Robust Subspace Computation Using L1 Norm

Qifa Ke and Takeo Kanade

August 2003

CMU-CS-03-172 3

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Carnegie Mellon University or the U.S. Government.

Keywords: Subspace, Robust, L1 norm, L2 norm, Weighted median, Quadratic programming, SVD

Abstract

Linear subspace has many important applications in computer vision, such as structure from motion, motion estimation, layer extraction, object recognition, and object tracking. Singular Value Decomposition (SVD) algorithm is a standard technique to compute the subspace from the input data. The SVD algorithm, however, is sensitive to outliers as it uses L2 norm metric, and it can not handle missing data either. In this paper, we propose using L1 norm metric to compute the subspace. We show that it is robust to outliers and can handle missing data. We present two algorithms to optimize the L1 norm metric: the weighted median algorithm and the quadratic programming algorithm.

1 Introduction

The measurements or observation data often lie only in a lower dimensional subspace in the original high dimensional data space. Such subspace, especially the linear subspace, has many important applications in computer vision, such as Structure from Motion (SFM) [17], motion estimation [8], layer extraction [9, 10], object recognition [19], and object tracking [1].

To compute the subspace, a *measurement matrix* W is first constructed, which is then factorized to compute the subspace. To construct W , each data item is first reshaped into a column vector m*. All of the reshaped column vectors are then stacked together to form the *measurement matrix* W = [mi, m$_2$, • • • , m_K}. To compute the subspace, we need to factorize this measurement matrix W into U and V :

$$\mathbf{W}_{D\times K} = \mathbf{U}_{D\times d}\mathbf{V}^{\mathbf{T}}_{d\times K} \qquad (1)$$

Here *D* is the dimension of the input data space; *K* is the number of input data items; and *d* is the dimension of the linear subspace. The *d* columns of the matrix U are the bases of the linear subspace that we want to compute.

The input data will contain noises in real cases. Depending on the distribution of the noises, the maximum likelihood estimation (MLE) of the subspace (U and V) is equivalent to minimize some reconstruction error function. For example, if the noise distribution can be modelled by Gaussian distribution, then the MLE is equivalent to minimize the following cost function:

$$E(U,V) = ||W-UV^T||_2 \qquad (2)$$

where || • ||$_2$ is the matrix Frobenious norm (L2 norm).

It is well known that the *Ul* norm is sensitive to the outliers in the input data. In this paper, we will first formulate the subspace computation problem as a probabilistic estimation problem. Then we will present several cost functions according to different assumptions on noise model. We will show that the cost function using *LI* norm metric is not only robust to outliers, but also computationally attractable.

2 Probabilistic view of subspace computation

It was shown in [15, 16] that principal subspace can be computed by maximum likelihood estimation, which in turn can be computed by EM algorith [4]. In a similar way, we formulate the subspace computation as a maximum likelihood estimation problem under different noise model. We will show that maximizing the likelihood is equivalent to minimization some cost function. The format of the cost function is determined by the distribution of the noise in the data.

In general, the observed datum (local measurement) m^ is a D-dimensional column vector contaminated by additive noise:

$$m^* = fji_t + ei \qquad i = 1, \ldots ,K \qquad (3)$$

where $\boldsymbol{\mu}_i$ is the unobservable (fixed but unknown) true value corresponding to the observed (measured) $\mathbf{m}_i$, and ε_i is the additive noise. We know that $\boldsymbol{\mu}_i$ resides in a d dimensional linear subspace ($d < PD$) such that:

$$\boldsymbol{\mu}_i = \mathtt{U}\mathbf{v}_i \tag{4}$$

where $\mathbf{v}_i$ is the projection of $\mathbf{m}_i$ on the subspace defined by the columns of $\mathtt{U}$.

Assuming that local measurements are independent, the log likelihood of the total K measurements is:

$$l(\boldsymbol{\mu}; \mathbf{m}) = \log p(\mathbf{m}_1, ..., \mathbf{m}_K \mid \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) = \sum_{i=1}^{K} \log p(\mathbf{m}_i \mid \boldsymbol{\mu}_i) \tag{5}$$

Therefore, the goal of subspace computation from measurement data is to find the true values $\boldsymbol{\mu}_i$'s that maximize the likelihood of the measurements $l(\boldsymbol{\mu}; \mathbf{m})$, subject to the condition that these $\boldsymbol{\mu}_i$'s reside in a low dimensional subspace defined by $\mathtt{U}$ in Eq. (4)).

2.1 Gaussian noise

If the noise ε_i follows zero-mean normal distribution with common standard deviation of σ, then $\mathbf{m}_i \backsim N(\boldsymbol{\mu}_i, \Sigma)$. By further assuming that the elements of each vector ($\mathbf{m}_i$ or $\boldsymbol{\mu}_i$) are independent, the probabilistic distribution of $\mathbf{m}_i$ conditioned on $\boldsymbol{\mu}_i$ is:

$$p(\mathbf{m}_i \mid \boldsymbol{\mu}_i) \backsim \exp\{-\frac{||\mathbf{m}_i - \boldsymbol{\mu}_i||_2^2}{2\sigma^2}\} \tag{6}$$

where $||\mathbf{x}||_2$ is the $L2$ norm of vector $\mathbf{x}$:

$$||\mathbf{x}||_2 = \left(\sum_i x_i^2\right)^{1/2} \tag{7}$$

The data log likelihood can be written as:

$$l(\boldsymbol{\mu}; \mathbf{m}) = -c\sum_{i=1}^{K} ||\mathbf{m}_i - \boldsymbol{\mu}_i||_2^2 \tag{8}$$

where c is some positive constant. Maximizing the data log likelihood is therefore equivalent to minimizing the term in the r.h.s. of Eq. (8), which is called the cost function or energy function:

$$E_G(\boldsymbol{\mu}) = \sum_{i=1}^{K} ||\mathbf{m}_i - \boldsymbol{\mu}_i||_2^2 \tag{9}$$

Substituting Eq. (4) into Eq. (9) and rewriting Eq. (9) in matrix format, we have:

$$E(\mathtt{U}, \mathtt{V}) = ||\mathtt{W} - \mathtt{U}\mathtt{V}^{\mathsf{T}}||_2^2 \tag{10}$$

Here $\mathtt{W}$ is the measurement matrix whose i-th column is $\mathbf{m}_i$. $\mathtt{V}^\top$ is the projection matrix, with its i-th column being the projection value of the i-th data item in the subspace defined by $\mathtt{U}$.

The assumption of identical and independent distributed (i.i.d.) Gaussian noise model transfers the maximum likelihood problem of $\max_{\mu} l(\boldsymbol{\mu}; \mathbf{m})$ into a minimization problem of a $L2$-norm cost function which is convex in $\mathtt{U}$ and $\mathtt{V}$. The SVD algorithm is a closed form solution to compute its global minimum.

2.2 Laplacian noise

If we assume the noise ε follows Laplacian distribution instead of normal distribution, we have:

$$p(\mathbf{m}_1, \cdots, \mathbf{m}_K \,|\, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) \sim \exp\{-\frac{\sum_{i=1}^K \|\mathbf{m}_i - \boldsymbol{\mu}_i\|_1}{s}\} \tag{11}$$

where $\|\mathbf{x}\|_1$ is the $L1$ norm of vector $\mathbf{x}$:

$$\|\mathbf{x}\|_1 = \sum_i |x_i| \tag{12}$$

The maximum likelihood of the observed data is given by minimizing the following $L1$ norm cost function:

$$E_L(\boldsymbol{\mu}) = \sum_{i=1}^{K} \|m_i - \boldsymbol{\mu}_i\|_1 \tag{13}$$

Written in matrix form, we have:

$$E_L(\mathtt{U}, \mathtt{V}) = \|\mathtt{W} - \mathtt{U}\mathtt{V}^\top\|_1 \tag{14}$$

where $\mathtt{W}$ is the measurement matrix with m_i its i-th column. Unlike the $L2$ norm cost function, the $L1$ norm cost function is in general non-convex in $\mathtt{U}$ and $\mathtt{V}$.

2.3 General case

In general, when the noise follows the same distribution model but with different model parameters for different data points, the data likelihood is:

$$p(\mathbf{m}_1, \cdots, \mathbf{m}_K \,|\, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) \sim \exp\{-\sum_{i=1}^{K}\sum_{j=1}^{D} \frac{h(m_{ij} - \mu_{ij})}{\sigma_{ij}}\} \tag{15}$$

where $h(\cdot)$ is some distance function, and σ_{ij} is related to the parameter of the noise distribution.

The maximum likelihood of the observed data is given by minimizing the following weighted cost function:

$$E(\boldsymbol{\mu}) = \sum_i \sum_j \frac{h(m_{ij} - \mu_{ij})}{\sigma_{ij}} \tag{16}$$

Notice that each data item is weighed by different component $1/\sigma_{ij}$. If we use the Euclidean distance $d(x) = cx^2$, the above cost function is simplified as a weighted sum:

$$E(\boldsymbol{\mu}) = \sum_i \sum_j s_{ij}(m_{ij} - \mu_{ij})^2 \tag{17}$$

Here $s_{ij} = \frac{c}{\sigma_{ij}}$. Written in matrix format, we have:

$$E(\mathtt{U}, \mathtt{V}) = ||\mathtt{S} \otimes (\mathtt{W} - \mathtt{UV})||_2 \tag{18}$$

where $\otimes$ denotes the component wise multiplication. In the low rank approximation context, the above cost function has been studied in robust PCA in [5], and recently in [13]. Unlike the $L2$ norm cost function, the above weighted cost function is in general non-convex in $\mathtt{U}$ and $\mathtt{V}$, due to the weight matrix $\mathtt{S}$.

In summary, the maximum likelihood (ML) solution to the matrix factorization (subspace computation) depends on the noise distribution assumed. When the noise follows independent and identical Gaussian distribution, the ML solution is obtained by minimizing a $L2$ norm cost function. When the noise follows independent and identical Laplacian distribution, the ML solution is achieved by minimizing a $L1$ norm cost function. In general when the noise distributions are no longer identical, the ML solution comes from minimizing a non-convex weighted cost function [5, 13], with the weights set according to some problem dependent distance function. Both the cases of $L1$ norm and weighted cost function can deal with outliers, as will be shown in the following sections.

For other noise distributions, such as the generalized exponential family, corresponding cost functions can also be derived [2].

3 $L2$-norm based subspace computation

Gaussian distribution is the most often assumed noise model. Under Gaussian noise model, the problem of estimating the subspace is equivalent to minimize the following $L2$-norm cost function:

$$E(\mathtt{U}, \mathtt{V}) = ||\mathtt{W}_{D\times K} - \mathtt{U}_{D\times d}\mathtt{V}^{\mathsf{T}}_{d\times K}||_2^2 \tag{19}$$

where d is the dimension of the subspace defined by $\mathtt{U}$, and $d < D$.

Singular Value Decomposition (SVD) is a popular approach to minimize $E(\mathtt{U}, \mathtt{V})$. The following theory of SVD explains how SVD can be used to minimize $E(\mathtt{U}, \mathtt{V})$ [6]:

Theorem 1. *[6] Let the SVD of matrix* $\mathtt{W}$ *be*

$$\mathtt{W}_{D\times K} = \mathtt{A}_{D\times D}\Sigma_{D\times D}\mathtt{B}^{\mathsf{T}}_{D\times K} \tag{20}$$

where $\Sigma = diag(\lambda_1, \cdots, \lambda_D)$, $\lambda_1 \geq \cdots \geq \lambda_D \geq 0$, *and* $\mathtt{A}$ *and* $\mathtt{B}$ *orthonormal matrix. Then for* $1 \leq d \leq D$, *we have:*

$$\min E(\mathtt{U}, \mathtt{V}) = \sum_{i=d+1}^{D} \lambda_i^2 \tag{21}$$

The above theorem states that the first d columns of $\mathtt{A}$ in Eq. (20) defines the subspace that minimizes the $L2$-norm cost function defined in Eq. (19), i.e.,

$$\mathtt{U} = \mathtt{A}(:, 1 : d)$$

Similarly we have

$$\mathtt{V} = \mathtt{B}(:, 1 : d)$$

SVD gives the closed form solution to the $L2$ norm cost function in Eq. (19). The problem with using the $L2$ norm cost function is that it is sensitive to outliers. With even a single influential outliers, the resulted subspace could be completely different from the desired solution. Detecting such outliers is therefore necessary.

Parametric approaches are often used to deal with outliers. The parametric approaches define a global parametric model that inliers should follow. Outliers are those items that do not follow such parametric model. Specifically, in parametric approaches, a parametric model is first fit to the data, and then outliers are identified as the data that violate the fit model. A more general scheme is to give each data item a weight in the range of $[0, 1]$ according to the degree that such data item violates the global parametric model. A zero weight indicates an outlier. Robust estimator is often used to weight each data item, where the objective function in Eq. (19) is rewritten as:

$$\min \sum_{i,j} \rho(m_{ij} - \mathbf{u}_{i\cdot}^{\mathsf{T}} \mathbf{v}_{\cdot j}) \tag{22}$$

where m_{ij} is the ij-th element of $\mathtt{W}$, $\mathtt{U}$ and $\mathtt{V}$ are the global parametric model (subspace model), $\mathbf{u}_{i\cdot}$ is the i-th row of $\mathtt{U}$, and $\mathbf{v}_{\cdot j}$ is the j-th column of $\mathtt{V}^{\mathsf{T}}$. The contribution to the cost function of each data element is controlled by the robust M-estimator $\rho(\cdot)$ based on the distance between the data element and the current subspace model, i.e., the residual $m_{ij} - \mathbf{u}_{i\cdot}\mathbf{v}_{\cdot j}$. For example the Geman-McClure robust function $\rho(x, \sigma) = \frac{x^2}{x^2+\sigma^2}$ is used in [18], where σ is the parameter that controls convexity of the robust estimator.

The use of robust M-estimator to solve Eq. (22) changes the convexity of the cost function. In general, there are many minimums in Eq. (22), and iterative procedures are often used to derive a good local minimum. In each iteration, each data item is first weighted based on its distance to the current parametric model, and then a new model is recomputed using the weighted data. When the dimension of the data is too high to afford computing the subspace model multiple times, gradient decent can be used to compute a local minima [18].

The convergence of the above iterative process depends on the model initialization. When a reasonably good initialization is available, the parametric method is highly effective since it takes the global data into account in detecting the outliers. Parametric approaches are effective for detecting structure outliers, since such outliers are not influential and a good initial model is possible if there are not extreme outliers. On the other hand, in the presence of influential extreme outliers, it would be hard, before the removal of the extreme outliers, to obtain a good initial model as the starting point for the iterative procedure.

4 $L1$-norm based subspace computation

In this section, we discuss the potential advantages of using $L1$ norm metric for subspace computation. Minimizing the $L1$ norm metric corresponds to the maximum likelihood estimation under Laplacian noise model. We first show that $L1$ norm metric is more robust than $L2$-norm through a simple illustrative line-fitting example. We then present two algorithms to compute the subspace using $L1$ norm metric: Alternating Weighted-Median algorithm and Alternating Convex Quadratic Programming. These two algorithms are efficient: weighted median has fast algorithm [14], and convex quadratic programming (see [11]) is well studied and has very efficient software package available. More extensive experiments of these above two approaches to subspace computation is part of the future work.

4.1 Robust $L1$ norm metric: example

One important advantage of using $L1$ norm is that it is more robust to outliers than $L2$ norm in statistical estimation. This can be seen from the following simple example where we try to find a 1D subspace from given 2D data items. In other words, the example is to fit a line to the given 2D data points.

Suppose we are given 10 two-dimensional points $\{(x_i, y_i) \,|\, i = 1, ..., 10\}$ where the response variable y is corrupted by Gaussian noise. We want to fit a line $y = kx$ to these 10 points, where k is the parameter (slope) that we need to estimate. In other words, we want to compute the one dimensional subspace from the given two dimensional data. Specifically, we use the following linear model:

$$y = kx + \varepsilon \tag{23}$$

where k is the parameter to estimate and ε is the noise that corrupts the response variable y.

4.1.1 L2 norm formulation

If ε is assumed to be Gaussian noise, then the ML estimation of the parameter k is given by minimizing the following $L2$-norm cost function (sum of squared difference):

$$E(k) = \sum_{i=1}^{10} (y_i - kx_i)^2 \tag{24}$$

The least squared solution to minimize the above cost function is:

$$k = \frac{\sum_{i=1}^{10} x_i y_i}{\sum_{i=1}^{10} x_i^2}$$

4.1.2 L1 norm formulation

If ε is assumed to have Laplacian distribution, then the ML estimation of the parameter k is given by minimizing the follow $L1$-norm cost function:

$$\sum_{i=1}^{10} |y_i - kx_i| = \sum_{i=1}^{10} |x_i| |\frac{y_i}{x_i} - k| \tag{25}$$

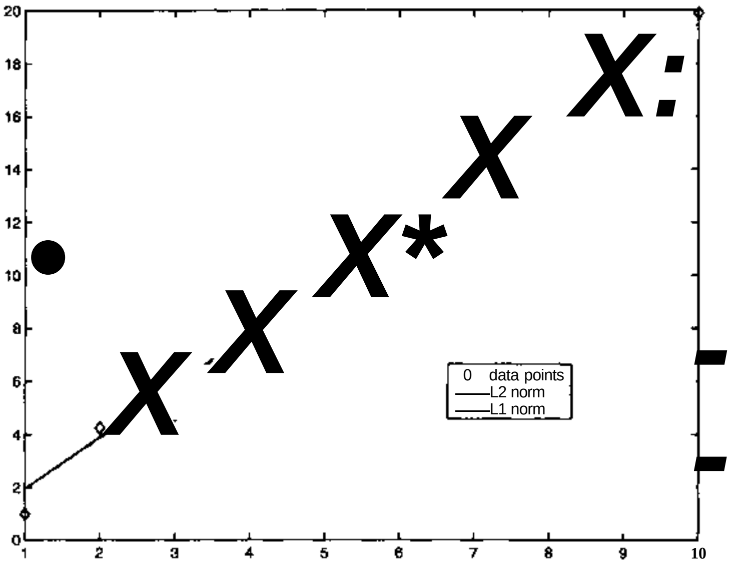


Figure 1: Fit a line to 10 given data points. All data points are inliers. The result of using L2 norm cost function is similar to that of using LI norm.

Global minimum of Eq. (25) can be obtained using the following well known result (see [14]):

Result 1. *The global minimum of the Ll-norm cost function E{k) = Yli$\overset{K}{=}$i \\Vi ~~ ^xi\\i is given by the weighted median of{^\ i = 1,..., K}, where \xi\ is the weight for the i-th item* $\frac{y_i}{x_i}$.

If *Xi* = 0, then its corresponding i-th data point is removed from the accumulation in Eq. (25), since the weight is equal to zero too.

4.1.3 Results

Fig. 1 shows the results when there is NOT any outlier in the given data. As we can see, the LI norm and *Ul* norm cost functions give similar estimation of *k.*

When there are outliers in the data, the results are different. In Fig.2 there are two outliers, *A* and *B.* The LI norm cost function still gives good results, while the *Ul* norm cost function gives erroneous estimation.

4.2 Alternative minimization

We have shown that by assuming Laplacian noise distribution, the maximum likelihood estimation of matrix factorization corresponds to minimizing a Ll-norm cost function, and that Ll-norm metric is more robust to outliers than L2-norm metric. In this section, we present algorithms on how to maximize the likelihood, i.e., minimize the Ll-norm based cost function:

$$E(\mathtt{U},\mathtt{V}) = ||W-UV^T||_1 \qquad (26)$$

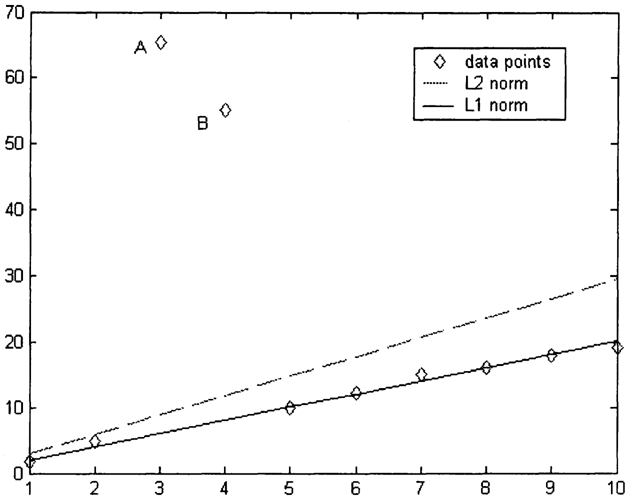


Figure 2: Fit a line to 10 given data points. Two data points A and B are outliers. Using $L2$ norm cost function gives erroneous result shown in dash line, while using $L1$ norm cost function still gives correct result shown in solid line.

$\mathtt{W}_{D\times K}$ is the measurement matrix with Column i the observed (measured) data x_i. The columns of $\mathtt{U}_{D\times d}$ are the d bases of the subspace to be estimated, with $d < \min(D, K)$.

While Eq.(25) has a global minimum that can be computed via the weighted median, the cost function for matrix factorization in Eq.(26) is in general non-convex, since both $\mathtt{U}$ and $\mathtt{V}$ are unknown. It requires some iterative scheme to achieve a good local minimum.

If $\mathtt{U}$ or $\mathtt{V}$ is known, then we can use weighted median to compute the global minimum of Eq.(26). This fact suggests a scheme that minimizes the cost function alternatively over $\mathtt{U}$ or $\mathtt{V}$, each time optimizing one argument while keeping the other one fixed. Such alternative minimization scheme [3] has be widely used in subspace computation using $L2$ norm [12] or other distance metric such as Bregman Divergences [2]. The alternative optimization can be written as:

$$\mathtt{U}^{(t)} = \arg\min_{\mathtt{U}} \|\mathtt{W} - \mathtt{U}\mathtt{V}^{(t-1)^{\mathsf{T}}}\|_1 \tag{27a}$$

$$\mathtt{V}^{(t)} = \arg\min_{\mathtt{V}} \|\mathtt{W} - \mathtt{U}^{(t)}\mathtt{V}^{\mathsf{T}}\|_1 \tag{27b}$$

4.2.1 Alternative minimization by weighted-median

Alternative minimization via weighted-median has been applied to robust SVD (unpublished document [7]). However, the algorithm presented in [7] contains mistakes and can not correctly handle the case where the rank of the measurement matrix is more than one, i.e. $\mathrm{rank}(\mathtt{W}) > 1$. We present the correct algorithm that can handle the case where $\mathrm{rank}(\mathtt{W})$ is more than one.

```
//Initialization
Set $\mathtt{U} = 0$ and $\mathtt{V} = 0$
//Cycle through d columns of $\mathtt{U}$ for N times
For $n = 1, \cdots, N, c = 1, \cdots, d$:
    //Optimize $\mathbf{u}_{\cdot c}$, the c-th column of $\mathtt{U}$ with other columns fixed
    If $n = 1$, initialize $\mathbf{v}_{\cdot c}^{(0)}$ randomly
    Set $\mathtt{W} = \mathtt{W} - \sum_{k \neq c} u_{ik} v_{kj}$
    For $t = 1, \cdots$, convergence
        For $i = 1 \cdots D$,   $u_i^{(t)} = \arg\min_u \|\mathbf{m}_{i\cdot} - u\mathbf{v}_{\cdot c}^{(t-1)^\top}\|_1$
        For $j = 1 \cdots K$,   $v_j^{(t)} = \arg\min_v \|\mathbf{m}_{\cdot j} - v\mathbf{u}_{\cdot c}^{(t)}\|_1$
```

Figure 3: Algorithm of using iterative weighted median to minimize the $L1$ norm cost function, and therefore to compute the subspace.

To simplify the presentation, we first consider the case where the dimension of the subspace is one. The alternating minimization problems are:

$$\text{For } i = 1 \cdots D, \quad u_i^{(t)} = \arg\min_u \|\mathbf{m}_{i\cdot} - u\mathtt{V}^{(t-1)^\top}\|_1 \tag{28a}$$

$$\text{For } j = 1 \cdots K, \quad v_j^{(t)} = \arg\min_v \|\mathbf{m}_{\cdot j} - v\mathtt{U}^{(t)}\|_1 \tag{28b}$$

where $u_i^{(t)}$ is the i-th element of the column vector $\mathtt{U}$ (similar definition of $v_i^{(t)}$) , t is the index of the iteration steps. The solutions (global minimums) to Eq. (28) can be obtained through the well known weighted median algorithm according to Result 1.

When the subspace dimension d is more than one, $\mathtt{U}$ and $\mathtt{V}$ contain more than one column. Our algorithm cycles through the d columns of $\mathtt{U}$ and $\mathtt{V}$, optimizing each column while fixing the others. The problem is therefore broken into d subproblems of Eq. (28). The overall algorithm is shown in Fig 3.

4.2.2 Alternative minimization by convex quadratic programming

We have presented the approach that cycles through the principal vectors (subspace bases) by optimizing over one principal vector while fixing the others. In this subsection, we convert the subspace computation problem to alternative convex optimization problem, which updates all principal vectors at a time in each iteration. The alternative convex optimization is potentially faster and achieve better local minimum than the alternative weighted median approach presented above.

In the following we show how to solve Eq. (27b). Eq. (27a) can be solved similarly. The cost function of Eq. (27b) can be written as:

$$
\begin{aligned}
E(\mathtt{V}) &= \|\mathtt{W} - \mathtt{U}^{(t)}\mathtt{V}^{\mathsf{T}}\|_1 \\
&= \sum_{j=1}^{K} \|\mathbf{m}_{\cdot j} - \mathtt{U}^{(t)}\mathbf{v}_{\cdot j}\|_1
\end{aligned}
$$

where $\mathbf{m}_{\cdot j}$ is the j-th column of $\mathtt{W}$, $\mathbf{v}_{\cdot j}$ is the j-th column of $\mathtt{V}^{\mathsf{T}}$. The problem of Eq. (27b) is therefore decomposed into K sub-problems, each one optimizing $\mathbf{v}_{\cdot j}$. Each sub-problem has the following general formula:

$$
\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{b} - \mathtt{A}\mathbf{x}\|_1 \tag{29}
$$

This problem can be reduced to a simple convex quadratic programming problem whose global minimum can be computed efficiently [11]:

$$
\begin{aligned}
&\min_{\mathbf{x},\mathbf{z},\mathbf{t}} \frac{1}{2}\|\mathbf{z}\|_2^2 + \gamma \mathbf{e}^{\mathsf{T}}\mathbf{t} \\
&s.t. - \mathbf{t} \leq \mathtt{A}\mathbf{x} - \mathbf{b} - \mathbf{z} \leq \mathbf{t}
\end{aligned} \tag{30}
$$

where $\mathbf{e}$ is a column vector of ones. γ is a small positive constant.

4.2.3 Convergence

The cost function $E(\mathtt{U}, \mathtt{V})$ is decrease at each alternative minimization step. Since the cost function $E(\mathtt{U}, \mathtt{V})$ is lower bound (≥ 0), the alternative minimization procedure will converge. By carefully design the algorithm, it will converge to a local minimum. We are investigating if it will converge to a local minimum in theory.

The convergence is achieved when the difference of the parameters between adjacent iterations is small enough. More specifically, the algorithm will stop if for each subspace base, the following holds:

$$
\theta(\mathbf{u}_c^t, \mathbf{u}_c^{t-1}) < \varepsilon \tag{31}
$$

Here $\theta(\mathbf{u}_1, \mathbf{u}_2)$ is the angle between the two vectors $\mathbf{u}_1$ and $\mathbf{u}_2$; $\mathbf{u}_c$ is the c-th subspace base; and ε is a small positive number.

4.3 Handling missing data

Missing data can be handled in both weighted-median algorithm and convex programming algorithm, by discarding the constraints corresponding to the missing data.

To see the reason, we rewrite Eq. (26) as:

$$
E(\mathtt{U}, \mathtt{V}) = \sum_{i=1}^{D}\sum_{j=1}^{K} |w_{ij} - \mathbf{u}_{i\cdot}\mathbf{v}_{\cdot j}| \tag{32}
$$

where *w^* is the element located at i-th row and j-th column of W, u*. is the i-th row of U, and v.j is the j-th column of V^{T}. If *Wij* is missing, then we discard the corresponding cumulative item of *\wij* — u^.v.j|.

For the weighted-median algorithm, discarding such item does not affect the result of the weighted median in Eq (28).

For the quadratic programming algorithm, discarding one such item removes one corresponding equation in the equation set in Eq (29). As long as the total number of missing elements in b (one column in the measurement matrix W) is no more than *D — d,* the equation set is still over-constrained and the quadratic programming is still solvable. In general, the original dimensions *D* is much larger than the subspace dimension d, which allows large number of missing data in each column of W .

4.4 Summary

In summary, the LI norm formulation of subspace computation that requires minimization of |W — UV^{T}||i can be decomposed into two alternative minimization problem. Each alternating problem is further divided into *D* and *K* independent sub-problems. Each sub-problem can be in turn reduced to a simple convex quadratic problem whose global minimum can be computed efficiently. Notice that while the global minimum of each sub-problem can be derived by convex quadratic programming, the original problem min$_{uv,}$ 1W — UV^{T}||i is in general non-convex.

5 Example

Let us consider an 8 x 6 measurement matrix, which consists of eight data points in the six dimensional column space (or 6 data points in the eight dimensional row space), as shown in Eq (33). The rank of this matrix is two, which means these eight data points actually lie in a 2D subspace. Now suppose we observe these eight data points but with outlier measurements. As shown by the red italic elements in Eq (34), every data point contains outlier measurement!

$$\text{\textasciicircum 6x8} = \begin{bmatrix} 9.47 & 8.42 & -12.49 & 1.03 & 1.69 & 3.83 & 1.84 & 8.08 \\ -7.30 & -0.13 & -5.71 & -4.56 & 11.26 & 9.48 & 5.83 & 8.97 \\ -2.43 & -2.03 & 2.88 & -0.34 & -0.17 & -0.72 & -0.32 & -1.75 \\ 8.13 & 6.99 & -10.15 & 1.02 & 0.99 & 2.83 & 1.31 & 6.37 \\ 7.87 & 5.83 & -7.55 & 1.55 & -0.90 & 0.89 & 0.19 & 3.91 \\ 7.56 & 1.50 & 2.62 & 3.92 & -8.97 & -7.16 & -4.48 & -6.03 \end{bmatrix} \quad (33)$$

$$\tilde{\mathtt{W}}_{6\times 8} = \begin{bmatrix} 9.47 & 8.42 & -12.49 & 1.03 & 1.69 & 3.83 & 1.84 & 8.08 \\ -7.30 & -0.13 & \mathit{200.0} & -4.56 & 11.26 & 9.48 & 5.83 & 8.97 \\ -2.43 & \mathit{-100.0} & 2.88 & -0.34 & -0.17 & \mathit{300.0} & \mathit{-300.0} & -1.75 \\ 8.13 & 6.99 & -10.15 & 1.02 & \mathit{-300.0} & 2.83 & 1.31 & \mathit{700.0} \\ 7.87 & 5.83 & -7.55 & \mathit{200.0} & -0.90 & 0.89 & 0.19 & 3.91 \\ \mathit{400.0} & 1.50 & 2.62 & 3.92 & -8.97 & -7.16 & -4.48 & -6.03 \end{bmatrix} \tag{34}$$

$$\hat{\mathtt{W}}_{L1} = \begin{bmatrix} 9.47 & 8.42 & -12.39 & 1.03 & 1.69 & 3.83 & 1.84 & 8.08 \\ -7.30 & -0.13 & -3.41 & -4.56 & 11.26 & 9.48 & 5.83 & 8.97 \\ -2.43 & -2.03 & 2.91 & -0.34 & -0.17 & -0.72 & -0.32 & -1.75 \\ 8.13 & 6.99 & -10.15 & 1.02 & 0.99 & 2.83 & 1.31 & 6.37 \\ 7.87 & 5.83 & -7.91 & 1.55 & -0.90 & 0.89 & 0.19 & 3.91 \\ 7.56 & 1.50 & 0.75 & 3.92 & -8.97 & -7.16 & -4.48 & -6.03 \end{bmatrix} \tag{35}$$

$$\hat{\mathtt{W}}_{L2} = \begin{bmatrix} 0.17 & 0.37 & -0.11 & 0.02 & -2.81 & -0.88 & 0.92 & 6.56 \\ -0.34 & -1.24 & 0.05 & -0.01 & -0.44 & 3.77 & -3.75 & 1.03 \\ -28.70 & -99.70 & 5.28 & -1.67 & 0.46 & 299.22 & -298.29 & -1.08 \\ 9.81 & 7.08 & -10.01 & 2.30 & -299.96 & 2.94 & 1.23 & 699.98 \\ 0.20 & 0.55 & -0.09 & 0.02 & -1.89 & -1.51 & 1.53 & 4.42 \\ 1.92 & 6.56 & -0.38 & 0.11 & -1.20 & -19.62 & 19.57 & 2.80 \end{bmatrix} \tag{36}$$

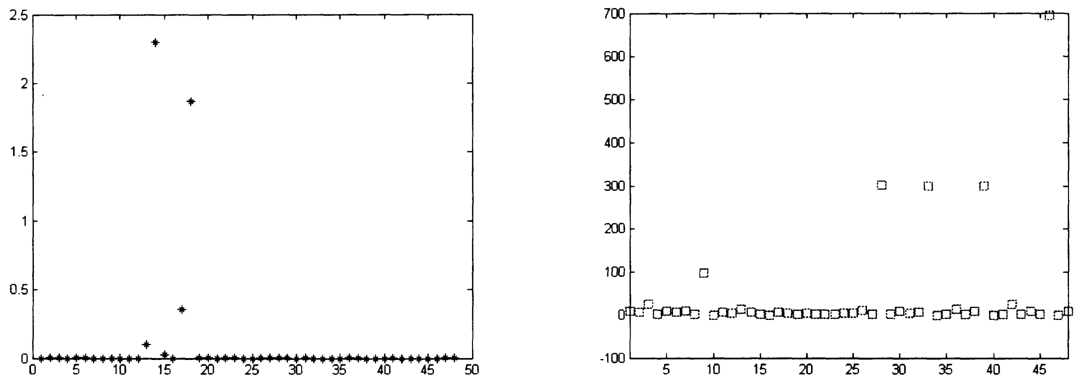
We apply the algorithm of alternative minimization by weighted-median to compute the 2D subspace. Eq (35) shows the reconstructed matrix in the 2D subspace, and Fig. 4(a) shows the reconstruction error. As we can see, the errors are small, and the outlier measurements have been successfully recovered.

We also apply the SVD algorithm ($L2$ norm) to compute the two dimensional subspace. Eq (36) shows the reconstructed matrix in the 2D subspace, and Fig. 4(b) shows the reconstruction error. The SVD algorithm is sensitive to the outlier measurements, as we can see from the erroneous reconstructed matrix.

For comparison purpose, we plot the reconstruction errors in same coordinate frame, as shown in Fig 5. The weighted median algorithm ($L1$ norm) achieves much better results than the SVD algorithm ($L2$ norm).

6 Conclusion

In this paper we study the problem of robust subspace computation. From the probabilistic view point, subspace computation can be formulated as maximum likelihood estimation problem, which in turn leads to the low rank matrix approximation. Under different noise models, subspace computation is formulated as minimizing the matrix reconstruction error using, respectively, $L2$ norm, $L1$ norm, or general weighted reconstruction error function.



(a) L1 norm

(b) L2 norm

Figure 4: Reconstruction error for each element in the measurement matrix W. (a) Black "$*$": weighted median using $L1$ norm metric; (b) Red "□": SVD algorithm using $L2$ norm metric.

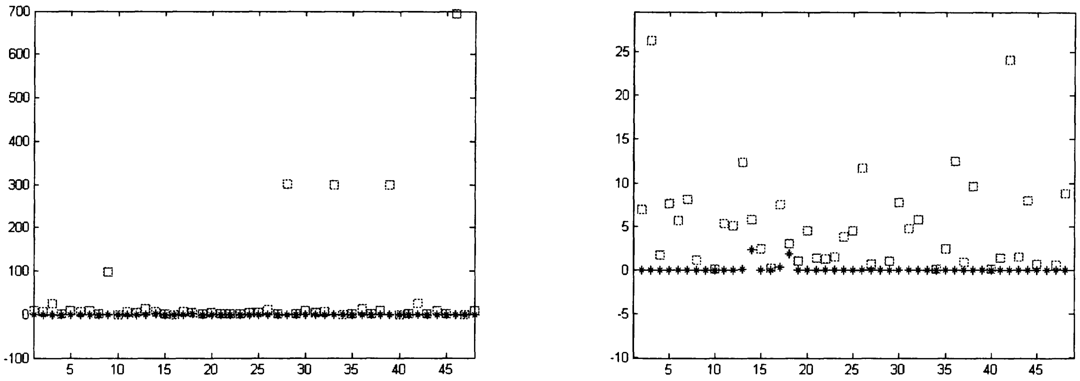
The un-weighted $L2$ norm error function is convex and its global minimum can be computed using SVD algorithm. But it is sensitive to outliers. $L1$ norm error function and general weighted error function are robust to outliers, but they are non-convex. Alternative minimization algorithms can be used to minimize such non-convex function. We study two alternative minimization algorithms to minimize the $L1$ norm error metric, namely the weighted median and quadratic programming. The weighted median algorithm is robust and simple, but it can only compute the subspace bases one by one, and therefore potentially easier to be trapped into a bad local minima. The quadratic programming can compute the subspace bases all at once in each iteration step, and is potentially more efficient since quadratic programming is a well-studied and well-tuned algorithm. Alternative minimization requires a good initialization for the algorithm to converge to a good solution. Currently we use random initialization. In the future, we will study how to initialize the alternative algorithm. Testing the algorithms on real data is also part of future work.

References

[1] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV (1)*, pages 329–342, 1996.

[2] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[3] I. Csiszár and G. TTusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions, Supplement Issue*, 1:205–237, 1984.



(a) errors in common coordinate frame; (b) zoom-in of (a)

Figure 5: Zoom in of reconstruction error.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximal likelihood form incomplete data via the em algorithm. *RoyalStat*, B 39:1–38, 1977.

[5] Ruben Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with choise of weights. *Technometrics*, 21(4):489–498, November 1979.

[6] G.H. Golub and C.F. Van Loan. *Mattrix Computation.* Johns Hopkins University Press, Baltimore, maryland, 2nd edition, 1989.

[7] Douglas M. Hawkinsa, Li Liu, and S. Stanley Youngc. Robust singular value decomposition, *http://www.niss.org/technicalreports/tr122.pdf.*

[8] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV99.*

[9] Qifa Ke and Takeo Kanade. A subspace approach to layer extraction. In *CVPR 2001.*

[10] Qifa Ke and Takeo Kanade. A robust subspace approach to layer extraction. In *IEEE Workshop on Motion and Video Computing (Motion 2002)*, 2002.

[11] O. L. Mangasarian and David R. Musicant. Robust linear and support vector regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):950–955, 2000.

[12] S. Roweis. Em algorithms for pca and spca. In *NIPS*, 1997.

[13] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *ICML 2003 (to appear).*

[14] T.H.Cormen and C.E.Leiserson andR.L.Rivest. *Introduction to Algorithms.* MIT Press, McGraw-Hill, New York, NY, 1990.

[15] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, 1997.

[16] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[17] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2), 1992.

[18] F. Torre and M. J. Black. Robust principal component analysis for computer vision. In *ICCV2001*.

[19] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.