \Phi_{k+1}^* + \frac{\gamma}{2} \|y^k - v^{k+1}\|^2 \\ &= (1 - 1/\sqrt{\kappa})\Phi_k(y^k) + f(y^k)/\sqrt{\kappa} \\ &= (1 - 1/\sqrt{\kappa})\Phi_k^* + \frac{\gamma}{2}(1 - 1/\sqrt{\kappa})\|y^k - v^k\|^2 + f(y^k)/\sqrt{\kappa}. \end{aligned}$$

1170 By substituting (6.4.11) into (6.4.12), we obtain

$$1171 \quad (6.4.13) \quad \begin{aligned} \Phi_{k+1}^* &= (1 - 1/\sqrt{\kappa})\Phi_k^* + f(y^k)/\sqrt{\kappa} + \gamma(1 - 1/\sqrt{\kappa})/(2\sqrt{\kappa})\|y^k - v^k\|^2 \\ &\quad - \frac{1}{2L}\|\nabla f(y^k)\|^2 + (1 - 1/\sqrt{\kappa})\nabla f(y^k)^\top (v^k - y^k)/\sqrt{\kappa} \\ &\geq (1 - 1/\sqrt{\kappa})\Phi_k^* + f(y^k)/\sqrt{\kappa} \\ &\quad - \frac{1}{2L}\|\nabla f(y^k)\|^2 + (1 - 1/\sqrt{\kappa})\nabla f(y^k)^\top (v^k - y^k)/\sqrt{\kappa}, \end{aligned}$$

1172 where we simply dropped a nonnegative term from the right-hand side to obtain  
1173 the inequality. The final step is to show that

$$1174 \quad (6.4.14) \quad v^k - y^k = \sqrt{\kappa}(y^k - x^k),$$

1175 which we do by induction. Note that  $v^0 = x^0 = y^0$ , so the claim holds for  $k = 0$ .

1176 We have

$$1177 \quad (6.4.15) \quad \begin{aligned} v^{k+1} - y^{k+1} &= (1 - 1/\sqrt{\kappa})v^k + y^k/\sqrt{\kappa} - \nabla f(y^k)/(\gamma\sqrt{\kappa}) - y^{k+1} \\ &= \sqrt{\kappa}y^k - (\sqrt{\kappa} - 1)x^k - \sqrt{\kappa}\nabla f(y^k)/L - y^{k+1} \\ &= \sqrt{\kappa}x^{k+1} - (\sqrt{\kappa} - 1)x^k - y^{k+1} \\ &= \sqrt{\kappa}(y^{k+1} - x^{k+1}), \end{aligned}$$

1178 where the first equality is from (6.4.10), the second equality is from the inductive  
1179 hypothesis, the third equality is from the iteration formula (6.3.2a), and the final  
1180 equality is from the iteration formula (6.3.2b) with the definition of  $\beta_{k+1}$  from  
1181 (6.4.1). We have thus proved (6.4.14), and by substituting this equality into (6.4.13),

1182 we obtain that  $\Phi_{k+1}^*$  is an upper bound on the right-hand side of (6.4.8). This  
 1183 establishes (6.4.6) and thus completes the proof of the theorem.  $\square$

1184 **6.5. Lower Bounds on Rates** The term “optimal” in Nesterov’s optimal method  
 1185 is used because the convergence rate achieved by the method is the best possible  
 1186 (possibly up to a constant), among algorithms that make use of gradient informa-  
 1187 tion at the iterates  $x^k$ . This claim can be proved by means of a carefully designed  
 1188 function, for which *no* method that makes use of all gradients observed up to and  
 1189 including iteration  $k$  (namely,  $\nabla f(x^i)$ ,  $i = 0, 1, 2, \dots, k$ ) can produce a sequence  
 1190  $\{x^k\}$  that achieves a rate better than that of Theorem 6.3.5. The function proposed  
 1191 in [32] is a convex quadratic  $f(x) = (1/2)x^T Ax - e_1^T x$ , where

$$1192 \quad A = \begin{bmatrix} 2 & -1 & 0 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & \dots & & 0 & -1 & 2 & -1 \\ 0 & \dots & & & 0 & -1 & 2 \end{bmatrix}, \quad e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

1193 The solution  $x^*$  satisfies  $Ax^* = e_1$ ; its components are  $x_i^* = 1 - i/(n+1)$ , for  
 1194  $i = 1, 2, \dots, n$ . If we use  $x^0 = 0$  as the starting point, and construct the iterate  
 1195  $x^{k+1}$  as

$$1196 \quad x^{k+1} = x^k + \sum_{j=0}^k \xi_j \nabla f(x^j),$$

1197 for some coefficients  $\xi_j$ ,  $j = 0, 1, \dots, k$ , an elementary inductive argument shows  
 1198 that each iterate  $x^k$  can have nonzero entries only in its first  $k$  components. It  
 1199 follows that for any such algorithm, we have

$$1200 \quad (6.5.1) \quad \|x^k - x^*\|^2 \geq \sum_{j=k+1}^n (x_j^*)^2 = \sum_{j=k+1}^n \left(1 - \frac{j}{n+1}\right)^2.$$

1201 A little arithmetic shows that

$$1202 \quad (6.5.2) \quad \|x^k - x^*\|^2 \geq \frac{1}{8} \|x^0 - x^*\|^2, \quad k = 1, 2, \dots, \frac{n}{2} - 1,$$

1203 It can be shown further that

$$1204 \quad (6.5.3) \quad f(x^k) - f^* \geq \frac{3L}{32(k+1)^2} \|x^0 - x^*\|^2, \quad k = 1, 2, \dots, \frac{n}{2} - 1,$$

1205 where  $L = \|A\|_2$ . This lower bound on  $f(x^k) - f^*$  is within a constant factor of  
 1206 the upper bound of Theorem 6.3.5.

1207 The restriction  $k \leq n/2$  in the argument above is not fully satisfying. A more  
 1208 compelling example would show that the lower bound (6.5.3) holds for *all*  $k$ , but  
 1209 an example of this type is not currently known.

## 7. Newton Methods

1210

1211 So far, we have dealt with methods that use first-order (gradient or subgra-  
 1212 dient) information about the objective function. We have shown that such algo-  
 1213 rithms can yield sequences of iterates that converge at linear or sublinear rates.  
 1214 We turn our attention in this chapter to methods that exploit second-derivative  
 1215 (Hessian) information. The canonical method here is Newton's method, named  
 1216 after Isaac Newton, who proposed a version of the method for polynomial equa-  
 1217 tions in around 1670.

1218 For many functions, including many that arise in data analysis, second-order  
 1219 information is not difficult to compute, in the sense that the functions that we  
 1220 deal with are simple (usually compositions of elementary functions). In compar-  
 1221 ing with first-order methods, there is a tradeoff. Second-order methods typically  
 1222 have local superlinear or quadratic convergence rates: Once the iterates reach a  
 1223 neighborhood of a solution at which second-order sufficient conditions are sat-  
 1224 isfied, convergence is rapid. Moreover, their global convergence properties are  
 1225 attractive. With appropriate enhancements, they can provably avoid convergence  
 1226 to saddle points. But the costs of calculating and handling the second-order infor-  
 1227 mation and of computing the step is higher. Whether this tradeoff makes them  
 1228 appealing depends on the specifics of the application and on whether the second-  
 1229 derivative computations are able to take advantage of structure in the objective  
 1230 function.

1231 We start by sketching the basic Newton's method for the unconstrained smooth  
 1232 optimization problem  $\min f(x)$ , and prove local convergence to a minimizer  $x^*$   
 1233 that satisfies second-order sufficient conditions. Subsection 7.2 discusses perfor-  
 1234 mance of Newton's method on convex functions, where the use of Newton search  
 1235 directions in the line search framework (4.0.1) can yield global convergence. Mod-  
 1236 ifications of Newton's method for nonconvex functions are discussed in Subsec-  
 1237 tion 7.3. Subsection 7.4 discusses algorithms for smooth nonconvex functions  
 1238 that use gradient and Hessian information but guarantee convergence to points  
 1239 that approximately satisfy second-order necessary conditions. Some variants of  
 1240 these methods are related closely to the trust-region methods discussed in Sub-  
 1241 section 7.3, but the motivation and mechanics are somewhat different.

1242 **7.1. Basic Newton's Method** Consider the problem

$$1243 \quad (7.1.1) \quad \min f(x),$$

1244 where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a Lipschitz twice continuously differentiable function, where  
 1245 the Hessian has Lipschitz constant  $M$ , that is,

$$1246 \quad (7.1.2) \quad \|\nabla^2 f(x') - \nabla^2 f(x'')\| \leq M \|x' - x''\|,$$

1247 where  $\|\cdot\|$  denotes the Euclidean vector norm and its induced matrix norm. New-  
 1248 ton's method generates a sequence of iterates  $\{x^k\}_{k=0,1,2,\dots}$ .

1249 A second-order Taylor series approximation to  $f$  around the current iterate  $x^k$   
 1250 is

$$1251 \quad (7.1.3) \quad f(x^k + p) \approx f(x^k) + \nabla f(x^k)^\top p + \frac{1}{2} p^\top \nabla^2 f(x^k) p.$$

1252 When  $\nabla^2 f(x^k)$  is positive definite, the minimizer  $p^k$  of the right-hand side is  
 1253 unique; it is

$$1254 \quad (7.1.4) \quad p^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

1255 This is the Newton step. In its most basic form, then, Newton's method is defined  
 1256 by the following iteration:

$$1257 \quad (7.1.5) \quad x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

1258 We have the following local convergence result in the neighborhood of a point  $x^*$   
 1259 satisfying second-order sufficient conditions.

1260 **Theorem 7.1.6.** *Consider the problem (7.1.1) with  $f$  twice Lipschitz continuously differ-*  
 1261 *entiable with Lipschitz constant  $M$  defined in (7.1.2). Suppose that the second-order suf-*  
 1262 *ficient conditions are satisfied for the problem (7.1.1) at the point  $x^*$ , that is,  $\nabla f(x^*) = 0$*   
 1263 *and  $\nabla^2 f(x^*) \succeq \gamma I$  for some  $\gamma > 0$ . Then if  $\|x^0 - x^*\| \leq \frac{\gamma}{2M}$ , the sequence defined by*  
 1264 *(7.1.5) converges to  $x^*$  at a quadratic rate, with*

$$1265 \quad (7.1.7) \quad \|x^{k+1} - x^*\| \leq \frac{M}{\gamma} \|x^k - x^*\|^2, \quad k = 0, 1, 2, \dots$$

*Proof.* From (7.1.4) and (7.1.5), and using  $\nabla f(x^*) = 0$ , we have

$$\begin{aligned} x^{k+1} - x^* &= x^k - x^* - \nabla^2 f(x^k)^{-1} \nabla f(x^k) \\ &= \nabla^2 f(x^k)^{-1} [\nabla^2 f(x^k)(x^k - x^*) - (\nabla f(x^k) - \nabla f(x^*))]. \end{aligned}$$

1266 so that

$$1267 \quad (7.1.8) \quad \|x^{k+1} - x^*\| \leq \|\nabla^2 f(x^k)^{-1}\| \|\nabla^2 f(x^k)(x^k - x^*) - (\nabla f(x^k) - \nabla f(x^*))\|.$$

1268 By using Taylor's theorem (see (3.3.4) with  $x = x^k$  and  $p = x^* - x^k$ ), we have

$$\nabla f(x^k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^k + t(x^* - x^k))(x^k - x^*) dt.$$

1269 By using this result along with the Lipschitz condition (7.1.2), we have

$$\begin{aligned} &\|\nabla^2 f(x^k)(x^k - x^*) - (\nabla f(x^k) - \nabla f(x^*))\| \\ &= \left\| \int_0^1 [\nabla^2 f(x^k) - \nabla^2 f(x^k + t(x^* - x^k))](x^k - x^*) dt \right\| \\ 1270 \quad (7.1.9) \quad &\leq \int_0^1 \|\nabla^2 f(x^k) - \nabla^2 f(x^k + t(x^* - x^k))\| \|x^k - x^*\| dt \\ &\leq \left( \int_0^1 M t dt \right) \|x^k - x^*\|^2 = \frac{1}{2} M \|x^k - x^*\|^2. \end{aligned}$$

1271 From the Weilandt-Hoffman inequality[28] and (7.1.2), we have that

$$1272 \quad |\lambda_{\min}(\nabla^2 f(x^k)) - \lambda_{\min}(\nabla^2 f(x^*))| \leq \|\nabla^2 f(x^k) - \nabla^2 f(x^*)\| \leq M \|x^k - x^*\|,$$

1273 where  $\lambda_{\min}(\cdot)$  denotes the smallest eigenvalue of a symmetric matrix. Thus for

$$1274 \quad (7.1.10) \quad \|x^k - x^*\| \leq \frac{\gamma}{2M},$$

1275 we have

$$1276 \quad \lambda_{\min}(\nabla^2 f(x^k)) \geq \lambda_{\min}(\nabla^2 f(x^*)) - M\|x^k - x^*\| \geq \gamma - M\frac{\gamma}{2M} \geq \frac{\gamma}{2},$$

1277 so that  $\|\nabla^2 f(x^k)^{-1}\| \leq 2/\gamma$ . By substituting this result together with (7.1.9) into  
1278 (7.1.8), we obtain

$$1279 \quad \|x^{k+1} - x^*\| \leq \frac{2}{\gamma} \frac{M}{2} \|x^k - x^*\|^2 = \frac{M}{\gamma} \|x^k - x^*\|^2,$$

1280 verifying the local quadratic convergence rate. By applying (7.1.10) again, we  
1281 have

$$1282 \quad \|x^{k+1} - x^*\| \leq \left( \frac{M}{\gamma} \|x^k - x^*\| \right) \|x^k - x^*\| \leq \frac{1}{2} \|x^k - x^*\|,$$

1283 so, by arguing inductively, we see that the sequence converges to  $x^*$  provided  
1284 that  $x^0$  satisfies (7.1.10), as claimed.  $\square$

1285 Of course, we do not need to explicitly identify a starting point  $x^0$  in the stated  
1286 region of convergence. Any sequence that approaches to  $x^*$  will eventually enter  
1287 this region, and thereafter the quadratic convergence guarantees apply.

1288 We have established that Newton's method converges rapidly once the iterates  
1289 enter the neighborhood of a point  $x^*$  satisfying second-order sufficient optimality  
1290 conditions. But what happens when we start far from such a point?

1291 **7.2. Newton's Method for Convex Functions** When the function  $f$  is convex as  
1292 well as smooth, we can devise variants of Newton's method for which global  
1293 convergence and complexity results (in particular, results based on those of Sec-  
1294 tion 4.5) can be proved in addition to local quadratic convergence.

1295 When  $f$  is strongly convex with modulus  $\gamma$  and satisfies Lipschitz continuity  
1296 of the gradient (3.3.6), the Hessian  $\nabla^2 f(x^k)$  is positive definite for all  $k$ , with  
1297 all eigenvalues in the interval  $[\gamma, L]$ . Thus, the Newton direction (7.1.4) is well  
1298 defined at all iterates  $x^k$ , and is a descent direction satisfying the condition (4.5.1)  
1299 with  $\eta = \gamma/L$ . To verify this claim, note first

$$1300 \quad \|p^k\| \leq \|\nabla^2 f(x^k)^{-1}\| \|\nabla f(x^k)\| \leq \frac{1}{\gamma} \|\nabla f(x^k)\|.$$

Then

$$\begin{aligned} (p^k)^T \nabla f(x^k) &= -\nabla f(x^k)^T \nabla^2 f(x^k)^{-1} \nabla f(x^k) \\ &\leq -\frac{1}{L} \|\nabla f(x^k)\|^2 \\ &\leq -\frac{\gamma}{L} \|\nabla f(x^k)\| \|p^k\|. \end{aligned}$$

1301 We can use the Newton direction in the line-search framework of Subsection 4.5  
1302 to obtain a method for which  $x^k \rightarrow x^*$ , where  $x^*$  is the (unique) global minimizer  
1303 of  $f$ . (This claim follows from the property (4.5.6) together with the fact that  $x^*$  is  
1304 the only point for which  $\nabla f(x^*) = 0$ .) We can even obtain a complexity result —  
1305 and  $O(1/\sqrt{T})$  bound on  $\min_{0 \leq k \leq T-1} \|\nabla f(x^k)\|$  — from Theorem 4.5.3.

1306 These global convergence properties are enhanced by the local quadratic con-  
 1307 vergence property of Theorem 7.1.6 if we modify the line-search framework by  
 1308 accepting the step length  $\alpha_k = 1$  in (4.0.1) whenever it satisfies the weak Wolfe  
 1309 conditions (4.5.2). (It can be shown, by again using arguments based on Taylor's  
 1310 theorem (Theorem 3.3.1), that these conditions *will* be satisfied by  $\alpha_k = 1$  for all  
 1311  $x^k$  sufficiently close to the minimizer  $x^*$ .)

1312 Consider now the case in which  $f$  is convex and satisfies condition (3.3.6) but  
 1313 is not strongly convex. Here, the Hessian  $\nabla^2 f(x^k)$  may be singular for some  $k$ , so  
 1314 the direction (7.1.4) may not be well defined. However, by adding any positive  
 1315 number  $\lambda_k > 0$  to the diagonal, we can ensure that the modified Newton direction  
 1316 defined by

$$1317 \quad (7.2.1) \quad p^k = -[\nabla^2 f(x^k) + \lambda_k I]^{-1} \nabla f(x^k),$$

1318 is well defined and is a descent direction for  $f$ . For any  $\eta \in (0, 1)$  in (4.5.1),  
 1319 we have by choosing  $\lambda_k$  large enough that  $\lambda_k / (L + \lambda_k) \geq \eta$  that the condition  
 1320 (4.5.1) is satisfied too, so we can use the resulting direction  $p^k$  in the line-search  
 1321 framework of Subsection 4.5, to obtain a method that convergence to a solution  
 1322  $x^*$  of (1.0.1), when one exists.

1323 If, in addition, the minimizer  $x^*$  is unique and satisfies a second-order suffi-  
 1324 cient condition (so that  $\nabla^2 f(x^*)$  is positive definite), then  $\nabla^2 f(x^k)$  will be positive  
 1325 definite too for  $k$  sufficiently large. Thus, provided that  $\eta$  is sufficiently small,  
 1326 the *unmodified* Newton direction (with  $\lambda_k = 0$  in (7.2.1)) will satisfy the condi-  
 1327 tion (4.5.1). If we use (7.2.1) in the line-search framework of Section 4.5, but set  
 1328  $\lambda_k = 0$  where possible, and accept  $\alpha_k = 1$  as the step length whenever it satisfies  
 1329 (4.5.2), we can obtain local quadratic convergence to  $x^*$ , in addition to the global  
 1330 convergence and complexity promised by Theorem 4.5.3.

1331 **7.3. Newton Methods for Nonconvex Functions** For smooth nonconvex  $f$ , the  
 1332 Hessian  $\nabla^2 f(x^k)$  may be indefinite for some  $k$ . The Newton direction (7.1.4)  
 1333 may not exist (when  $\nabla^2 f(x^k)$  is singular) or it may not be a descent direction  
 1334 (when  $\nabla^2 f(x^k)$  has negative eigenvalues). However, we can still define a modified  
 1335 Newton direction as in (7.2.1), which will be a descent direction for  $\lambda_k$  sufficiently  
 1336 large, and thus can be used in the line-search framework of Section 4.5. For a  
 1337 given  $\eta$  in (4.5.1), a sufficient condition for  $p^k$  from (7.2.1) to satisfy (4.5.1) is that

$$1338 \quad \frac{\lambda_k + \lambda_{\min}(\nabla^2 f(x^k))}{\lambda_k + L} \geq \eta,$$

1339 where  $\lambda_{\min}(\nabla^2 f(x^k))$  is the minimum eigenvalue of the Hessian, which may be  
 1340 negative. The line-search framework of Section 4.5 can then be applied to ensure  
 1341 that  $\nabla f(x^k) \rightarrow 0$ .

1342 Once again, if the iterates  $\{x^k\}$  enter the neighborhood of a local solution  $x^*$   
 1343 for which  $\nabla^2 f(x^*)$  is positive definite, some enhancements of the strategy for  
 1344 choosing  $\lambda_k$  and the step length  $\alpha_k$  can recover the local quadratic convergence  
 1345 of Theorem 7.1.6.

1346 Formula (7.2.1) is not the only way to modify the Newton direction to ensure  
 1347 descent in a line-search framework. Other approaches are outlined in [36, Chap-  
 1348 ter 3]. One such technique is to modify the Cholesky factorization of  $\nabla^2 f(x^k)$  by  
 1349 adding positive elements to the diagonal only as needed to allow the factoriza-  
 1350 tion to proceed (that is, to avoid taking the square root of a negative number),  
 1351 then using the modified factorization in place of  $\nabla^2 f(x^k)$  in the calculation of the  
 1352 Newton step  $p^k$ . Another technique is to compute an eigenvalue decomposition  
 1353  $\nabla^2 f(x^k) = Q_k \Lambda_k Q_k^T$  (where  $Q_k$  is orthogonal and  $\Lambda_k$  is the diagonal matrix con-  
 1354 taining the eigenvalues), then define  $\tilde{\Lambda}_k$  to be a modified version of  $\Lambda_k$  in which  
 1355 all the diagonals are positive. Then, following (7.1.4),  $p^k$  can be defined as

$$1356 \quad p^k := -Q_k \tilde{\Lambda}_k^{-1} Q_k^T \nabla f(x^k).$$

1357 When an appropriate strategy is used to define  $\tilde{\Lambda}_k$ , we can ensure satisfaction  
 1358 of the descent condition (4.5.1) for some  $\eta > 0$ . As above, the line-search frame-  
 1359 work of Section 4.5 can be used to obtain an algorithm that generates a sequence  
 1360  $\{x^k\}$  such that  $\nabla f(x^k) \rightarrow 0$ . We noted earlier that this condition ensures that all  
 1361 accumulation points  $\hat{x}$  are stationary points, that is, they satisfy  $\nabla f(\hat{x}) = 0$ .

1362 Stronger guarantees can be obtained from a *trust-region* version of Newton's  
 1363 method, which ensures convergence to a point satisfying second-order necessary  
 1364 conditions, that is,  $\nabla^2 f(\hat{x}) \succeq 0$  in addition to  $\nabla f(\hat{x}) = 0$ . The trust-region approach  
 1365 was developed in the late 1970s and early 1980s, and has become popular again  
 1366 recently because of this appealing global convergence behavior. A trust-region  
 1367 Newton method also recovers quadratic convergence to solutions  $x^*$  satisfying  
 1368 second-order-sufficient conditions, without any special modifications. (The trust-  
 1369 region Newton approach is closely related to cubic regularization [26, 35], which  
 1370 we discuss in the next section.)

1371 We now outline the trust-region approach. (Further details can be found in  
 1372 [36, Chapter 4].) The subproblem to be solved at each iteration is

$$1373 \quad (7.3.1) \quad \min_d f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d \quad \text{subject to } \|d\|_2 \leq \Delta_k.$$

1374 The objective is a second-order Taylor-series approximation while  $\Delta_k$  is the *radius*  
 1375 of the trust region — the region within which we trust the second-order model  
 1376 to capture the true behavior of  $f$ . Somewhat surprisingly, the problem (7.3.1) is  
 1377 not too difficult to solve, even when the Hessian  $\nabla^2 f(x^k)$  is indefinite. In fact, the  
 1378 solution  $d^k$  of (7.3.1) satisfies the linear system

$$1379 \quad (7.3.2) \quad [\nabla^2 f(x^k) + \lambda I] d^k = -\nabla f(x^k), \quad \text{for some } \lambda \geq 0,$$

1380 where  $\lambda$  is chosen such that  $\nabla^2 f(x^k) + \lambda I$  is positive semidefinite and  $\lambda > 0$  only if  
 1381  $\|d^k\| = \Delta_k$  (see [31]). Solving (7.3.1) thus reduces to a search for the appropriate  
 1382 value of the scalar  $\lambda_k$ , for which specialized methods have been devised.

1383 For large-scale problems, it may be too expensive to solve (7.3.1) near-exactly,  
 1384 since the process may require several factorizations of an  $n \times n$  matrix (namely,  
 1385 the coefficient matrix in (7.3.2), for different values of  $\lambda$ ). A popular approach

1386 for finding approximate solutions of (7.3.1), which can be used when  $\nabla^2 f(x^k)$   
 1387 is positive definite, is the *dogleg* method. In this method the curved path traced  
 1388 out by solutions of (7.3.2) for values of  $\lambda$  in the interval  $[0, \infty)$  is approximated  
 1389 by simpler path consisting of two line segments. The first segment joins 0 to  
 1390 the point  $d_C^k$  that minimizes the objective in (7.3.1) along the direction  $-\nabla f(x^k)$ ,  
 1391 while the second segment joins  $d_C^k$  to the pure Newton step defined in (7.1.4). The  
 1392 approximate solution is taken to be the point at which this “dogleg” path crosses  
 1393 the boundary of the trust region  $\|d\| \leq \Delta_k$ . If the dogleg path lies entirely inside  
 1394 the trust region, we take  $d^k$  to be the pure Newton step. See [36, Section 4.1].

1395 Having discussed the trust-region subproblem (7.3.1), let us outline how it can  
 1396 be used as the basis for a complete algorithm. A crucial role is played by the ratio  
 1397 between the *amount of decrease in  $f$  predicted by the quadratic objective in (7.3.1)* and  
 1398 the *actual decrease in  $f$ , namely,  $f(x^k) - f(x^k + d^k)$* . Ideally, this ratio would be close  
 1399 to 1. If it is at least greater than a small tolerance (say,  $10^{-4}$ ) we accept the step  
 1400 and proceed to the next iteration. Otherwise, we conclude that the trust-region  
 1401 radius  $\Delta_k$  is too large, so we do not take the step, shrink the trust region, and  
 1402 re-solve (7.3.1) to obtain a new step. Additionally, when the actual-to-predicted  
 1403 ratio is close to 1, we conclude that a larger trust region may hasten progress, so  
 1404 we increase  $\Delta$  for the next iteration, provided that the bound  $\|d^k\| \leq \Delta_k$  really is  
 1405 active at the solution of (7.3.1).

1406 Unlike a basic line-search method, the trust-region Newton method can “es-  
 1407 cape” from a saddle point. Suppose we have  $\nabla f(x^k) = 0$  and  $\nabla^2 f(x^k)$  indefinite  
 1408 with some strictly negative eigenvalues. Then, the solution  $d^k$  to (7.3.1) will be  
 1409 nonzero, and the algorithm will step away from the saddle point, in the direc-  
 1410 tion of most negative curvature for  $\nabla^2 f(x^k)$ . Another appealing feature of the  
 1411 trust-region Newton approach is that when the sequence  $\{x^k\}$  approaches a point  
 1412  $x^*$  satisfying second-order sufficient conditions, the trust region bound becomes  
 1413 inactive, and the method takes pure Newton steps (7.1.4) for all sufficiently large  
 1414  $k$  so the local quadratic convergence that characterizes Newton’s method.

1415 The basic difference between line-search and trust-region methods can be sum-  
 1416 marized as follows. Line-search methods first choose a direction  $p^k$ , then decide  
 1417 how far to move along that direction. Trust-region methods do the opposite: They  
 1418 choose the distance  $\Delta_k$  first, then find the direction that makes the best progress  
 1419 for this step length.

1420 **7.4. A Cubic Regularization Approach** Trust-region Newton methods have the  
 1421 significant advantage of guaranteeing that any accumulation points will satisfy  
 1422 second-order necessary conditions. A related approach based on *cubic regulariza-*  
 1423 *tion* has similar properties, plus some additional complexity guarantees. Cubic  
 1424 regularization requires the Hessian to be Lipschitz continuous, as in (7.1.2). It  
 1425 follows that the following cubic function yields a *global upper* bound for  $f$ :

$$1426 \quad (7.4.1) \quad T_M(z; x) := f(x) + \nabla f(x)^T(z - x) + \frac{1}{2}(z - x)^T \nabla^2 f(x)(z - x) + \frac{M}{6} \|z - x\|^3.$$

1427 Specifically, we have for any  $x$  that

$$1428 \quad f(z) \leq T_M(z; x), \quad \text{for all } z.$$

1429 The basic cubic regularization algorithm starting from  $x^0$  proceeds as follows:

$$1430 \quad (7.4.2) \quad x^{k+1} = \arg \min_z T_M(z; x^k), \quad k = 0, 1, 2, \dots$$

1431 The complexity properties of this approach were analyzed in [35], with variants  
1432 being studied in [26] and [12, 13]. Rather than present the theory for the method  
1433 based on (7.4.2), we describe an elementary algorithm that makes use of the ex-  
1434 pansion (7.4.1) as well as the steepest-descent theory of Subsection 4.1. Our algo-  
1435 rithm aims to identify a point that *approximately* satisfies second-order necessary  
1436 conditions, that is,

$$1437 \quad (7.4.3) \quad \|\nabla f(x)\| \leq \epsilon_g, \quad \lambda_{\min}(\nabla^2 f(x)) \geq -\epsilon_H,$$

1438 where  $\epsilon_g$  and  $\epsilon_H$  are two small constants. In addition to Lipschitz continuity of  
1439 the Hessian (7.1.2), we assume Lipschitz continuity of the gradient with constant  
1440  $L$  (see (3.3.6)), and also that the objective  $f$  is lower-bounded by some number  $\bar{f}$ .

1441 Our algorithm takes steps of two types: a steepest-descent step, as in Subsec-  
1442 tion 4.1, or a step in a negative curvature direction for  $\nabla^2 f$ . Iteration  $k$  proceeds  
1443 as follows:

- 1444 (i) If  $\|\nabla f(x^k)\| > \epsilon_g$ , take the steepest descent step (4.1.1).  
1445 (ii) Otherwise, if  $\lambda_{\min}(\nabla^2 f(x^k)) < -\epsilon_H$ , choose  $p^k$  to be the eigenvector cor-  
1446 responding to the most negative eigenvalue of  $\nabla^2 f(x^k)$ . Choose the size  
1447 and sign of  $p^k$  such that  $\|p^k\| = 1$  and  $(p^k)^\top \nabla f(x^k) \leq 0$ , and set

$$1448 \quad (7.4.4) \quad x^{k+1} = x^k + \alpha_k p^k, \quad \text{where } \alpha_k = \frac{2\epsilon_H}{M}.$$

1449 If neither of these conditions hold, then  $x^k$  satisfies the approximate second-order  
1450 necessary conditions (7.4.3), so we terminate.

1451 For the steepest-descent step (i), we have from (4.1.3) that

$$1452 \quad (7.4.5) \quad f(x^{k+1}) \leq f(x^k) - \frac{1}{2L} \|\nabla f(x^k)\|^2 \leq f(x^k) - \frac{\epsilon_g^2}{2L}.$$

1453 For a step of type (ii), we have from (7.4.1) that

$$1454 \quad (7.4.6) \quad \begin{aligned} f(x^{k+1}) &\leq f(x^k) + \alpha_k \nabla f(x^k)^\top p^k + \frac{1}{2} \alpha_k^2 (p^k)^\top \nabla^2 f(x^k) p^k + \frac{1}{6} M \alpha_k^3 \|p^k\|^3 \\ &\leq f(x^k) - \frac{1}{2} \left( \frac{2\epsilon_H}{M} \right)^2 \epsilon_H + \frac{1}{6} M \left( \frac{2\epsilon_H}{M} \right)^3 \\ &= f(x^k) - \frac{2}{3} \frac{\epsilon_H^3}{M^2}. \end{aligned}$$

1455 By aggregating (7.4.5) and (7.4.6), we have that at each  $x^k$  for which the condition  
1456 (7.4.3) does *not* hold, we attain a decrease in the objective of at least

$$1457 \quad \min \left( \frac{\epsilon_g^2}{2L}, \frac{2}{3} \frac{\epsilon_H^3}{M^2} \right).$$

1458 Using the lower bound  $\bar{f}$  on the objective  $f$ , we see that the number of iterations  
 1459  $K$  required must satisfy the condition

$$1460 \quad K \min \left( \frac{\epsilon_g^2}{2L}, \frac{2}{3} \frac{\epsilon_H^3}{M^2} \right) \leq f(x^0) - \bar{f},$$

1461 from which we conclude that

$$1462 \quad K \leq \max \left( 2L\epsilon_g^{-2}, \frac{3}{2} M^2 \epsilon_H^{-3} \right) (f(x^0) - \bar{f}).$$

1463 We also observe that that the maximum number of iterates required to identify a  
 1464 point at which only the approximate stationarity condition  $\|\nabla f(x^k)\| \leq \epsilon_g$  holds  
 1465 is  $2L\epsilon_g^{-2}(f(x^0) - \bar{f})$ . (We can just omit the second-order part of the algorithm.)  
 1466 Note too that it is easy to devise *approximate* versions of this algorithm with simi-  
 1467 lar complexity. For example, the negative curvature direction  $p^k$  in step (ii) above  
 1468 can be replaced by an approximation to the direction of most negative curvature,  
 1469 obtained by the Lanczos iteration with random initialization.

1470 In algorithms that make more complete use of the cubic model (7.4.1), the term  
 1471  $\epsilon_g^{-2}$  in the complexity expression becomes  $\epsilon_g^{-3/2}$ , and the constants are different.  
 1472 The subproblems (7.4.1) are more complicated to solve than those in the simple  
 1473 scheme above. Active research is going on into other algorithms that achieve  
 1474 complexities similar to those of the cubic regularization approach. A variety of  
 1475 methods that make use of Newton-type steps, approximate negative curvature di-  
 1476 rections, accelerated gradient methods, random perturbations, randomized Lanc-  
 1477 zos and conjugate gradient methods, and other algorithmic elements have been  
 1478 proposed.

## 1479 8. Conclusions

1480 We have outlined various algorithmic tools from optimization that are useful  
 1481 for solving problems in data analysis and machine learning, and presented their  
 1482 basic theoretical properties. The intersection of optimization and machine learn-  
 1483 ing is a fruitful and very popular area of current research. All the major machine  
 1484 learning conferences have a large contingent of optimization papers, and there is  
 1485 a great deal of interest in developing algorithmic tools to meet new challenges  
 1486 and in understanding their properties. The edited volume [41] contains a snap-  
 1487 shot of the state of the art circa 2010, but this is a fast-moving field and there have  
 1488 been many developments since then.

## 1489 Acknowledgments

1490 I thank Ching-pei Lee for a close reading and many helpful suggestions, and  
 1491 David Hong and an anonymous referee for detailed, excellent comments.

1492

## References

- 1493 [1] L. Balzano, R. Nowak, and B. Recht, *Online identification and tracking of subspaces from highly incom-*  
 1494 *plete information*, 48th Annual Allerton Conference on Communication, Control, and Computing,  
 1495 2010, pp. 704–711. ←9
- 1496 [2] L. Balzano and S. J. Wright, *Local convergence of an algorithm for subspace identification from partial*  
 1497 *data*, Foundations of Computational Mathematics **14** (2014), 1–36. DOI: 10.1007/s10208-014-9227-  
 1498 7. ←10
- 1499 [3] A. Beck and M. Teboulle, *A fast iterative shrinkage-threshold algorithm for linear inverse problems*,  
 1500 SIAM Journal on Imaging Sciences **2** (2009), no. 1, 183–202. ←32, 35
- 1501 [4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, *A training algorithm for optimal margin classifiers*,  
 1502 Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 1992, pp. 144–  
 1503 152. ←12
- 1504 [5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learn-*  
 1505 *ing via the alternating direction methods of multipliers*, Foundations and Trends in Machine Learning  
 1506 **3** (2011), no. 1, 1–122. ←3
- 1507 [6] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2003. ←3
- 1508 [7] S. Bubeck, *Convex optimization: Algorithms and complexity*, Foundations and Trends in Machine  
 1509 Learning **8** (2015), no. 3–4, 231–357. ←35, 36
- 1510 [8] S. Bubeck, Y. T. Lee, and M. Singh, *A geometric alternative to Nesterov’s accelerated gradient descent*,  
 1511 Technical Report [arXiv:1506.08187](https://arxiv.org/abs/1506.08187), Microsoft Research, 2015. ←35
- 1512 [9] S. Burer and R. D. C. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs*  
 1513 *via low-rank factorizations*, Mathematical Programming, Series B **95** (2003), 329–257. ←7
- 1514 [10] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computa-  
 1515 tional Mathematics **9** (2009), 717–772. ←7
- 1516 [11] E. J. Candès, X. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, Journal of the ACM  
 1517 **58.3** (2011), 11. ←9
- 1518 [12] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Adaptive cubic regularisation methods for unconstrained op-*  
 1519 *timization. Part I: Motivation, convergence and numerical results*, Mathematical Programming, Series  
 1520 A **127** (2011), 245–295. ←46
- 1521 [13] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Adaptive cubic regularisation methods for unconstrained*  
 1522 *optimization. Part II: Worst-case function-and-derivative-evaluation complexity*, Mathematical Program-  
 1523 ming, Series A **130** (2011), no. 2, 295–319. ←46
- 1524 [14] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, *Rank-sparsity incoherence for*  
 1525 *matrix decomposition*, SIAM Journal on Optimization **21** (2011), no. 2, 572–596. ←9
- 1526 [15] Y. Chen and M. J. Wainwright, *Fast low-rank estimation by projected gradient descent: General statistical*  
 1527 *and algorithmic guarantees*, Technical Report [arXiv:1509.03025](https://arxiv.org/abs/1509.03025), University of California-Berkeley,  
 1528 2015. ←9
- 1529 [16] C. Cortes and V. N. Vapnik, *Support-vector networks*, Machine Learning **20** (1995), 273–297. ←12
- 1530 [17] A. d’Aspremont, O. Banerjee, and L. El Ghaoui, *First-order methods for sparse covariance selection*,  
 1531 SIAM Journal on Matrix Analysis and Applications **30** (2008), 56–66. ←8
- 1532 [18] A. d’Aspremont, L. El Ghaoui, M. I. Jordan, and G. Lanckriet, *A direct formulation for sparse PCA*  
 1533 *using semidefinite programming*, SIAM Review **49** (2007), no. 3, 434–448. ←8
- 1534 [19] T. Dasu and T. Johnson, *Exploratory data mining and data cleaning*, John Wiley & Sons, 2003. ←4
- 1535 [20] P. Drineas and M. W. Mahoney, *Lectures on randomized numerical linear algebra*, The mathematics  
 1536 of data, 2018. ←6
- 1537 [21] D. Drusvyatskiy, M. Fazel, and S. Roy, *An optimal first-order method based on optimal quadratic av-*  
 1538 *eraging*, Technical Report [arXiv:1604.06543](https://arxiv.org/abs/1604.06543), University of Washington, 2016. To appear in SIAM  
 1539 Journal on Optimization. ←35
- 1540 [22] J. C. Duchi, *Introductory lectures on stochastic optimization*, The mathematics of data, 2018. ←3, 16,  
 1541 23
- 1542 [23] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point*  
 1543 *algorithm for maximal monotone operators*, Mathematical Programming **55** (1992), 293–318. ←3
- 1544 [24] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Research Logistics Quarterly  
 1545 **3** (1956), 95–110. ←28
- 1546 [25] J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*,  
 1547 Biostatistics **9** (2008), no. 3, 432–441. ←8

- 1548 [26] A. Griewank, *The modification of Newton's method for unconstrained optimization by bounding cubic*  
1549 *terms*, Technical Report NA/12, DAMTP, Cambridge University, 1981. ←44, 46
- 1550 [27] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of  
1551 Research of the National Bureau of Standards **49** (1952), 409–436. ←33
- 1552 [28] A. J. Hoffman and H. Weilandt, *The variation of the spectrum of a normal matrix*, Duke Mathematical  
1553 Journal **20** (1953), no. 1, 37–39. ←41
- 1554 [29] J. D Lee, M. Simchowitz, M. I Jordan, and B. Recht, *Gradient descent only converges to minimizers*,  
1555 Conference on learning theory, 2016, pp. 1246–1257. ←27
- 1556 [30] D. C. Liu and J. Nocedal, *On the limited-memory BFGS method for large scale optimization*, Mathe-  
1557 matical Programming **45** (1989), 503–528. ←3, 34
- 1558 [31] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM Journal on Scientific and  
1559 Statistical Computing **4** (1983), 553–572. ←44
- 1560 [32] A. S. Nemirovski and D. B. Yudin, *Problem complexity and method efficiency in optimization*, John  
1561 Wiley, 1983. ←39
- 1562 [33] Y. Nesterov, *A method for unconstrained convex problem with the rate of convergence  $O(1/k^2)$* , Dok-  
1563 lady AN SSSR **269** (1983), 543–547. ←32
- 1564 [34] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, Springer Science and Busi-  
1565 ness Media, New York, 2004. ←32, 36
- 1566 [35] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math-  
1567 ematical Programming, Series A **108** (2006), 177–205. ←44, 46
- 1568 [36] J. Nocedal and S. J. Wright, *Numerical Optimization*, Second, Springer, New York, 2006. ←3, 26,  
1569 34, 44, 45
- 1570 [37] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Computational  
1571 Mathematics and Mathematical Physics **4.5** (1964), 1–17. ←32
- 1572 [38] B. T. Polyak, *Introduction to optimization*, Optimization Software, 1987. ←32, 33
- 1573 [39] B. Recht, M. Fazel, and P. Parrilo, *Guaranteed minimum-rank solutions to linear matrix equations via*  
1574 *nuclear norm minimization*, SIAM Review **52** (2010), no. 3, 471–501. ←7
- 1575 [40] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, N.J., 1970. ←17
- 1576 [41] S. Sra, S. Nowozin, and S. J. Wright (eds.), *Optimization for machine learning*, NIPS Workshop  
1577 Series, MIT Press, 2011. ←47
- 1578 [42] R. Tibshirani, *Regression shrinkage and selection via the LASSO*, Journal of the Royal Statistical  
1579 Society B **58** (1996), 267–288. ←6
- 1580 [43] M. J. Todd, *Semidefinite optimization*, Acta Numerica **10** (2001), 515–560. ←3
- 1581 [44] B. Turlach, W. N. Venables, and S. J. Wright, *Simultaneous variable selection*, Technometrics **47**  
1582 (2005), no. 3, 349–363. ←9
- 1583 [45] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review **38** (1996), 49–95. ←3
- 1584 [46] S. J. Wright, *Primal-dual interior-point methods*, SIAM, Philadelphia, PA, 1997. ←3
- 1585 [47] S. J. Wright, *Coordinate descent algorithms*, Mathematical Programming, Series B **151** (2015Decem-  
1586 ber), 3–34. ←3
- 1587 [48] X. Yi, D. Park, Y. Chen, and C. Caramanis, *Fast algorithms for robust PCA via gradient descent*,  
1588 Advances in Neural Information Processing Systems **29**, 2016, pp. 4152–4160. ←9
- 1589 Computer Sciences Department, University of Wisconsin-Madison, Madison, WI 53706  
1590 E-mail address: swright@cs.wisc.edu