Comput Manag Sci (2014) 11:319–340
DOI 10.1007/s10287-013-0181-7

ORIGINAL PAPER

Numerical study of learning algorithms on Stiefel manifold

Takafumi Kanamori · Akiko Takeda

Received: 21 June 2012 / Accepted: 26 July 2013 / Published online: 10 August 2013

Abstract Convex optimization methods are used for many machine learning models such as support vector machine. However, the requirement of a convex formulation can place limitations on machine learning models. In recent years, a number of machine learning methods not requiring convexity have emerged. In this paper, we study non-convex optimization problems on the Stiefel manifold in which the feasible set consists of a set of rectangular matrices with orthonormal column vectors. We present examples of non-convex optimization problems in machine learning and apply three nonlinear optimization methods for finding a local optimal solution; geometric gradient descent method, augmented Lagrangian method of multipliers, and alternating direction method of multipliers. Although the geometric gradient method is often used to solve non-convex optimization problems on the Stiefel manifold, we show that the alternating direction method of multipliers generally produces higher quality numerical solutions within a reasonable computation time.

Keywords Non-convex optimization · Stiefel manifold · Alternating direction method of multipliers · Dimensionality reduction

T. Kanamori
Nagoya University, Furocho, Chikusa-ku, Nagoya-shi,
Aichi 464-8603, Japan
e-mail: kanamori@is.nagoya-u.ac.jp

A. Takeda (✉)
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
e-mail: takeda@mist.i.u-tokyo.ac.jp

1 Introduction

Convex optimization is widely used to solve statistical problems such as classification and high-dimensional regression problems (Sra et al. 2011). Convexity is an important feature of high-performance learning algorithms. However, convex formulations have limitations. In recent years, a number of machine learning methods that do not require convexity have emerged. For example, submodular maximization in structured learning, as discussed in Krause and Guestrin (2008), and difference of convex (d.c.) programming in clustering, feature selection, etc. (An et al. 2008). Efficient learning algorithms for non-convex optimization problems have a wide range of statistical data analysis applications.

In this paper, we study optimization problems on the Stiefel manifold that is defined as the set of all rectangular matrices with orthonormal column vectors. The problems are formulated as nonlinear optimization, and our purpose is to find a local optimal solution of those optimization problems. We show several machine learning problems that can be expressed in this framework.

1.1 Examples of learning models on Stiefel manifold

Dimensionality reduction problems: A lower dimensional representation of the original data structure is estimated in dimensionality reduction problems. The projection of the data onto a subspace should preserve the original data structure as precisely as possible. The subspace is expressed by using a matrix in the Stiefel manifold; i.e., the column vectors of the matrix correspond to a set of the basis vectors of the subspace. Hence, dimensionality reduction problems can be formulated as optimization problems on the Stiefel manifold.

Independent component analysis: We describe independent component analysis (ICA) as another application. Suppose that the observed data is a linear mixture of signal components. The goal of ICA is to recover the original signals on the assumption that the components of the signals are independent of each other. To recover the signals, we estimate the mixing matrix from the observed data. Without loss of generality, let us assume that the multi-dimensional data has the zero mean vector and the variance-covariance matrix is the identity matrix. When the dimension of the signals and that of the observed data are the same, the problem is to find an appropriate orthogonal transformation that converts the data to the signals with independent components. When the dimension of the signals is lower than that of the observed data, the mixing matrix is expressed as a matrix in the Stiefel manifold. Hence, ICA is formulated as the maximization of the degree of the independence on the Stiefel manifold.

Robust classification models: Recently, Takeda et al. (2012) proposed robust classification models that unify many learning algorithms from the perspective of robust optimization (Ben-Tal et al. 2009). The model includes an unit norm equality constraint, that is a special case of the Stiefel manifold.

1.2 Standard local optimization method for optimization on Stiefel manifold

The geometric gradient method has been often used to solve non-convex optimization problems on the Stiefel manifold (Nishimori and Akaho 2005). Indeed, for that purpose many authors (Absil et al. 2008; Edelman et al. 1998; Nishimori and Akaho 2005) have studied geometric constraints consisting of matrix manifolds. The geometric properties of matrix manifolds are investigated by using Riemannian geometry. By taking geometric properties into account, we can develop efficient numerical algorithms on manifolds. A drawback of the geometric approach is that we need to calculate analytic solutions of geometric quantities such as the gradient direction and the geodesic defined in terms of the Riemannian metric.

1.3 Our local optimization approach: alternating direction method of multipliers

The purpose of this paper is to show the superiority of the *alternating direction method of multipliers* (ADMM), a variant of the *augmented Lagrangian method of multipliers* (AL), over the geometric gradient method for non-convex optimization problems on the Stiefel manifold. The AL and ADMM are popular methods in nonlinear programming (Luenberger and Ye 2008). Learning algorithms based on these simple yet powerful optimization methods have attracted much attention in machine learning community; for example, ADMM has been used to solve complex problems with huge datasets (Boyd et al. 2011). The local convergence property of ADMM for non-convex constraint problems was theoretically studied by Zhang (2010).

We evaluate the performance of these numerical algorithms at solving non-convex optimization problems on the Stiefel manifold. In particular, we deal with classification problems based on robust classification models (Takeda et al. 2012) and dimensionality reduction problems for estimating the density ratio (Sugiyama et al. 2012). Since these problems are relatively new in the machine learning community, efficient optimization algorithms for them should be developed.

This paper is organized as follows. Section 2 describes the formulation of optimization problems on the Stiefel manifold. Then, it describes the optimization algorithms, i.e., the geometric gradient method, the augmented Lagrangian method of multipliers and the alternating direction method of multipliers. In Sect. 3, we study the stability and efficiency of optimization algorithms by using condition number analysis. Applying these optimization methods to an eigenvalue problem, we clarify the behavior of each optimization method. Section 4 introduces a robust classification model for classification problems. The robust classification model is formulated as a non-convex optimization problem on the unit sphere. We apply the optimization methods to this model and compare their numerical performances. Section 5 discusses the problem of estimating the density ratio with dimensionality reduction. The density ratio is a ratio of two probability densities, and it is used in a great variety of statistical problems (Sugiyama et al. 2012). Here, the Stiefel manifold is used to represent a subspace of Euclidean space. The dimensionality reduction onto the subspace can be formulated as a minimization problem on the Stiefel manifold. This section evaluates the non-convex optimization algorithms for dimensionality reduction problems.

Section 6 summarizes the results of the numerical studies and shows future research directions.

Let us start by describing the notation to be used throughout the paper. $\mathbb{R}^{n\times p}$ denotes the set of all n by p matrices. For a matrix A, A^T denotes its transposition. For two matrices $A, B \in \mathbb{R}^{n\times p}$, $A \bullet B$ is the canonical inner product, i.e., $A \bullet B = \sum_{i=1}^{n}\sum_{j=1}^{p} A_{ij}B_{ij}$. The norm $\|A\|_F^2$ of the matrix A is defined by the canonical inner product. The p-dimensional identity matrix is expressed as $\boldsymbol{I}_p \in \mathbb{R}^{p\times p}$, and the zero-matrix is denoted as $\boldsymbol{O}$. The size of the zero-matrix is not explicitly specified if there is no confusion. The column vector is written in boldface, e.g., $\boldsymbol{x} \in \mathbb{R}^n$, and the zero vector is denoted as $\mathbf{0}$. The norm on the Euclidean space is denoted as $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$.

2 Optimization on Stiefel manifold

The Stiefel manifold, denoted as $\mathcal{S}_{n,p}$, consists of n by p rectangular matrices whose column vectors are orthonormal, i.e.,

$$\mathcal{S}_{n,p} = \{W \in \mathbb{R}^{n\times p} : W^T W = \boldsymbol{I}_p\},$$

where $n \geq p$ is assumed. The Stiefel manifold with $n = p$ is equivalent to the set of all orthogonal matrices. On the other hand, the Stiefel manifold with $p = 1$ is reduced to the unit sphere in $\mathbb{R}^n$.

The optimization problem on the Stiefel manifold can be formulated as

$$\min_{W} f(W) \quad \text{subject to} \quad W \in \mathcal{S}_{n,p}, \tag{1}$$

where $f(W)$ is a real-valued function on the set of rectangular matrices $\mathbb{R}^{n\times p}$. Below, we introduce three optimization methods for solving problem (1).

2.1 Geometric approach

In the geometric approach, the gradient descent direction is computed on the basis of the Riemannian structure of the Stiefel manifold. Let $\nabla_W f(W) \in \mathbb{R}^{n\times p}$ be the gradient of the function f,

$$(\nabla_W f(W))_{ij} = \frac{\partial f}{\partial W_{ij}}(W),$$

and $\mathrm{grad} f(W) \in \mathbb{R}^{n\times p}$ be the steepest gradient direction of f with respect to the Riemannian metric on the Stiefel manifold. For $t \in \mathbb{R}$, let $\phi(W,t)$ be the geodesic on the Stiefel manifold satisfying $\phi(W,0) = W$ and $\frac{d}{dt}\phi(W,0) = -\mathrm{grad} f(W)$. The geodesic $\phi(W,t)$ is represented as

$$\phi(W,t) = \exp\{-t(\nabla_W f(W) W^T - W\nabla_W f(W)^T)\}W, \tag{2}$$

where $\exp\{\cdots\}$ is the matrix exponential (Edelman et al. 1998; Nishimori and Akaho 2005). Note that $\phi(W,t) \in \mathcal{S}_{n,p}$ holds for all $t \in \mathbb{R}$, since the matrix exponential of a skew symmetric matrix is an orthogonal matrix. The line search for the one-dimensional optimization problem,

$$\min_{t \geq 0} f(\phi(W,t)),$$

determines the step length of t from the present point W to the gradient descent direction. The use of quasi-geodesics is a promising way to reduce the computational cost of the matrix exponential. Details on the geometric algorithm on the Stiefel manifold can be found in Nishimori and Akaho (2005).

2.2 Augmented Lagrangian method of multipliers

We introduce a versatile method called the augmented Lagrangian method of multipliers, or AL for short. The augmented Lagrangian of problem (1) is defined as

$$L_\alpha(W,\Lambda) = \alpha f(W) - \Lambda \bullet (W^T W - \boldsymbol{I}_p) + \frac{1}{2}\|W^T W - \boldsymbol{I}_p\|_F^2, \tag{3}$$

where Λ is a p by p matrix and α is a positive number. The dual ascent method for (3) yields the following update formula:

$$\begin{aligned} W_{t+1} &:= \operatorname*{argmin}_{W \in \mathbb{R}^{n \times p}} L_\alpha(W, \Lambda_t), \\ \Lambda_{t+1} &:= \Lambda_t - (W_{t+1}^T W_{t+1} - \boldsymbol{I}_p). \end{aligned} \tag{4}$$

The augmented Lagrangian method with sufficiently small positive α generates a convergent sequence under a modest condition on the objective function f. The method is described in detail in (Luenberger and Ye 2008, Chap. 14).

Note that the optimization of $L_\alpha(W,\Lambda_t)$ might be ill-conditioned when α is small and the optimal solution of $L_\alpha(W,\Lambda_t)$ is close to the Stiefel manifold. An example of an ill-conditioned problem is presented in Sect. 3.

2.3 Alternating direction method of multipliers

A survey of ADMM and its applications are found in Boyd et al. (2011). We will describe the algorithm of the ADMM on the Stiefel manifold by following the presentation of Zhang (2010). By introducing an extra parameter $V \in \mathbb{R}^{n \times p}$, problem (1) can be represented as

$$\min_{V,W \in \mathbb{R}^{n \times p}} f(V) \quad \text{subject to} \quad V - W = \boldsymbol{O}, \quad W^T W - \boldsymbol{I}_p = \boldsymbol{O}.$$

Here, let us define the partial augmented Lagrangian as

$$\mathcal{L}_\alpha(V, W, \Lambda) = \alpha f(V) - \Lambda \bullet (V - W) + \frac{1}{2}\|V - W\|_F^2, \tag{5}$$

where Λ is an n by p matrix and α is a positive number. The constraint $W^T W - \boldsymbol{I}_p = \boldsymbol{O}$ is not included in (5). Different from AL, ADMM has two stages, i.e., the update of V and update of W. In each iteration of the ADMM using $\mathcal{L}_\alpha(W, V, \Lambda)$, the parameters V, W and Λ are updated as follows:

$$V_{t+1} := \operatorname*{argmin}_{V \in \mathbb{R}^{n\times p}} \mathcal{L}_\alpha(V, W_t, \Lambda_t) = \operatorname*{argmin}_{V \in \mathbb{R}^{n\times p}} \alpha f(V) + \frac{1}{2}\|V - W_t - \Lambda_t\|_F^2, \tag{6}$$

$$\begin{aligned} W_{t+1} &:= \operatorname*{argmin}_{W : W^T W = \boldsymbol{I}_p} \mathcal{L}_\alpha(V_{t+1}, W, \Lambda_t) = \operatorname*{argmin}_{W : W^T W = \boldsymbol{I}_p} \|W - (V_{t+1} - \Lambda_t)\|_F^2, \\ \Lambda_{t+1} &:= \Lambda_t - (V_{t+1} - W_{t+1}). \end{aligned} \tag{7}$$

The above computation is repeated until $f(W_t)$ converges. Note that the optimal solution of (7) is found by the singular value decomposition of $V_{t+1} - \Lambda_t$, whereas (6) is solved by using nonlinear optimization. The solution of (7) always satisfies the constraint $W^T W = \boldsymbol{I}_p$. Hence, in each iteration, ADMM produces a feasible solution of (1).

3 Stability and efficiency of computation

The stability and efficiency of the optimization algorithm is governed by the condition number of the Hessian matrix of the objective function (Demmel 1997; Luenberger and Ye 2008). Here, we perform a condition number analysis to study the computational properties of AL and ADMM.

3.1 Condition number analysis

In the ADMM algorithm, we need to solve (6). When the parameter α is small, the condition number of the Hessian of the objective function in (6) can be approximated by the condition number determined from the quadratic term in (6). Clearly, the condition number of the Hessian of the quadratic term $\|V - W_t - \Lambda_t\|_F^2$ with respect to V is equal to 1. This result implies that ADMM does not significantly worsen the condition number of the Hessian of the objective function.

Now let us compute the condition number of the Hessian of the objective function in AL. Suppose that $n > p$ holds for $\mathcal{S}_{n,p}$. We show that the condition number of the Hessian of the objective function in the AL can be extremely large. We repeatedly solve the problem,

$$\min_{W \in \mathcal{S}_{n,p}} \alpha f(W) - \Lambda \bullet (W^T W - \boldsymbol{I}_p) + \frac{1}{2}\|W^T W - \boldsymbol{I}_p\|_F^2, \tag{8}$$

for a given matrix $A \in \mathbb{R}^{p \times p}$. Suppose that α is small and that the optimal solution of (8) is close to the Stiefel manifold. Then, the condition number determined by the third term becomes dominant. The Hessian matrix of the third term is equal to

$$4\boldsymbol{I}_p \otimes WW^T + 2W^T W \otimes \boldsymbol{I}_n - 2\boldsymbol{I}_p \otimes \boldsymbol{I}_n, \tag{9}$$

where $\otimes$ is the Kronecker product of matrices. Let $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ be the eigenvalues of $W^T W \in \mathbb{R}^{p \times p}$, where $W \in \mathbb{R}^{n \times p}$ is not necessarily a member of the Stiefel manifold. In addition, let us define $\lambda_{p+1} = \cdots = \lambda_n = 0$. The singular values of (9) are given by $|4\lambda_i + 2\lambda_a - 2|$ for $i = 1, \ldots, n$ and $a = 1, \ldots, p$. The maximum singular value is equal to $\max_{i,a} |4\lambda_i + 2\lambda_a - 2|$, which is greater than or equal to $|6\lambda_1 - 2|$. The minimum singular value is $\min_{i,a} |4\lambda_i + 2\lambda_a - 2|$. Hence, the condition number is greater than or equal to

$$\frac{|6\lambda_1 - 2|}{\min_{i,a} |4\lambda_i + 2\lambda_a - 2|}.$$

When the matrix W is close to the Stiefel manifold, all singular values λ_a, $a = 1, \ldots, p$ are close to 1. Accordingly, the denominator of the condition number,

$$\min_{i,a} |4\lambda_i + 2\lambda_a - 2|,$$

tends to zero for $n > p$, and the numerator is close to $|6 \cdot 1 - 2| = 4$. As a result, the condition number may be extremely large. This implies that AL is unstable and inefficient when the optimal solution of (8) is close to the Stiefel manifold.

The above calculation implies that ADMM is preferable to AL in view of the stability and efficiency of the computation.

3.2 Experiments for eigenvalue problem

The following eigenvalue problem is used for investigating computational stability and efficiency of the optimization algorithms described in Sect. 2:

$$\min_W \operatorname{Tr} W^T A W \quad \text{subject to} \quad W \in \mathcal{S}_{n,p}, \tag{10}$$

where A is an n by n symmetric positive definite matrix. The solution is directly given by the eigenvalue decomposition of A, and the optimal value is equal to $\sum_{i=1}^{p} \lambda_i(A)$, where $\lambda_i(A)$ stands for the i-th smallest eigenvalue of A. Note that the principal component analysis involves maximization of the objective function instead of minimization; see Bishop (2006) for details.

In the numerical simulations, the matrix A was defined as $A = B^T B$ for $B \in \mathbb{R}^{n \times n}$ whose elements were independent copies of a random variable with the standard normal distribution. The size of the matrix W was $n = 16$, $p = 15$, $n = 60$, $p = 4$ or $n = 120$, $p = 2$. The number of the parameters was $np = 240$ in all cases.

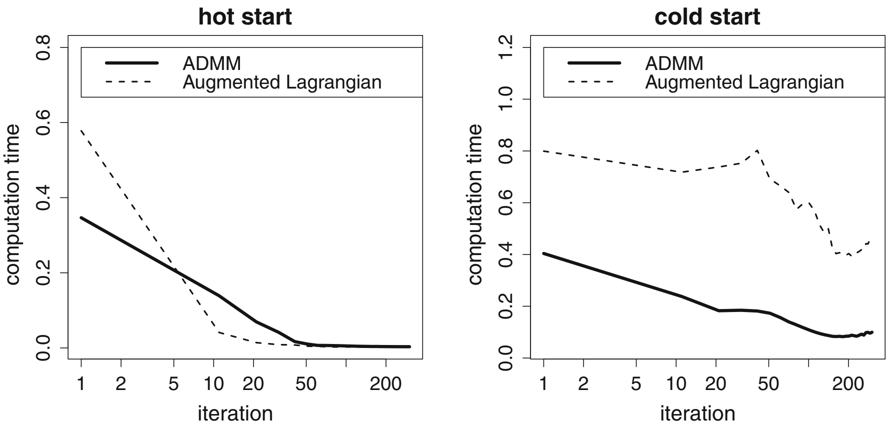


Fig. 1 Eigenvalue problem for $n = 60$, $p = 4$: The computation time of each step is shown. The *solid* (*dashed*) *line* shows the computation time of ADMM (AL). *Left panel* hot start is used to solve the inner optimization problems. *Right panel* cold start is used to solve the inner optimization problems

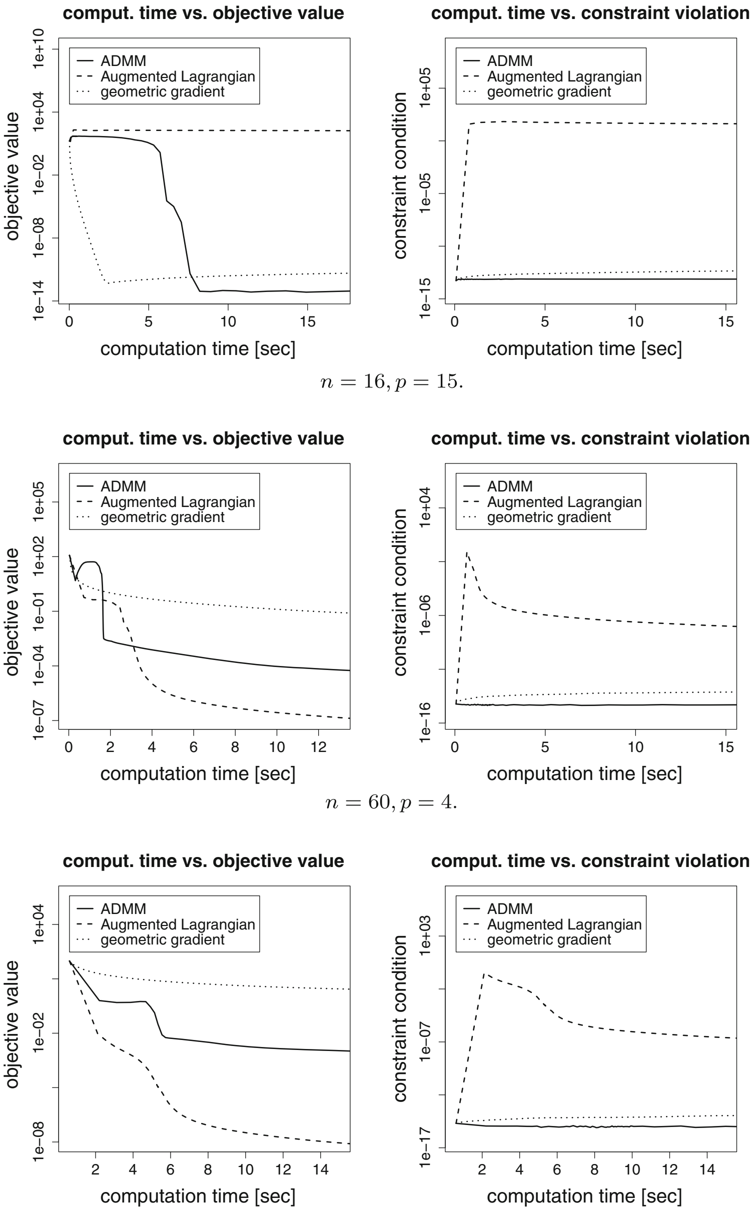
We implemented numerical algorithms with R language (R Development Core Team 2012). The command `optim` was used to solve (4) and (6). The `optim` command invokes the BFGS quasi-Newton method (Luenberger and Ye 2008). The `optimize` command was used for the line search in the geometric gradient method. The numerical experiments were conducted on a computer with an AMD Opteron Processor 6176 (2.3GHz), 128 GB of physical memory, and 0.5 MB of L2 cache and 12 MB of L3 cache running CentOS Linux release 5.4.

The objective functions in (4) and (6) have a parameter α. The parameter α may depend on the number of iterations t in order to improve the convergence speed. In preliminary experiments, we found that $\alpha = \alpha_t = 10/t$ was a good choice for both AL and ADMM.

First, we verified the results of Sect. 3 for the case of $n = 60$, $p = 4$. The computational cost of the optimization depends on the initial point of the BFGS quasi-Newton method for solving (10). The condition number analysis showed that the computational cost of solving (4) is larger than that of solving (6) when the initial point is fixed to a matrix, say W_0. The use of a fixed initial point in each iteration is called *cold-start*. In contrast, the optimal solution of the last step can be used as the initial point for the optimization in the next step. For example, the matrix W_t (resp. V_t) can be used as the initial point to solve (4) (resp. (6)). Using the solution of the last step is called *hot-start*.

Figure 1 shows the computation time needed to solve (4) and (6) in each step. In the hot-start setup, the computation time of AL was almost the same as that of ADMM. In contrast, in the cold-start setup, the computation time of AL was much longer than that of ADMM. The numerical results were thus in good agreement with the condition number analysis of Sect. 3.1. In what follows, the hot-start setup is used in all of the numerical experiments.

Next, the convergence properties of the optimization methods were investigated. Figure 2 shows the average results over 30 runs. A matrix A was randomly generated



$n = 16, p = 15.$

$n = 60, p = 4.$

$n = 120, p = 2.$

Fig. 2 Eigenvalue problem: results of numerical experiments are plotted. The size of the matrix W is $n = 16$, $p = 15$ (*upper panels*), $n = 60$, $p = 4$ (middle panels) or $n = 120$, $p = 2$ (*lower panels*). In the *left panels*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the objective value $\mathrm{Tr}\, W^T AW - \sum_{i=1}^{p} \lambda_i(A)$ that should converge to zero. In the *right panels*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the constraint condition, $\|W_t^T W_t - \boldsymbol{I}_p\|_1$. For the AL in $n = 16$, $p = 15$, the objective value and error of the constraint condition only started to decrease after a computation time of 200 s

in each run of the experiment. The objective value in the figures was given as $\mathrm{Tr}\, W^T A W - \sum_{i=1}^{p} \lambda_i(A)$ that converges to zero and the error of the constraint condition was $\|W_t^T W_t - \boldsymbol{I}_p\|_1 = \sum_{i,j=1}^{p} |(W_t^T W_t - \boldsymbol{I}_p)_{ij}|$. At the optimal solution, both the objective value and error of the constraint condition are equal to zero.

AL did not converge in the case of $n = 16,\ p = 15$ in Fig. 2. The objective value and error of the constraint condition only started to decrease after 200 s. The convergence of the AL may be improved by making a better choice of the sequence $\alpha = \alpha_t$. On the other hand, the computation of the geodesic (2) in the geometric gradient descent method is hard for large n. Thus, the convergence of the gradient descent method was slower than the other methods for $n = 60$ and $n = 120$. The constraint conditions of the gradient descent method and ADMM were satisfactory. AL did not produce a feasible solution in each step, but it decreased the objective value faster than the other methods for $n = 60$ and $n = 120$.

ADMM is preferable if the numerical solution needs to strictly satisfy the orthonormality. In contrast, AL may be a good choice if p is small enough and the objective value is more important than the feasibility of the solution, but AL is rather sensitive to the choice of α_t.

4 Robust classification models for binary classification

Here, the optimization methods presented in Sect. 2 are applied to classification problems.

4.1 Problem setup

Let $\mathcal{X} \subset \mathbb{R}^n$ be the input domain and $\{+1, -1\}$ be the set of the binary labels. Suppose that we have training samples,

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m) \in \mathcal{X} \times \{+1, -1\}.$$

Based on these samples, we predict the label for a given input point $\boldsymbol{x} \in \mathcal{X}$. For this purpose, the decision function $h(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$ is used. If $h(\boldsymbol{x})$ is positive (resp. negative), the label of $\boldsymbol{x}$ is predicted to be $+1$ (resp. -1). The equality $\boldsymbol{w}^T \boldsymbol{w} = 1$ is assumed, since scaling by a positive number does not change the sign of the decision function value. The parameters, $\boldsymbol{w}$ and b, in the decision function are estimated from the training samples. There are a number of estimation methods for binary classification problems (Bishop 2006; Hastie et al. 2001; Schölkopf and Smola 2002).

Recently, Takeda et al. (2012) proposed robust classification models that unify many learning algorithms from the perspective of robust optimization. Robust optimization (Ben-Tal et al. 2009) is an approach that handles optimization problems defined by uncertain inputs. There are several existing works (Caramanis et al. 2011; Xanthopoulos et al. 2012) which have used robust optimization for binary classification problems. However, they used robust optimization in a different way from us by making statistical learning able to handle uncertain observations.

Let us briefly describe our approach, i.e., the robust classification model. Suppose that $\boldsymbol{x}_+$ (resp. $\boldsymbol{x}_-$) is a representative of the inputs with positive (resp. negative) labels. For example, $\boldsymbol{x}_+$ can be defined as the mean vector of the samples with positive labels, and $\boldsymbol{x}_-$ can be defined in the same way for negative labels. For the decision function $h(\boldsymbol{x})$ to be good, the difference $h(\boldsymbol{x}_+) - h(\boldsymbol{x}_-) = \boldsymbol{w}^T(\boldsymbol{x}_+ - \boldsymbol{x}_-)$ should be large. The representatives, $\boldsymbol{x}_\pm$, may be uncertain, since the samples are affected by noise. Accordingly, let us define uncertainty sets for $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$. Let M_+ (resp. M_-) be the set of indices of training samples with positive (resp. negative) labels. We can construct a convex set $\mathcal{U}_+ \subset \mathbb{R}^n$ from the positive inputs $\{\boldsymbol{x}_i : i \in M_+\}$ and another convex set $\mathcal{U}_- \subset \mathbb{R}^n$ from the negative inputs $\{\boldsymbol{x}_i : i \in M_-\}$. These convex sets $\mathcal{U}_\pm$ represent the *uncertainty* of the representatives $\boldsymbol{x}_\pm$. Later, we present an example of such uncertainty sets. In the robust classification model, the parameter $\boldsymbol{w}$ in the decision function is estimated by solving the optimization problem in the worst-case setup,

$$\max_{\boldsymbol{w}:\boldsymbol{w}^T\boldsymbol{w}=1} \min_{\substack{\boldsymbol{x}_+\in\mathcal{U}_+ \\ \boldsymbol{x}_-\in\mathcal{U}_-}} \boldsymbol{w}^T(\boldsymbol{x}_+ - \boldsymbol{x}_-). \tag{11}$$

The above problem is non-convex because of the constraint $\boldsymbol{w}^T\boldsymbol{w} = 1$. However, when $\mathcal{U}_+$ and $\mathcal{U}_-$ do not intersect, the non-convex constraint can be replaced with a convex constraint, $\boldsymbol{w}^T\boldsymbol{w} \le 1$, without changing the optimal value of (11). In this case, the optimal value is positive. Note that $\mathcal{U}_+ \cap \mathcal{U}_- = \emptyset$ does not imply that the training samples are separable by a linear classifier (see Example 1 below). In contrast, if $\mathcal{U}_+$ and $\mathcal{U}_-$ intersect, the constraint $\boldsymbol{w}^T\boldsymbol{w} = 1$ can be replaced with $\boldsymbol{w}^T\boldsymbol{w} \ge 1$ without changing the optimal value of (11), and the optimal value is non-positive. (See Takeda et al. (2012) for details.) If we do not know whether or not $\mathcal{U}_+ \cap \mathcal{U}_- = \emptyset$, we need to solve problem (11) with the non-convex constraint $\boldsymbol{w}^T\boldsymbol{w} = 1$ for given uncertainty sets.

Given the estimator of $\boldsymbol{w}$, the bias term $b \in \mathbb{R}$ in the decision function $h(\boldsymbol{x})$ is estimated by applying several criteria, such as the minimum training error, or the maximum margin criterion. Here, we will not go into detail on how the bias term b is estimated.

Let us define the objective function $f(\boldsymbol{w})$ by

$$f(\boldsymbol{w}) = \max\{-\boldsymbol{w}^T(\boldsymbol{x}_+ - \boldsymbol{x}_-) : \boldsymbol{x}_+ \in \mathcal{U}_+,\ \boldsymbol{x}_- \in \mathcal{U}_-\}.$$

Accordingly, solving problem (11) is equivalent to solving

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) \quad \text{subject to } \boldsymbol{w}^T\boldsymbol{w} = 1. \tag{12}$$

The above problem is an optimization on the unit sphere, i.e, the Stiefel manifold $\mathcal{S}_{n,1}$.

Example 1 (*Ellipsoidal uncertainty set*) Let $\boldsymbol{\mu}_+$, $\boldsymbol{\mu}_-$ be mean vectors,

$$\boldsymbol{\mu}_\pm = \frac{1}{|M_\pm|} \sum_{i \in M_\pm} \boldsymbol{x}_i,$$

where $|M|$ is the size of the finite set M. For each label, the variance–covariance matrix is estimated by

$$\Sigma_\pm = \frac{1}{|M_\pm|} \sum_{i \in M_\pm} (\boldsymbol{x}_i - \boldsymbol{\mu}_\pm)(\boldsymbol{x}_i - \boldsymbol{\mu}_\pm)^T \in \mathbb{R}^{n \times n}.$$

Let us define the ellipsoidal uncertainty sets $\mathcal{U}_+$, $\mathcal{U}_-$ as

$$\mathcal{U}_\pm = \{\boldsymbol{\mu}_\pm + \Sigma_\pm^{1/2} \boldsymbol{u} \ : \ \|\boldsymbol{u}\| \le c_\pm\},$$

where c_+ and c_- are positive constants that determine the size of the ellipsoids, i.e., the degree of the uncertainty. Even if the samples are non-separable, $\mathcal{U}_+ \cap \mathcal{U}_- = \emptyset$ can occur for small tuning parameters $c_\pm$. For the ellipsoidal uncertainty sets, the objective function $f(\boldsymbol{w})$ is

$$\begin{aligned} f(\boldsymbol{w}) &= \max\{-\boldsymbol{w}^T(\boldsymbol{x}_+ - \boldsymbol{x}_-) : \boldsymbol{x}_+ \in \mathcal{U}_+, \ \boldsymbol{x}_- \in \mathcal{U}_-\} \\ &= -\boldsymbol{w}^T(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) + c_+\sqrt{\boldsymbol{w}^T \Sigma_+ \boldsymbol{w}} + c_-\sqrt{\boldsymbol{w}^T \Sigma_- \boldsymbol{w}}. \end{aligned}$$

The objective function is convex in $\boldsymbol{w} \in \mathbb{R}^n$. Upon replacing the non-convex constraint $\boldsymbol{w}^T\boldsymbol{w} = 1$ in (12) with a convex one, $\boldsymbol{w}^T\boldsymbol{w} \le 1$, problem (12) reduces to a second-order cone program (SOCP) (see, e.g., Boyd and Vandenberghe 2004).

4.2 Experiments

The optimization methods presented in Sect. 2 are applied to robust classification models (12). Perez-Cruz et al. (2003) and Takeda et al. (2012) proposed a two-stage procedure to solve (12). Their algorithms approximate the quadratic surface, $\boldsymbol{w}^T\boldsymbol{w} = 1$, by a linear one, $\boldsymbol{w}_t^T\boldsymbol{w} = 1$, with the use of a feasible solution $\boldsymbol{w}_t$ and solve an SOCP on the linear surface in every iteration until they converge. The computational cost for solving SOCPs is large. Compared with these algorithms (Perez-Cruz et al. 2003; Takeda et al. 2012), an ADMM-based algorithm is much faster and more easily implementable to solve (12), and therefore, we compare three optimization methods presented in Sect. 2.

The problem setup is the following. We assumed that the conditional probabilities, $p(\boldsymbol{x}|y = +1)$ and $p(\boldsymbol{x}|y = -1)$, were multivariate normal distributions. The dimension of the input vector $\boldsymbol{x}$ was set to $n = 100$ or $n = 300$. The conditional probability $p(\boldsymbol{x}|y = +1)$ was defined as the multivariate standard normal distribution; i.e., the mean vector was the zero vector $\boldsymbol{\mu}_+ = \boldsymbol{0} = (0, \ldots, 0) \in \mathbb{R}^n$, and the variance-covariance matrix was the identity matrix $\Sigma_+ = \boldsymbol{I}_n \in \mathbb{R}^{n \times n}$.

For the other conditional probability, $p(\boldsymbol{x}|y=-1)$, the variance-covariance matrix Σ_- was defined as $\Sigma_- = SS^T$, where each element of the n by n matrix S was an independent copy of a random variable obeying the one-dimensional standard normal distribution. The mean vector $\boldsymbol{\mu}_-$ of $p(\boldsymbol{x}|y=-1)$ was defined as $\boldsymbol{\mu}_- = (10/\sqrt{n}, \ldots, 10/\sqrt{n}) \in \mathbb{R}^n$. The marginal probability of the label was defined as $\Pr(y=+1) = \Pr(y=-1) = 0.5$; i.e., the label probability was balanced. The training sample size was set to $m = 1000$.

The robust classification model with the ellipsoidal uncertainty set defined in Example 1 was used. The mean vectors and the variance-covariance matrices were estimated on the basis of training samples. The parameter $c_\pm$ in Example 1 was set to $c_\pm = 1$ or 3. For $c_\pm = 1$, problem (11) reduces to a convex problem, because $\mathcal{U}_+$ and $\mathcal{U}_-$ do not intersect. When the parameter $c_\pm$ is large, such as $c_\pm = 3$, problem (11) is essentially non-convex.

In this experiment, we investigated the computational aspect of the learning algorithm. It is straightforward to see that there is a real number λ such that the optimality condition $\nabla f(\boldsymbol{w}) = \lambda \boldsymbol{w}$ holds if and only if $\nabla f(\boldsymbol{w}) - \boldsymbol{w}\boldsymbol{w}^T \nabla f(\boldsymbol{w}) = \boldsymbol{0}$ is satisfied. Thus, the optimality condition of problem (12) can be expressed as

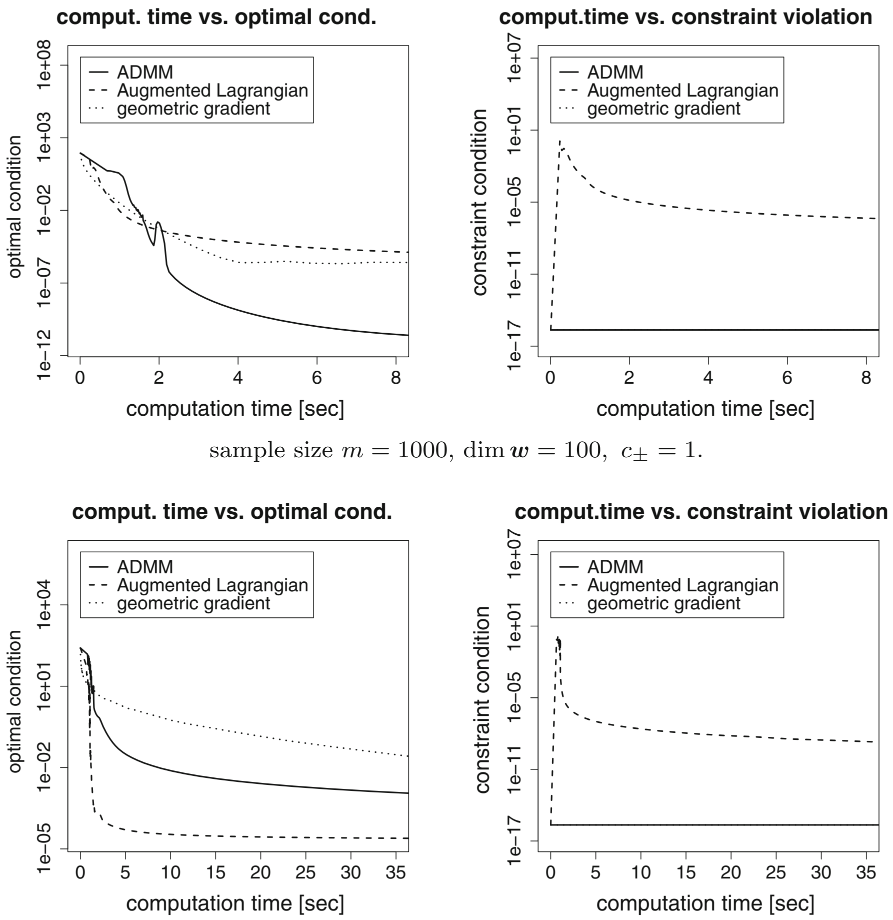
$$\begin{aligned} \|\nabla f(\boldsymbol{w}) - \boldsymbol{w}\boldsymbol{w}^T \nabla f(\boldsymbol{w})\|_1 &= 0, \\ |\boldsymbol{w}^T \boldsymbol{w} - 1| &= 0, \end{aligned} \tag{13}$$

where $\|\boldsymbol{a}\|_1$ is the L_1-norm of the vector $\boldsymbol{a}$, i.e., $\|\boldsymbol{a}\|_1 = \sum_i |a_i|$. For the sequence $\{\boldsymbol{w}_t\}_{t=0}^\infty$ generated by the optimization algorithm, it is expected that $\|\nabla f(\boldsymbol{w}_t) - \boldsymbol{w}_t \boldsymbol{w}_t^T \nabla f(\boldsymbol{w}_t)\|_1$ and $|\boldsymbol{w}_t^T \boldsymbol{w}_t - 1|$ converge to zero.

To improve the convergence properties of AL and ADMM, a parameter α depending on the number of iterations t was used; i.e., $\alpha = \alpha_t$ for the t-th iteration in the algorithms. In preliminary experiments using small problems, $\alpha_t = 1/\sqrt{t}$, $1/t$ and $\alpha_t = 1$ were examined and $\alpha_t = 1/t$ was chosen for both algorithms.

We implemented the algorithms with R language (R Development Core Team 2012) and ran them on the same computer that was used in experiments described in Sect. 3.2. The average numerical results over 40 runs are shown in Figs. 3 and 4. For ADMM and geometric gradient method, the error of the constraint condition was almost equal to zero. For AL, the constraint condition was not satisfied with a sufficient accuracy. For $c_\pm = 1$, the solution of ADMM satisfied the optimality condition (13) with high accuracy. For $c_\pm = 3$, AL provided a better solution in the sense of the error of the optimality condition, but it did not exactly satisfy the constraint $\boldsymbol{w}^T \boldsymbol{w} - 1 = 0$. The solution of ADMM satisfied the optimality condition with higher accuracy than that of the geometric gradient method, while their numerical accuracies of the constraint condition were almost the same. Hence, in this experiment, ADMM was superior to the geometric gradient method.

In the numerical experiments of this section, the results of the optimization methods using the Lagrangian function are better than those of the geometric gradient method. The superiority of AL or ADMM depends on the problem setup. ADMM is a good choice for the problems with a small $c_\pm$; i.e., the problem is essentially convex. On



sample size $m = 1000$, $\dim \boldsymbol{w} = 100$, $c_{\pm} = 1$.

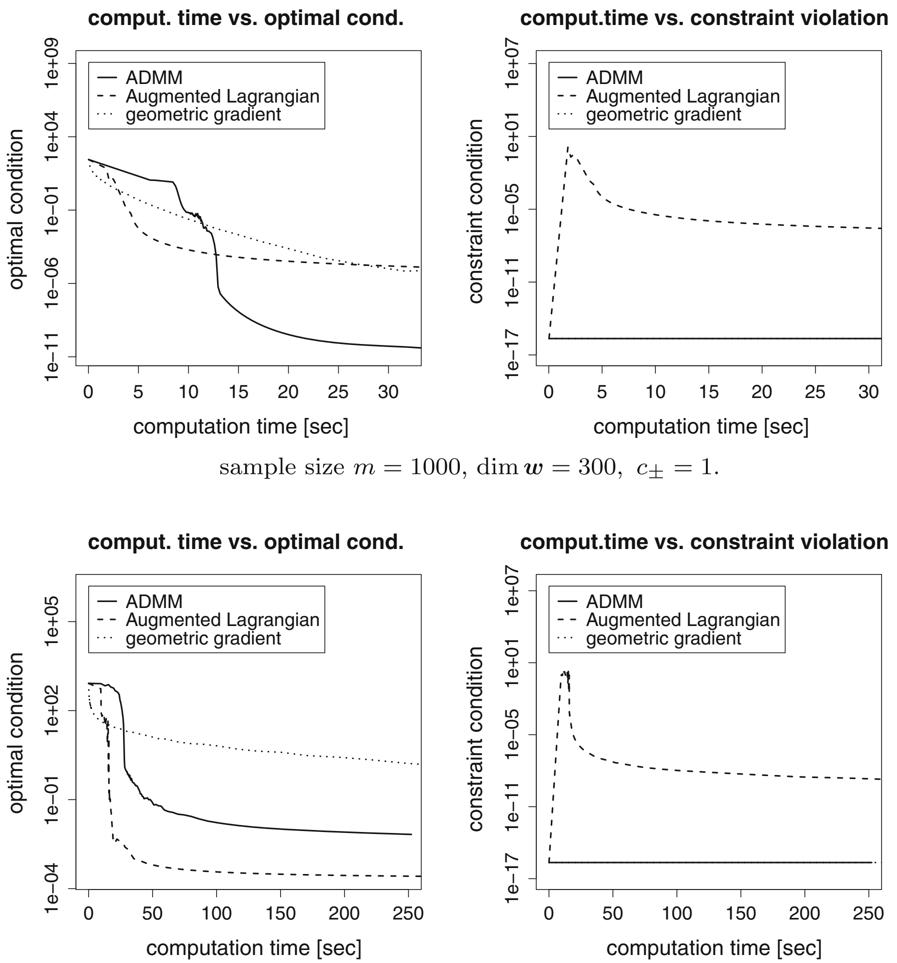
sample size $m = 1000$, $\dim \boldsymbol{w} = 100$, $c_{\pm} = 3$.

Fig. 3 Robust classification models for binary classification: results of numerical experiments are depicted. The sample size is $m = 1{,}000$, and the dimension of the parameter is $\dim \boldsymbol{w} = 100$. The size of the ellipsoidal uncertainty set is $c_{\pm} = 1$ (*upper panels*) or $c_{\pm} = 3$ (*lower panels*). In the *left panel*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the optimality condition, $\|\nabla f(\boldsymbol{w}_t) - \boldsymbol{w}_t \boldsymbol{w}_t^T \nabla f(\boldsymbol{w}_t)\|_1$. In the *right panel*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the constraint condition, $|\boldsymbol{w}_t^T \boldsymbol{w}_t - 1|$. For both ADMM and the geometric gradient method, the error of the constraint condition is close to the machine epsilon, i.e., 10^{-16}

the other hand, AL works better for the problems with a large $c_{\pm}$, i.e., the case that the feasible region of (11) is not reduced to the convex set.

5 Density ratio estimation with dimensionality reduction

Here, the numerical optimization methods presented in Sect. 2 are used to the dimensionality reduction in the density ratio estimation.



sample size $m = 1000$, $\dim \boldsymbol{w} = 300$, $c_{\pm} = 1$.

sample size $m = 1000$, $\dim \boldsymbol{w} = 300$, $c_{\pm} = 3$.

Fig. 4 Robust classification models for binary classification: results of numerical experiments are depicted. The sample size is $m = 1{,}000$, and the dimension of the parameter is $\dim \boldsymbol{w} = 300$. The size of the ellipsoidal uncertainty set is $c_{\pm} = 1$ (*upper panels*) or $c_{\pm} = 3$ (*lower panels*). In the *left panel*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the optimality condition, $\|\nabla f(\boldsymbol{w}_t) - \boldsymbol{w}_t \boldsymbol{w}_t^T \nabla f(\boldsymbol{w}_t)\|_1$. In the *right panel*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the constraint condition, $|\boldsymbol{w}_t^T \boldsymbol{w}_t - 1|$. For both ADMM and the geometric gradient method, the error of the constraint condition is close to the machine epsilon, i.e., 10^{-16}

5.1 Problem setup

Suppose that we are given independent and identically distributed (i.i.d.) samples $\boldsymbol{x}_i, i = 1, \ldots, m$ from a probability density $p_0(\boldsymbol{x})$ on $\mathcal{X} \subset \mathbb{R}^n$ and another set of i.i.d. samples $\boldsymbol{x}'_j$, $j = 1, \ldots, m'$ from a probability density $p_1(\boldsymbol{x})$ on $\mathcal{X}$. The problem is to estimate the density ratio,

$$r(\boldsymbol{x}) = \frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}$$

from the training samples $\boldsymbol{x}_i, i = 1, \dots, m$ and $\boldsymbol{x}'_j, j = 1, \dots, m'$. The density ratio is used in a great variety of statistical problems, including regression problems under covariate shifts, statistical tests, and independent component analysis (see Sugiyama et al. (2012) for details).

Suppose that the difference between p_0 and p_1 is concentrated in a low-dimensional subspace. Then, there exists an n by p orthonormal matrix W and a function $\bar{r} : \mathbb{R}^p \to \mathbb{R}$ such that

$$r(\boldsymbol{x}) = \bar{r}(W^T \boldsymbol{x})$$

holds. This assumption stems from the homogeneity of the conditional probability of $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$, as follows. Suppose that $W \in \mathcal{S}_{n,p}$ and $U \in \mathcal{S}_{n,n-p}$ and that $W^T U = O$ holds. Then, (W, U) is an orthogonal matrix. The probability density can be decomposed into the conditional probability and the marginal probability. Hence, we have

$$\begin{aligned} p_0(\boldsymbol{x}) &= p_0(U^T \boldsymbol{x} | W^T \boldsymbol{x}) q_0(W^T \boldsymbol{x}), \\ p_1(\boldsymbol{x}) &= p_1(U^T \boldsymbol{x} | W^T \boldsymbol{x}) q_1(W^T \boldsymbol{x}). \end{aligned}$$

Let us assume that the conditional probability densities above are the same, i.e.,

$$p_0(U^T \boldsymbol{x} | W^T \boldsymbol{x}) = p_1(U^T \boldsymbol{x} | W^T \boldsymbol{x}).$$

Accordingly, the density ratio $r(\boldsymbol{x})$ can be described as

$$r(\boldsymbol{x}) = \frac{q_0(W^T \boldsymbol{x})}{q_1(W^T \boldsymbol{x})}.$$

The right-hand side is expressed as $\bar{r}(W^T \boldsymbol{x})$, which is regarded as a function on the p-dimensional subspace of $\mathbb{R}^n$.

Following Sugiyama et al. (2011), we introduce an estimator of the function $\bar{r}$ and the matrix $W \in \mathcal{S}_{n,p}$. In particular, a kernel-based density-ratio estimator (Kanamori et al. 2012) is used. Let us define the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2\}$, $\gamma > 0$ and assume that on $\mathcal{X}$, the density ratio $r(z) = \bar{r}(W^T z)$ can be approximated by a linear combination of the Gaussian kernel functions,

$$\sum_{i=1}^{m'} \alpha_i k(W^T z, W^T \boldsymbol{x}'_i) + \sum_{j=1}^{m} \beta_j k(W^T z, W^T \boldsymbol{x}_j).$$

For a fixed $W \in \mathcal{S}_{n,p}$, the coefficient vectors $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{m'})^T \in \mathbb{R}^{m'}$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^T \in \mathbb{R}^m$ can be estimated by using a least squares estimator. We use the explicit form of the estimated coefficients proposed by Kanamori et al. (2012). Let us define an m' by m' gram matrix K_{11} whose components are $(K_{11})_{ij} = k(W^T\boldsymbol{x}_i', W^T\boldsymbol{x}_j')$, and m' by m gram matrix K_{10} whose components are $(K_{10})_{ij} = k(W^T\boldsymbol{x}_i', W^T\boldsymbol{x}_j)$. The vector $\mathbf{1}_m$ stands for $(1,\ldots,1)^T \in \mathbb{R}^m$. The estimators of the coefficient vectors, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, are written as

$$\boldsymbol{\alpha} = -\frac{1}{m\lambda}(K_{11} + m'\lambda \boldsymbol{I}_{m'})^{-1}K_{10}\mathbf{1}_m, \qquad \boldsymbol{\beta} = \frac{1}{m\lambda}\mathbf{1}_m, \tag{14}$$

where λ is a positive number used to avoid overfitting the training samples. Kanamori et al. (2012) proved that the kernel-based density ratio estimator with $\lambda = 1/\min\{m, m'\}^\delta,\ 0 < \delta < 1$ is statistically consistent.

The density ratio is estimated by using the function $\widehat{r}$:

$$\widehat{r}(\boldsymbol{u}; W) = \sum_{i=1}^{m'} \widehat{\alpha}_i k(\boldsymbol{u}, W^T\boldsymbol{x}_i') + \frac{1}{m\lambda}\sum_{j=1}^{m} k(\boldsymbol{u}, W^T\boldsymbol{x}_j), \quad \boldsymbol{u} \in \mathbb{R}^p,$$

where $\widehat{\alpha}_i, i = 1,\ldots,m'$ are the estimated coefficients defined in (14). Note that the $\widehat{\alpha}_i$ depends on W via the gram matrices. Now we are in a position to solve the optimization problem,

$$\min_W -\frac{1}{m}\sum_{i=1}^{m}\widehat{r}(W^T\boldsymbol{x}_i; W) \quad \text{subject to} \quad W \in \mathcal{S}_{n,p}. \tag{15}$$

The optimal solution $\widehat{W}$ provides a potentially accurate estimator of W. Solving problem (15) is equivalent to finding the density ratio that is far from the constant function. A detailed exposition of the statistical properties of the above estimator can be found in Sugiyama et al. (2011). The computational cost of evaluating the function value in (15) is high, and therefore, an efficient optimization algorithm to solve (15) is needed.

5.2 Experiments

The optimization methods presented in Sect. 2 are used to the dimensionality reduction for the density ratio estimation. The problem setup is as follows.

Define matrices $W \in \mathcal{S}_{n,p}$ and $U \in \mathbb{R}^{n\times(n-p)}$ such that $(U, W) \in \mathbb{R}^{n\times n}$ is an orthogonal matrix. Assume that the probability density $p_0(\boldsymbol{x})$ is

$$p_0(\boldsymbol{x}) = q(U^T\boldsymbol{x})q_0(W^T\boldsymbol{x}),$$

where $q(\boldsymbol{u}),\ \boldsymbol{u} \in \mathbb{R}^{n-p}$ and $q_0(\boldsymbol{w}),\ \boldsymbol{w} \in \mathbb{R}^p$ are probability densities of a multivariate standard normal distribution. For $p_1(\boldsymbol{x})$, assume that

$$p_1(\boldsymbol{x}) = q(U^T\boldsymbol{x})q_1(W^T\boldsymbol{x}),$$

where $q_1(\boldsymbol{w})$, $\boldsymbol{w} \in \mathbb{R}^p$ is a p-dimensional normal distribution with a unit mean vector $\boldsymbol{\mu} \in \mathbb{R}^p$ and the variance-covariance matrix $1.5^2 \boldsymbol{I}_p$. The density ratio is $r(\boldsymbol{x}) = \bar{r}(W^T \boldsymbol{x}) = q_0(W^T \boldsymbol{x})/q_1(W^T \boldsymbol{x})$. The sample size is set to $m = m' = 500$.

Let us consider problem (15) defined from the observed data. The experiments focused on the computational cost of the learning algorithms. It is straightforward to see that there is a p by p matrix Λ such that the optimality condition $\nabla_W f(W) = W\Lambda$ holds if and only if $\nabla_W f(W) - WW^T \nabla_W f(W) = \boldsymbol{O}$ is satisfied. Thus, the optimality condition of problem (15) is

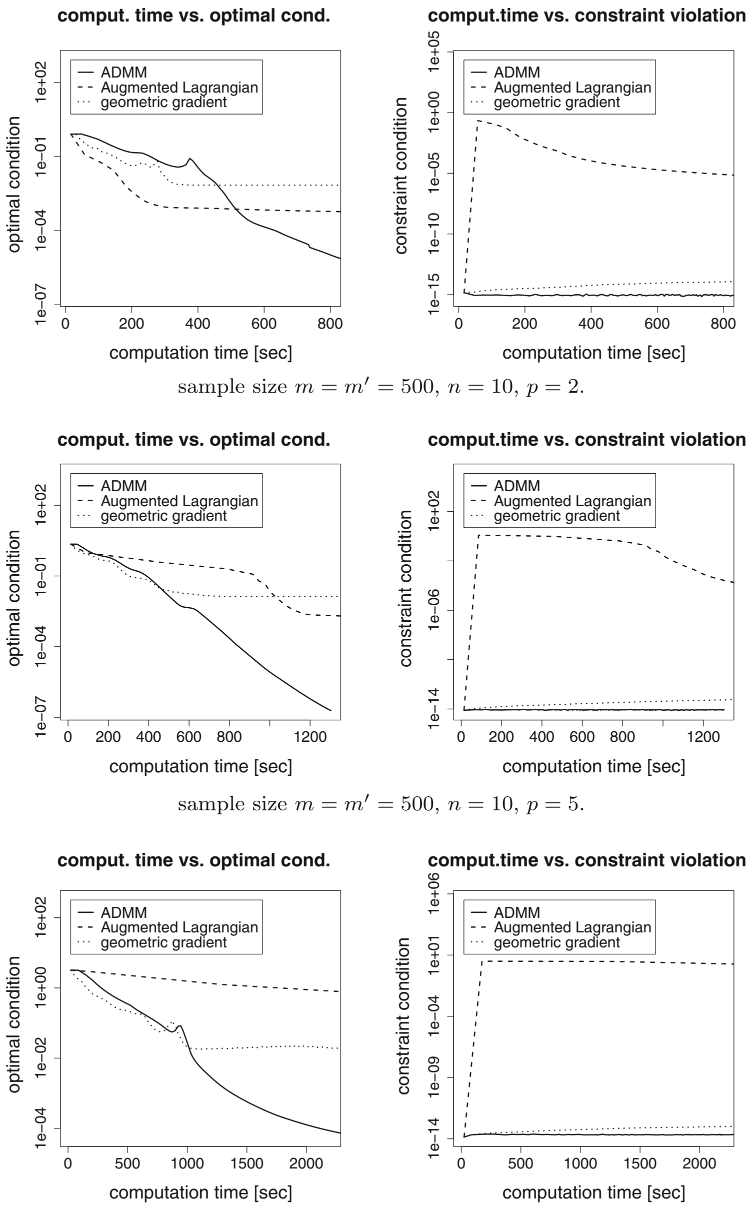
$$\begin{aligned} \|\nabla_W f(W) - WW^T \nabla_W f(W)\|_1 &= 0, \\ \|W^T W - \boldsymbol{I}_p\|_1 &= 0, \end{aligned}$$

where $\|A\|_1$ is the L_1-norm of the matrix A, i.e., $\|A\|_1 = \sum_{i,j} |A_{ij}|$. For the sequence $\{W_t\}_{t=0}^{\infty}$ generated by the optimization algorithm, it is expected that $\|\nabla_W f(W_t) - W_t W_t^T \nabla_W f(W_t)\|_1$ and $\|W_t^T W_t - \boldsymbol{I}_p\|_1$ converge to 0.

We implemented the algorithms with R language (R Development Core Team 2012) and ran them on the same computer used in the experiments described in Sect. 3.2. The numerical experiments were repeated 30 times. In (4) and (6), a parameter α depending on the number of iterations t was used, i.e., $\alpha = \alpha_t$. Preliminary experiments conducted on small problems indicated $\alpha_t = 1$ to be a good choice for ADMM and $\alpha_t = 1/t$ to be a good choice for AL.

The geometric gradient method and subspace rotation method were mainly proposed to solve (15) in Sugiyama et al. (2011). The subspace rotation method is a variant of the gradient method in which the variable W is represented by using a skew symmetric matrix $M \in \mathbb{R}^{n \times n}$ and the matrix M is optimized. The subspace rotation approach was recommended in Sugiyama et al. (2011), since it may be computationally more efficient than the geometric gradient algorithms for large p. In our preliminary experiments, however, the computational cost of the subspace rotation method was almost the same as that of the geometric gradient method. This result is not surprising, since basically the subspace rotation method is also a gradient descant method in another coordinate system. Although an efficient implementation of the subspace rotation method may be possible, there was no concrete description of the algorithm in Sugiyama et al. (2011). Hence, we show only the numerical results of the geometric gradient method.

Figures 5 and 6 show good and stable performance of ADMM. ADMM eventually achieved the smallest error of the optimality condition among the three algorithms. As for AL, the error of the constraint condition hardly becomes smaller, whereas the error of the optimality condition smoothly becomes smaller in the early stage of the optimization for the setup of $p = 2$ (upper panels). AL with a different α_t may speed up convergence, but the computational cost needed to find a good α_t sequence will be high. In contrast, ADMM with $\alpha_t = 1/t$ performed well in all experiments. As for the geometric gradient method, it has similar levels of error for the constraint condition with ADMM, but the performance with respect to the optimality condition is not stable; it achieved relatively good performance for the case of $p = 8$ but it did

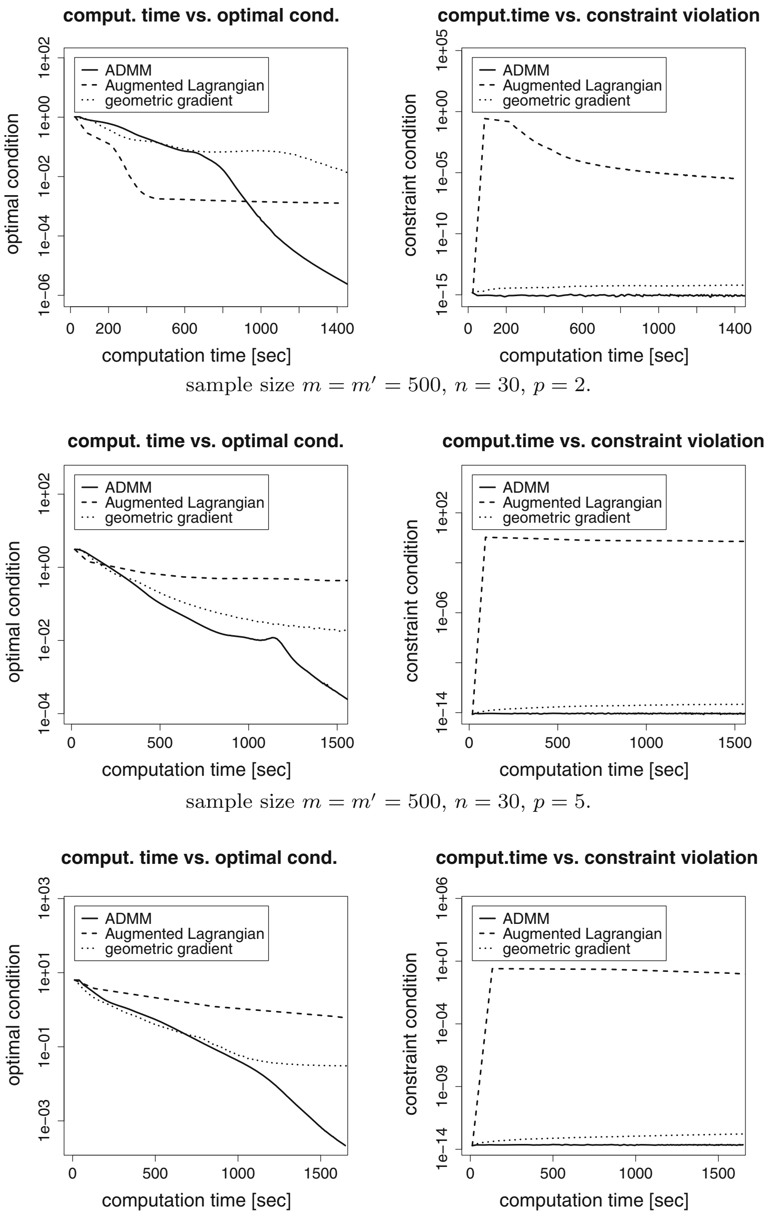


sample size $m = m' = 500$, $n = 10$, $p = 2$.

sample size $m = m' = 500$, $n = 10$, $p = 5$.

sample size $m = m' = 500$, $n = 10$, $p = 8$.

Fig. 5 Density ratio estimation with dimensionality reduction: results of numerical experiments are depicted. The sample size is $m = m' = 500$, and the size of the matrix is $n = 10, p = 2$ (*upper panels*), $n = 10$, $p = 5$ (*middle panels*) or $n = 10$, $p = 8$ (*lower panels*). In the *left panels*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the optimality condition, $\|\nabla f(W_t) - W_t W_t^T \nabla f(W_t)\|_1$. In the *right panels*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the constraint condition, $\|W_t^T W_t - \boldsymbol{I}_p\|_1$. For AL in $n = 10$, $p = 8$, the error of the optimality condition only starts to decrease after the computation has run for more than 6,000 s



sample size $m = m' = 500,\ n = 30,\ p = 2.$

sample size $m = m' = 500,\ n = 30,\ p = 5.$

sample size $m = m' = 500,\ n = 30,\ p = 8.$

Fig. 6 Density ratio estimation with dimensionality reduction: Results of numerical experiments are depicted. The sample size is $m = m' = 500$, and the size of the matrix is $n = 10, p = 2$ (*upper panels*), $n = 10$, $p = 5$ (*middle panels*) or $n = 10$, $p = 8$ (*lower panels*). In the *left panels*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error of the optimality condition, $\|\nabla f(W_t) - W_t W_t^T \nabla f(W_t)\|_1$. In the *right panels*, the *horizontal axis* is the computation time in seconds, and the *vertical axis* is the error the constraint condition, $\|W_t^T W_t - \boldsymbol{I}_p\|_1$. For AL in $n = 30$, $p = 8$, the error of the optimality condition starts to decrease after the computation has run for more than 2,000 s

not for $p = 2$. Therefore ADMM generally achieved good and stable performance in the experiment.

6 Conclusions

We used the geometric gradient descent method and Lagrangian-based methods to solve optimization problems over the Stiefel manifold $\mathcal{S}_{n,p}$.

The numerical results are summarized as follows.

1. The condition number analysis suggested that ADMM is more efficient than AL in the cold-start setup. The validity of this analysis was illustrated by conducting numerical experiments on eigenvalue problems. The inefficiency of AL is compensated by using the hot-start strategy.
2. ADMM, AL, and the geometric gradient descent method were compared for eigenvalue problems with different matrix sizes. When n is almost the same as p, the geometric gradient descent outperforms other methods. When n is much larger than p, ADMM and AL are superior to geometric gradient descent method. This is because the computational cost of the geodesic (2) becomes high in such a case. In addition, AL for the problem with large p did not work well.
3. At solving robust classification problems and density ratio estimation with the dimensionality reduction, ADMM is generally better than the geometric gradient descent method. The convergence of the geometric gradient method is not necessarily as fast as AL or ADMM. The superiority of AL or ADMM depends on the problem setup. We found that AL converged extremely slowly when p is large. AL with a good choice of α_t converges quickly, but it is rather sensitive to the choice of α_t. In addition, the constraint condition, $W^T W = \boldsymbol{I}_p$, is not exactly satisfied in each step of AL. On the other hand, the convergence property of ADMM is good. ADMM did not fail for any problem presented in this paper.

We conclude that ADMM is a promising method for optimization on the Stiefel manifold $\mathcal{S}_{n,p}$. In many applications such as dimensionality reduction, n is much larger than p. In such a situation, geometric gradient descent will be inferior to optimization methods using the Lagrangian function. For the problems with large p, AL tends to converge extremely slowly. On the other hand, ADMM produced fairly good results for all problems in the numerical experiments. AL and ADMM both need good choices of the sequence α_t. It would thus be worthwhile to develop a simple way to determine α_t in practical problems.

Many problems in statistics and machine learning can be formulated as optimizations on the Stiefel manifold, for example, principal component analysis, independent component analysis, and dimensionality reduction. We often need to introduce additional constraints besides the orthonormality. For example, sparse principal component analysis seeks a sparse representation of the principal component (Zou et al. 2006) and imposes an L_1 constraint such as $\|W\|_1 \leq c$. ADMM can deal with some of such non-differentiable constraints (Boyd et al. 2011). It is important to develop optimization algorithms for problems over the Stiefel manifold including non-differentiable objective functions and constraints.

Acknowledgments The authors are grateful to the anonymous reviewers for their helpful comments. Kanamori is partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 24500340. Takeda is supported by JSPS KAKENHI Grant Number 23710174 and by JST, CREST.

References

Absil P-A, Mahony R, Sepulchre R (2008) Optimization algorithms on matrix manifolds. Princeton University Press, Princeton

An LTH, Minh LH, Phuc NT, Tao PD (2008) Noisy image segmentation by a robust clustering algorithm based on dc programming and dca. In: Proceedings of the 8th industrial conference on advances in data mining: medical applications, E-commerce, marketing, and theoretical aspects. pp 72–86

Ben-Tal A, El-Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton University Press, Princeton

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–124

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, New York

Caramanis C, Mannor S, Xu H (2011) Robust optimization in machine learning. In: Nowozin S, Sra S, Wright S (eds) Optimization for machine learning. MIT press, Cambridge

Demmel JW (1997) Applied numerical linear algebra. Soc Ind Appl Math

Edelman A, Arias TA, Smith ST (1998) The geometry of algorithms with orthogonality constraints. SIAM J Matrix Anal Appl 20(2):303–353

Hastie T, Tibishirani R, Friedman J (2001) The elements of statistical learning. Springer, New York

Kanamori T, Suzuki T, Sugiyama M (2012) Statistical analysis of kernel-based least-squares density-ratio estimation. Mach Learn 86(3):335–367

Krause A, Guestrin C (2008) Beyond convexity: submodularity in machine learning. http://www.select.cs.cmu.edu/tutorials/icml08submodularity.html

Luenberger D, Ye Y (2008) Linear and nonlinear programming. Springer, Berlin

Nishimori Y, Akaho S (2005) Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. Neurocomputing 67:106–135

Perez-Cruz F, Weston J, Hermann DJL, Schölkopf B (2003) Extension of the ν-SVM range for classification. In: Suykens JAK, Horvath G, Basu S, Micchelli C, Vandewalle J (eds) Advances inlearning theory: methods, models and applications, vol 190. IOS Press, Amsterdam, pp 179–196

R Development Core Team (2012) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. ISBN 3-900051-07-0

Schölkopf B, Smola AJ (2002) Learning with Kernels. MIT Press, Cambridge

Sra S, Nowozin S, Wright SJ (eds) (2011) Optimization for machine learning. MIT Press, Cambridge

Sugiyama M, Suzuki T, Kanamori T (2012) Density ratio estimation in machine learning. Cambridge University Press, Cambridge

Sugiyama M, Yamada M, von Bünau P, Suzuki T, Kanamori T, Kawanabe M (2011) Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. Neural Netw 24(2):183–198

Takeda A, Mitsugi H, Kanamori T (2012) A unified robust classification model. In: Proceedings of 29th international conference on machine learning (ICML2012). (in press)

Xanthopoulos P, Pardalos PM, Trafalis TB (2012) Robust Data Mining. In: Springer briefs in optimization. Springer, Berlin

Zhang Y (2010) Recent advances in alternating direction methods: theory and practice. IPAM workshop: numerical methods for continuous optimization. UCLA. Los Angeles, California, Oct 2010

Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comput Graph Stat 15:265–286