# Fundamentals of Optimization Theory With Applications to Machine Learning

Jean Gallier and Jocelyn Quaintance
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
e-mail: jean@cis.upenn.edu



December 16, 2019

# Preface

In recent years, computer vision, robotics, machine learning, and data science have been some of the key areas that have contributed to major advances in technology. Anyone who looks at papers or books in the above areas will be baffled by a strange jargon involving exotic terms such as kernel PCA, ridge regression, lasso regression, support vector machines (SVM), Lagrange multipliers, KKT conditions, *etc.* Do support vector machines chase cattle to catch them with some kind of super lasso? No! But one will quickly discover that behind the jargon which always comes with a new field (perhaps to keep the outsiders out of the club), lies a lot of "classical" linear algebra and techniques from optimization theory. And there comes the main challenge: in order to understand and use tools from machine learning, computer vision, and so on, one needs to have a firm background in linear algebra and optimization theory. To be honest, some probability theory and statistics should also be included, but we already have enough to contend with.

Many books on machine learning struggle with the above problem. How can one understand what are the dual variables of a ridge regression problem if one doesn't know about the Lagrangian duality framework? Similarly, how is it possible to discuss the dual formulation of SVM without a firm understanding of the Lagrangian framework?

The easy way out is to sweep these difficulties under the rug. If one is just a consumer of the techniques we mentioned above, the cookbook recipe approach is probably adequate. But this approach doesn't work for someone who really wants to do serious research and make significant contributions. To do so, we believe that one must have a solid background in linear algebra and optimization theory.

This is a problem because it means investing a great deal of time and energy studying these fields, but we believe that perseverance will be amply rewarded.

This second volume covers some elements of optimization theory and applications, especially to machine learning. This volume is divided in five parts:

(1) Preliminaries of Optimization Theory.

(2) Linear Optimization.

(3) Nonlinear Optimization.

(4) Applications to Machine Learning.

(5) An appendix consisting of two chapers; one on Hilbert bases and the Riesz–Fischer theorem, the other one containing `Matlab` code.

Part I is devoted to some preliminaries of optimization theory. The goal of most optimization problems is to minimize (or maximize) some objective function $J$ subject to equality or inequality constraints. Therefore it is important to understand when a function $J$ has a minimum or a maximum (an optimum). In most optimization problems, we need to find necessary conditions for a function $J\colon \Omega \to \mathbb{R}$ to have a local extremum with respect to a subset $U$ of $\Omega$ (where $\Omega$ is open). This can be done in two cases:

(1) The set $U$ is defined by a set of equations,

$$U = \{x \in \Omega \mid \varphi_i(x) = 0,\ \ 1 \leq i \leq m\},$$

where the functions $\varphi_i\colon \Omega \to \mathbb{R}$ are continuous (and usually differentiable).

(2) The set $U$ is defined by a set of inequalities,

$$U = \{x \in \Omega \mid \varphi_i(x) \leq 0,\ \ 1 \leq i \leq m\},$$

where the functions $\varphi_i\colon \Omega \to \mathbb{R}$ are continuous (and usually differentiable).

The case of equality constraints is much easier to deal with and is treated in Chapter 4.

In the case of equality constraints, a necessary condition for a local extremum with respect to $U$ can be given in terms of *Lagrange multipliers*.

Part II deals with the special case where the objective function is a linear form and the constraints are affine inequality and equality constraints. This subject is known as *linear programming*, and the next four chapters give an introduction to the subject.

Part III is devoted to nonlinear optimization, which is the case where the objective function $J$ is not linear and the constaints are inequality constraints. Since it is practically impossible to say anything interesting if the constraints are not convex, we quickly consider the convex case.

Chapter 13 is devoted to some general results of optimization theory. A main theme is to find sufficient conditions that ensure that an objective function has a minimum which is achieved. We define gradient descent methods (including Newton's method), and discuss their convergence.

Chapter 14 contains the most important results of nonlinear optimization theory. Theorem 14.6 gives necessary conditions for a function $J$ to have a minimum on a subset $U$ defined by convex inequality constraints in terms of the Karush–Kuhn–Tucker conditions. Furthermore, if $J$ is also convex and if the KKT conditions hold, then $J$ has a global minimum.

We illustrate the KKT conditions on an interesting example from machine learning the so-called *hard margin support vector machine*; see Sections 14.5 and 14.6. The problem is to separate two disjoint sets of points, $\{u_i\}_{i=1}^p$ and $\{v_j\}_{j=1}^q$, using a hyperplane satisfying some optimality property (to maximize the margin).

Section 14.7 contains the most important results of the chapter. The notion of Lagrangian duality is presented and we discuss *weak duality* and *strong duality*.

In Chapter 15, we consider some deeper aspects of the the theory of convex functions that are not necessarily differentiable at every point of their domain. Some substitute for the gradient is needed. Fortunately, for convex functions, there is such a notion, namely *subgradients*. A major motivation for developing this more sophisticated theory of differentiation of convex functions is to extend the Lagrangian framework to convex functions that are not necessarily differentiable.

Chapter 16 is devoted to the presentation of one of the best methods known at the present for solving optimization problems involving equality constraints, called ADMM (alternating direction method of multipliers). In fact, this method can also handle more general constraints, namely, membership in a convex set. It can also be used to solve *lasso minimization*.

In Section 16.4, we prove the convergence of ADMM under exactly the same assumptions as in Boyd et al. [17]. It turns out that Assumption (2) in Boyd et al. [17] implies that the matrices $A^\top A$ and $B^\top B$ are invertible (as we show after the proof of Theorem 16.1). This allows us to prove a convergence result stronger than the convergence result proven in Boyd et al. [17].

The next four chapters constitute Part IV, which covers some applications of optimization theory (in particular Lagrangian duality) to machine learning.

Chapter 17 is an introduction to positive definite kernels and the use of kernel functions in machine learning called a *kernel function*.

We illustrate the kernel methods on kernel PCA.

In Chapter 18 we return to the problem of separating two disjoint sets of points, $\{u_i\}_{i=1}^p$ and $\{v_j\}_{j=1}^q$, but this time we do not assume that these two sets are separable. To cope with nonseparability, we allow points to invade the safety zone around the separating hyperplane, and even points on the wrong side of the hyperplane. Such a method is called *soft margin support vector machine (SVM)*. We discuss variations of this method, including $\nu$-SV classification. In each case we present a careful derivation of the dual. We prove rigorous results about the existence of support vectors.

In Chapter 19, we discuss *linear regression*, *ridge regression*, *lasso regression* and *elastic net regression*.

In Chapter 20 we present *$\nu$-SV Regression*. This method is designed in the same spirit as soft margin SVM, in the sense that it allows a margin of error. Here the errors are penalized in the $\ell^1$-sense. We present a careful derivation of the dual and discuss the existence of support vectors.

The methods presented in Chapters 18, 19 and 20 have all been implemented in `Matlab`, and much of this code is given in Appendix B. Remarkably, ADMM emerges as the main engine for solving most of these optimization problems. Thus it is nice to see the continuum spanning from theoretical considerations of convergence and correctness to practical matters of implementation. It is fun to see how these abstract Lagrange multipliers yield concrete results such as the weight vector $w$ defining the desired hyperplane in regression or SVM.

Except for a few exceptions we provide complete proofs. We did so to make this book self-contained, but also because we believe that no deep knowledge of this material can be acquired without working out some proofs. However, our advice is to skip some of the proofs upon first reading, especially if they are long and intricate.

The chapters or sections marked with the symbol $\circledast$ contain material that is typically more specialized or more advanced, and they can be omitted upon first (or second) reading.

*Acknowledgement*: We would like to thank Christine Allen-Blanchette, Kostas Daniilidis, Carlos Esteves, Spyridon Leonardos, Stephen Phillips, João Sedoc, Stephen Shatz, Jianbo Shi, and Marcelo Siqueira, for reporting typos and for helpful comments. Thanks to Gilbert Strang. We learned much from his books which have been a major source of inspiration. Special thanks to Steven Boyd. We learned a lot from his remarkable book on convex optimization and his papers, and Part III of our book is significantly inspired by his writings. The first author also wishes to express his deepest gratitute to Philippe G. Ciarlet who was his teacher and mentor in 1970-1972 while he was a student at ENPC in Paris. Professor Ciarlet was by far his best teacher. He also knew how to instill in his students the importance of intellectual rigor, honesty, and modesty. He still has his typewritten notes on measure theory and integration, and on numerical linear algebra. The latter became his wonderful book Ciarlet [25], from which we have borrowed heavily.

# Contents

# Chapter 1

# Introduction

This second volume covers some elements of optimization theory and applications, especially to machine learning. This volume is divided in five parts:

(1) Preliminaries of Optimization Theory.

(2) Linear Optimization.

(3) Nonlinear Optimization.

(4) Applications to Machine Learning.

(5) An appendix consisting of two chapers; one on Hilbert bases and the Riesz–Fischer theorem, the other one containing `Matlab` code.

Part I is devoted to some preliminaries of optimization theory. The goal of most optimization problems is to minimize (or maximize) some objective function $J$ subject to equality or inequality constraints. Therefore it is important to understand when a function $J$ has a minimum or a maximum (an optimum). If the function $J$ is sufficiently differentiable, then a necessary condition for a function to have an optimum typically involves the derivative of the function $J$, and if $J$ is real-valued, its gradient $\nabla J$.

Thus it is desirable to review some basic notions of topology and calculus, in particular, to have a firm grasp of the notion of derivative of a function between normed vector spaces. Partial derivatives $\partial f / \partial A$ of functions whose range and domain are spaces of matrices tend to be used casually, even though in most cases a correct definition is never provided. It is possible, and simple, to define rigorously derivatives, gradients, and directional derivatives of functions defined on matrices and to avoid these nonsensical partial derivatives.

Chapter 2 contains a review of basic topological notions used in analysis. We pay particular attention to complete metric spaces and complete normed vector spaces. In fact, we provide a detailed construction of the completion of a metric space (and of a normed vector space) using equivalence classes of Cauchy sequences. Chapter 3 is devoted to some notions of differential calculus, in particular, directional derivatives, total derivatives, gradients, Hessians, and the inverse function theorem.

Chapter 4 deals with extrema of real-valued functions. In most optimization problems, we need to find necessary conditions for a function $J\colon \Omega \to \mathbb{R}$ to have a local extremum with respect to a subset $U$ of $\Omega$ (where $\Omega$ is open). This can be done in two cases:

(1) The set $U$ is defined by a set of equations,

$$U = \{x \in \Omega \mid \varphi_i(x) = 0,\ 1 \leq i \leq m\},$$

where the functions $\varphi_i\colon \Omega \to \mathbb{R}$ are continuous (and usually differentiable).

(2) The set $U$ is defined by a set of inequalities,

$$U = \{x \in \Omega \mid \varphi_i(x) \leq 0,\ 1 \leq i \leq m\},$$

where the functions $\varphi_i\colon \Omega \to \mathbb{R}$ are continuous (and usually differentiable).

In (1), the equations $\varphi_i(x) = 0$ are called *equality constraints*, and in (2), the inequalities $\varphi_i(x) \leq 0$ are called *inequality constraints*. The case of equality constraints is much easier to deal with and is treated in Chapter 4.

If the functions $\varphi_i$ are convex and $\Omega$ is convex, then $U$ is convex. This is a very important case that we will discuss later. In particular, if the functions $\varphi_i$ are affine, then the equality constraints can be written as $Ax = b$, and the inequality constraints as $Ax \leq b$, for some $m \times n$ matrix $A$ and some vector $b \in \mathbb{R}^m$. We will also discuss the case of affine constraints later.

In the case of equality constraints, a necessary condition for a local extremum with respect to $U$ can be given in terms of *Lagrange multipliers*. In the case of inequality constraints, there is also a necessary condition for a local extremum with respect to $U$ in terms of generalized Lagrange multipliers and the *Karush–Kuhn–Tucker* conditions. This will be discussed in Chapter 14.

In Chapter 5 we discuss Newton's method and some of its generalizations (the Newton–Kantorovich theorem). These are methods to find the zeros of a function.

Chapter 6 covers the special case of determining when a quadratic function has a minimum, subject to affine equality constraints. A complete answer is provided in terms of the notion of symmetric positive semidefinite matrices.

The Schur complement is introduced in Chapter 7. We give a complete proof of a criterion for a matrix to be positive definite (or positive semidefinite) stated in Boyd and Vandenberghe [18] (Appendix B).

Part II deals with the special case where the objective function is a linear form and the constraints are affine inequality and equality constraints. This subject is known as linear programming, and the next four chapters give an introduction to the subject. Although linear programming has been supplanted by convex programming and its variants, it is still a great workhorse. It is also a great warm up for the general treatment of Lagrangian duality. We pay particular attention to versions of Farkas' lemma, which is at the heart of duality in linear programming.

Part III is devoted to nonlinear optimization, which is the case where the objective function $J$ is not linear and the constaints are inequality constraints. Since it is practically impossible to say anything interesting if the constraints are not convex, we quickly consider the convex case.

In optimization theory one often deals with function spaces of infinite dimension. Typically, these spaces either are Hilbert spaces or can be completed as Hilbert spaces. Thus it is important to have some minimum knowledge about Hilbert spaces, and we feel that this minimum knowledge includes the projection lemma, the fact that a closed subset has an orthogonal complement, the Riesz representation theorem, and a version of the Farkas–Minkowski lemma. Chapter 12 covers these topics. A more detailed introduction to Hilbert spaces is given in Appendix A.

Chapter 13 is devoted to some general results of optimization theory. A main theme is to find sufficient conditions that ensure that an objective function has a minimum which is achieved. We define the notion of a coercive function. The most general result is Theorem 13.2, which applies to a coercive convex function on a convex subset of a separable Hilbert space. In the special case of a coercive quadratic functional, we obtain the Lions–Stampacchia theorem (Theorem 13.6), and the Lax–Milgram theorem (Theorem 13.7). We define elliptic functionals, which generalize quadratic functions defined by symmetric positive definite matrices. We define gradient descent methods, and discuss their convergence. A gradient descent method looks for a descent direction and a stepsize parameter, which is obtained either using an exact line search or a backtracking line search. A popular technique to find the search direction is steepest descent. In addition to steepest descent for the Euclidean norm, we discuss steepest descent for an arbitrary norm. We also consider a special case of steepest descent, Newton's method. This method converges faster than the other gradient descent methods, but it is quite expensive since it requires computing and storing Hessians. We also present the method of conjugate gradients and prove its correctness. We briefly discuss the method of gradient projection and the penalty method in the case of constrained optima.

Chapter 14 contains the most important results of nonlinear optimization theory. We begin by defining the cone of feasible directions and then state a necessary condition for a function to have local minimum on a set $U$ that is not necessarily convex in terms of the cone of feasible directions. The cone of feasible directions is not always convex, but it is if the constraints are inequality constraints. An inequality constraint $\varphi(u) \leq 0$ is said to be *active* is $\varphi(u) = 0$. One can also define the notion of *qualified constraint*. Theorem 14.5 gives necessary conditions for a function $J$ to have a minimum on a subset $U$ defined by qualified inequality constraints in terms of the Karush–Kuhn–Tucker conditions (for short

KKT conditions), which involve nonnegative Lagrange multipliers. The proof relies on a version of the Farkas–Minkowski lemma. Some of the KTT conditions assert that $\lambda_i \varphi_i(u) = 0$, where $\lambda_i \geq 0$ is the Lagrange multiplier associated with the constraint $\varphi_i \leq 0$. To some extent, this implies that active constaints are more important than inactive constraints, since if $\varphi_i(u) < 0$ is an inactive constraint, then $\lambda_i = 0$. In general, the KKT conditions are useless unlesss the constraints are convex. In this case, there is a manageable notion of qualified constraint given by Slater's conditions. Theorem 14.6 gives necessary conditions for a function $J$ to have a minimum on a subset $U$ defined by convex inequality constraints in terms of the Karush–Kuhn–Tucker conditions. Furthermore, if $J$ is also convex and if the KKT conditions hold, then $J$ has a global minimum.

In Section 14.4, we apply Theorem 14.6 to the special case where the constraints are equality constraints, which can be expressed as $Ax = b$. In the special case where the convex objective function $J$ is a convex quadratic functional of the form

$$J(x) = \frac{1}{2} x^\top P x + q^\top x + r,$$

where $P$ is a $n \times n$ symmetric positive semidefinite matrix, the necessary and sufficient conditions for having a minimum are expressed by a linear system involving a matrix called the KKT matrix. We discuss conditions that guarantee that the KKT matrix is invertible, and how to solve the KKT system. We also briefly discuss variants of Newton's method dealing with equality constraints.

We illustrate the KKT conditions on an interesting example, the so-called hard margin support vector machine; see Sections 14.5 and 14.6. The problem is a classification problem, or more accurately a separation problem. Suppose we have two nonempty disjoint finite sets of $p$ *blue* points $\{u_i\}_{i=1}^p$ and $q$ *red* points $\{v_j\}_{j=1}^q$ in $\mathbb{R}^n$. Our goal is to find a hyperplane $H$ of equation $w^\top x - b = 0$ (where $w \in \mathbb{R}^n$ is a nonzero vector and $b \in \mathbb{R}$), such that all the blue points $u_i$ are in one of the two open half-spaces determined by $H$, and all the red points $v_j$ are in the other open half-space determined by $H$.

If the two sets are indeed separable, then in general there are infinitely many hyperplanes separating them. Vapnik had the idea to find a hyperplane that maximizes the smallest distance between the points and the hyperplane. Such a hyperplane is indeed unique and is called a maximal hard margin hyperplane, or hard margin support vector machine. The support vectors are those for which the constraints are active.

Section 14.7 contains the most important results of the chapter. The notion of Lagrangian duality is presented. Given a primal optimization problem $(P)$ consisting in minimizing an objective function $J(v)$ with respect to some inequality constraints $\varphi_i(v) \leq 0$, $i = 1, \ldots, m$, we define the *dual function* $G(\mu)$ as the result of minimizing the Lagrangian

$$L(v, \mu) = J(v) + \sum_{i=1}^m \mu_i \varphi_i(v)$$

with respect to $v$, with $\mu \in \mathbb{R}^m_+$. The dual program (D) is then to maximize $G(\mu)$ with respect to $\mu \in \mathbb{R}^m_+$. It turns out that $G$ is a concave function, and the dual program is an unconstrained maximization. This is actually a misleading statement because $G$ is generally a partial function, so maximizing $G(\mu)$ is equivalent to a constrained maximization problem in which the constraints specify the domain of $G$, but in many cases, we obtain a dual program simpler than the primal program. If $d^*$ is the optimal value of the dual program and if $p^*$ is the optimal value of the primal program, we always have

$$d^* \leq p^*,$$

which is known as *weak duality*. Under certain conditions, $d^* = p^*$, that is, the duality gap is zero, in which case we say that *strong duality* holds. Also, under certain conditions, a solution of the dual yields a solution of the primal, and if the primal has an optimal solution, then the dual has an optimal solution, but beware that the converse is generally false (see Theorem 14.17). We also show how to deal with equality constraints, and discuss the use of conjugate functions to find the dual function. Our coverage of Lagrangian duality is quite thorough, but we do not discuss more general orderings such as the semidefinite ordering. For these topics which belong to convex optimization, the reader is referred to Boyd and Vandenberghe [18].

In Chapter 15, we consider some deeper aspects of the the theory of convex functions that are not necessarily differentiable at every point of their domain. Some substitute for the gradient is needed. Fortunately, for convex functions, there is such a notion, namely *subgradients*. Geometrically, given a (proper) convex function $f$, the subgradients at $x$ are vectors normal to supporting hyperplanes to the epigraph of the function at $(x, f(x))$. The *subdifferential* $\partial f(x)$ to $f$ at $x$ is the set of all subgradients at $x$. A crucial property is that $f$ is differentiable at $x$ iff $\partial f(x) = \{\nabla f_x\}$, where $\nabla f_x$ is the gradient of $f$ at $x$. Another important property is that a (proper) convex function $f$ attains its minimum at $x$ iff $0 \in \partial f(x)$. A major motivation for developing this more sophisticated theory of "differentiation" of convex functions is to extend the Lagrangian framework to convex functions that are not necessarily differentiable.

Experience shows that the applicability of convex optimization is significantly increased by considering extended real-valued functions, namely functions $f\colon S \to \mathbb{R} \cup \{-\infty, +\infty\}$, where $S$ is some subset of $\mathbb{R}^n$ (usually convex). This is reminiscent of what happens in measure theory, where it is natural to consider functions that take the value $+\infty$.

In Section 15.1, we introduce extended real-valued functions, which are functions that may also take the values $\pm\infty$. In particular, we define proper convex functions, and the closure of a convex function. Subgradients and subdifferentials are defined in Section 15.2. We discuss some properties of subgradients in Section 15.3 and Section 15.4. In particular, we relate subgradients to one-sided directional derivatives. In Section 15.5, we discuss the problem of finding the minimum of a proper convex function and give some criteria in terms of subdifferentials. In Section 15.6, we sketch the generalization of the results presented in

Chapter 14 about the Lagrangian framework to programs allowing an objective function and inequality constraints which are convex but not necessarily differentiable.

This chapter relies heavily on Rockafellar [61]. We tried to distill the body of results needed to generalize the Lagrangian framework to convex but not necessarily differentiable functions. Some of the results in this chapter are also discussed in Bertsekas [9, 12, 10].

Chapter 16 is devoted to the presentation of one of the best methods known at the present for solving optimization problems involving equality constraints, called ADMM (alternating direction method of multipliers). In fact, this method can also handle more general constraints, namely, membership in a convex set. It can also be used to solve *lasso minimization*.

In this chapter, we consider the problem of minimizing a convex function $J$ (not necessarily differentiable) under the equality constraints $Ax = b$. In Section 16.1, we discuss the dual ascent method. It is essentially gradient descent applied to the dual function $G$, but since $G$ is maximized, gradient descent becomes gradient ascent.

In order to make the minimization step of the dual ascent method more robust, one can use the trick of adding the penalty term $(\rho/2)\left\|Au - b\right\|_2^2$ to the Lagrangian. We obtain the *augmented Lagrangian*

$$L_\rho(u, \lambda) = J(u) + \lambda^\top (Au - b) + (\rho/2)\left\|Au - b\right\|_2^2,$$

with $\lambda \in \mathbb{R}^m$, and where $\rho > 0$ is called the *penalty parameter*. We obtain the minimization Problem $(P_\rho)$,

$$\begin{aligned} &\text{minimize} && J(u) + (\rho/2)\left\|Au - b\right\|_2^2 \\ &\text{subject to} && Au = b, \end{aligned}$$

which is equivalent to the original problem.

The benefit of adding the penalty term $(\rho/2)\left\|Au - b\right\|_2^2$ is that by Proposition 15.37, Problem $(P_\rho)$ has a unique optimal solution under mild conditions on $A$. Dual ascent applied to the dual of $(P_\rho)$ is called the *method of multipliers* and is discussed in Section 16.2.

The new twist in ADMM is to split the function $J$ into two independent parts, as $J(x, z) = f(x) + g(z)$, and to consider the Minimization Problem $(P_{\mathrm{admm}})$,

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && Ax + Bz = c, \end{aligned}$$

for some $p \times n$ matrix $A$, some $p \times m$ matrix $B$, and with $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, and $c \in \mathbb{R}^p$. We also assume that $f$ and $g$ are convex.

As in the method of multipliers, we form the augmented Lagrangian

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^\top (Ax + Bz - c) + (\rho/2)\left\|Ax + Bz - c\right\|_2^2,$$

with $\lambda \in \mathbb{R}^p$ and for some $\rho > 0$. The major difference with the method of multipliers is that instead of performing a minimization step jointly over $x$ and $z$, ADMM first performs an $x$-minimization step and then a $z$-minimization step. Thus $x$ and $z$ are updated in an alternating or sequential fashion, which accounts for the term *alternating direction*. Because the Lagrangian is augmented, some mild conditions on $A$ and $B$ imply that these minimization steps are guaranteed to terminate. ADMM is presented in Section 16.3.

In Section 16.4, we prove the convergence of ADMM under exactly the same assumptions as in Boyd et al. [17]. It turns out that Assumption (2) in Boyd et al. [17] implies that the matrices $A^\top A$ and $B^\top B$ are invertible (as we show after the proof of Theorem 16.1). This allows us to prove a convergence result stronger than the convergence result proven in Boyd et al. [17]. In particular, we prove that *all* of the sequences $(x^k)$, $(z^k)$, and $(\lambda^k)$ converge to optimal solutions $(\widetilde{x}, \widetilde{z})$, and $\widetilde{\lambda}$.

In Section 16.5, we discuss stopping criteria. In Section 16.6, we present some applications of ADMM, in particular, minimization of a proper closed convex function $f$ over a closed convex set $C$ in $\mathbb{R}^n$ and quadratic programming. The second example provides one of the best methods for solving quadratic problems, in particular, the SVM problems discussed in Chapter 18. Section 16.8 gives applications of ADMM to $\ell^1$-norm problems, in particular, lasso regularization which plays an important role in machine learning.

The next four chapters constitute Part IV, which covers some applications of optimization theory (in particular Lagrangian duality) to machine learning.

Chapter 17 is an introduction to positive definite kernels and the use of kernel functions in machine learning.

Let $X$ be a nonempty set. If the set $X$ represents a set of highly nonlinear data, it may be advantageous to map $X$ into a space $F$ of much higher dimension called the *feature space*, using a function $\varphi\colon X \to F$ called a *feature map*. This idea is that $\varphi$ "unwinds" the description of the objects in $F$ in an attempt to make it linear. The space $F$ is usually a vector space equipped with an inner product $\langle -, - \rangle$. If $F$ is infinite dimensional, then we assume that it is a Hilbert space.

Many algorithms that analyze or classify data make use of the inner products $\langle \varphi(x), \varphi(y) \rangle$, where $x, y \in X$. These algorithms make use of the function $\kappa\colon X \times X \to \mathbb{C}$ given by

$$\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle, \qquad x, y \in X,$$

called a *kernel function*.

The kernel trick is to pretend that we have a feature embedding $\varphi\colon X \to F$ (actuallly unknown), but to only use inner products $\langle \varphi(x), \varphi(y) \rangle$ that can be evaluated using the original data through the known kernel function $\kappa$. It turns out that the functions of the form $\kappa$ as above can be defined in terms of a condition which is reminiscent of positive semidefinite matrices (see Definition 17.2). Furthermore, every function satisfying Definition 17.2 arises from a suitable feature map into a Hilbert space; see Theorem 17.8.

We illustrate the kernel methods on kernel PCA (see Section 17.4).

In Chapter 18 we return to the problem of separating two disjoint sets of points, $\{u_i\}_{i=1}^p$ and $\{v_j\}_{j=1}^q$, but this time we do not assume that these two sets are separable. To cope with nonseparability, we allow points to invade the safety zone around the separating hyperplane, and even points on the wrong side of the hyperplane. Such a method is called soft margin support vector machine. We discuss variations of this method, including $\nu$-SV classification. In each case we present a careful derivation of the dual and we explain how to solve it using ADMM. We prove rigorous results about the existence of support vectors.

In Chapter 19 we discuss linear regression. This problem can be cast as a learning problem. We observe a sequence of (distinct) pairs $((x_1, y_1), \ldots, (x_m, y_m))$ called a *set of training data*, where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, viewed as input-output pairs of some unknown function $f$ that we are trying to infer. The simplest kind of function is a linear function $f(x) = x^\top w$, where $w \in \mathbb{R}^n$ is a vector of coefficients usually called a *weight vector*. Since the problem is overdetermined and since our observations may be subject to errors, we can't solve for $w$ exactly as the solution of the system $Xw = y$, so instead we solve the least-squares problem of minimizing $\|Xw - y\|_2^2$, where $X$ is the $m \times n$ matrix whose *rows* are the row vectors $x_i^\top$. In general there are still infinitely many solutions so we add a regularizing term. If we add the term $K\|w\|_2^2$ to the objective function $J(w) = \|Xw - y\|_2^2$, then we have *ridge regression*. This problem is discussed in Section 19.1.

We derive the dual program. The dual has a unique solution which yields a solution of the primal. However, the solution of the dual is given in terms of the matrix $XX^\top$ (whereas the solution of the primal is given in terms of $X^\top X$), and since our data points $x_i$ are represented by the rows of the matrix $X$, we see that this solution only involves inner products of the $x_i$. This observation is the core of the idea of kernel functions, which we introduce. We also explain how to solve the problem of learning an affine function $f(x) = x^\top w + b$.

In general the vectors $w$ produced by ridge regression have few zero entries. In practice it is highly desirable to obtain sparse solutions, that is, vectors $w$ with many components equal to zero. This can be achieved by replacing the regularizing term $K\|w\|_2^2$ by the regularizing term $K\|w\|_1$; that is, to use the $\ell^1$-norm instead of the $\ell^2$-norm; see Section 19.4. This method has the exotic name of *lasso regression*. This time there is no closed-form solution, but this is a convex optimization problem and there are efficient iterative methods to solve it. We show that ADMM provides an efficient solution.

Lasso has some undesirable properties, in particular when the dimension of the data is much larger than the number of data. In order to alleviate these problems, *elastic net regression* penalizes $w$ with *both* an $\ell^2$ regularizing term $K\|w\|_2^2$ and an $\ell^1$ regularizing term $\tau\|w\|_1$. The method of elastic net blends ridge regression and lasso and attempts to retain their best properties; see Section 19.6. It can also be solved using ADMM but it appears to be much slower than lasso when $K$ is small and the dimension of the data is much larger than the number of data.

In Chapter 20 we present $\nu$-SV Regression. This method is designed in the same spirit as soft margin SVM, in the sense that it allows a margin of error. Here the errors are penalized in the $\ell^1$-sense. We discuss several variations of the method and show how to solve them using ADMM. We present a careful derivation of the dual and discuss the existence of support vectors.