Mathematical Modeling I

Dr. Zachariah Sinkala
Department of Mathematical Sciences
Middle Tennessee State University
Murfreesboro Tennessee 37132, USA

September 1, 2011

- We were able to understand the limitations of exact inversion in the deblurring example because, in the Fourier representation, the forward problem could be written as a set of uncoupled scalar equations and therefore consisted of componentwise multiplication.
- Each Fourier component in the image is multiplied by the corresponding value of the Fourier transform of the point-spread function and then has noise added to give the data.
- The inverse of the forward problem is then componentwise division of the Fourier transform of the data by the appropriate scalar from the Fourier transform of the point-spread function; problems arose when the divisor had small magnitude.
- We would like to have a similar description of other linear inverse problems so that we can similarly predict when problems will arise.

Singular Value Decomposition

- Linear forward problem will not necessarily be shift-invariant and a Fourier representation of image and data space will not lead to the forward mapping being a set of uncoupled scalar equations.
- Usually the size of image and data spaces are different (an extreme case is continuous-discrete problems) in which case the matrix representing the forward mapping is not even square.
- In all these cases there is an analogue of Fourier space given by the singular value decomposition (SVD) in which the forward operator reduces to componentwise scalar multiplication and the straightforward inverse is by scalar division.

Discrete-discrete problems

- The simplest case to understand is when both image and data space are finite-dimensional.
- We will develop the SVD for that case first and then go on to the case where image or data space is continuous.

- Linear algebra forms the mathematical basis for the vector and matrix analysis that we use to analyze linear inverse problems where both image and data space is discrete.
- When viewed from a certain perspective, it has a natural generalization to functional analysis, which is the basic tool in analysis of continuous inverse problems

- M-dimensional vector space is denoted R^M and consists of the set of all ordered $M-$tuples of real numbers, usually written as a column

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{pmatrix}$$

- Addition of vectors is defined by $(u+v)k = uk + vk$ and scalar multiplication by $(cu)k = cuk, c \in R$.

Systems of linear equations

- We now consider the system of equations

$$Af = d$$

  we are asking how we can synthesize the vector d by taking the appropriate linear combination (specified by the solution f) of the columns of the matrix A.
- Traditionally: A system of equations $Af = d$ is said to be over-determined if there is no solution and under-determined if there are infinitely many solutions.

- The input (or image) f is the collection of quantities we wish to reconstruct and the output d are the data we measure.
- In a linear inverse problem, the relationship between f and d is

$$d = Af + n$$

where A is a linear transformation, and n represents an additive noise process which prevents us from knowing the noise-free data $y = Af$ precisely.
- Different applications give rise to image and data spaces of different sizes, but we shall wish to present a unified treatment of all of these.

Anatomy of a linear transformation

- One way to treat the system of equations $d = Af$ where A is rectangular (and hence certainly not invertible) is to consider the operator $A^T A$ (or AA^T) which is square and potentially invertible.
- This idea of analyzing the properties of $A^T A$ (or AA^T) in order to give information about A leads to a very important way of characterizing the behavior of any finite dimensional linear transformation called the singular value decomposition.
- The advantage of first considering symmetric matrices is that the linear transformation defined by such a matrix maps a space into itself, whereas the domain and range spaces of the transformation defined by a rectangular matrix are different.

Eigenvalues and eigenvectors of real symmetric matrices

- A real, symmetric $m \times m$ matrix M always has real eigenvalues and the eigenvectors.
- Eigenvectors of a symmetric matrix can always be chosen to form an orthonormal basis of R^m.
- If the eigenvalues are denoted μ_i and the corresponding eigenvectors are denoted U_i, then $Mu_i = \mu_i u_i$
- .

- If we now write the column vectors $u_1, ..., u_m$ next to each other to form the square matrix

$$U = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ u_1 & u_2 & \dots & u_m \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

then

$$MU = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ \mu_1 u_1 & \mu_2 u_2 & \dots & \mu_m u_m \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

- $$= \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ u_1 & u_2 & \dots & u_m \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \mu_1 & & & \\ & \mu_2 & & \\ & & \ddots & \\ & & & \mu_m \end{pmatrix} = UD,$$

  where D is the diagonal matrix with the eigenvalues on the diagonal.
- Since U is an orthogonal matrix, it is invertible and so

$$M = UDU^{-1} = UDU^T = \sum_{k=1}^{m} \mu_k u_k u_k^T.$$

The decomposition $M = \sum_{k=1}^{m} \mu_k u_k u_k^T$ means that the action of the real symmetric matrix M on an input vector $x \in R^m$ may be understood in terms of three steps

- It resolves the input vector along each of the eigenvectors u_k, the component of the input vector along the ith eigenvector being given by $u_k^T x$,
- The amount along the kth eigenvector is multiplied by the eigenvalue μ_k,
- The product tells us how much of the kth eigenvector u_k is present in the product Mx.

- The eigenvectors of M define a basis in which the action of M is particularly simple.
- Each of the components of x along the m eigenvectors is stretched independently by an amount given by the eigenvalue.

Functions of a real symmetric matrix

- Since a real symmetric matrix M may be written as in $M = UDU^T$, it is easy to compute any power of M

$$M^n = \left(UDU^T\right)^n = UD^nU^T,$$

since $U^TU = UU^T = I$ as U is unitary.

- Since D is diagonal, raising D to the nth power simply raises each of its (diagonal) elements to the nth power. If we define arbitrary functions of a matrix in terms of the associated power series, we see that

$$f(M) = Uf(D)U^T = \sum_{k=1}^{m} f(\mu_k)u_k u_k^T.$$

- The inverse of the matrix M is

$$M^{-1} = \sum_{k=1}^{m} \frac{1}{\mu_k} u_k u_k^T$$

- The matrix is invertible provided that no eigenvalue is equal to zero.
- The eigenvectors of M^{-1} are the same as those of M, only the eigenvalues are reciprocated.
- Each direction which is stretched when M is applied contracted by M^{-1} and vice versa.

Singular value decomposition of a real rectangular matrix

- Let us suppose that $A \in R^{m \times n}$ is a real rectangular matrix which maps vectors in R^n to vectors in R^m.
- We may consider the two square symmetric matrices $A^T A$ and AA^T which may be formed from A and which are of size $n \times n$ and $m \times m$ respectively
- Since each of these matrices are square and symmetric, we may obtain the eigenvectors and eigenvalues of each. The eigenvectors may be chosen to form orthonormal bases of the respective spaces.
- We note that each of the matrices is positive semidefinite, which means that all their eigenvalues are non-negative.

-
- This is easy to see for if v is an eigenvector of $A^T A$, belonging to eigenvalue λ, then $A^T A v = \lambda v$.
- Multiplying on the left by v^T and grouping the terms,

$$v^T A^T (Av) = \lambda v^T v$$

- On the left hand side we have a non-negative quantity, the square of the norm of Av. On the right, $v^T v$ is positive and so λ must be non-negative.

- Label the n orthonormal eigenvectors of $A^T A$ as v_i with associated eigenvalues λ_i and assume that we have sorted them so that

$$\lambda_1 \geq \lambda_2 \geq \ \ldots \lambda_n \geq 0.$$

- Label the m orthonormal eigenvectors of AA^T as u_i with associated eigenvalues μ_i and sort them so that

$$\mu_1 \geq \mu_2 \geq \ldots \geq \mu_m \geq 0.$$

- Let v_1 be the first eigenvector of $A^T A$ and suppose that λ_1 is not equal to zero. The vector Av_1 is then non-zero. We wish to show that Av_1 is in fact an eigenvector of AA^T. To check this, we notice that

$$(AA^T)(Av_1) = A(A^T A)v_1 = \lambda_1 (Av_1).$$

- This shows that Av_1 is indeed an eigenvector of AA^T which belongs to the eigenvalue λ_1. If we normalize Av_1 to have unit length by forming
$$\frac{Av_1}{\|Av_1\|},$$
this is a normalized eigenvector of AA^T and so must be one of the u_i mentioned above provided that the eigenvalues of AA^T are not degenerate.
- Continuing the above argument, we see that each non-zero eigenvalue λ_i of A^TA must also be an eigenvalue of AA^T. A similar argument starting with an eigenvector u_i of AA^T belonging to a non-zero eigenvalue μ_i shows that the vector $ATu_1/\|A^Tu_1\|$ is a normalized eigenvector of A^TA with the same eigenvalue.

- The conclusion to the above is that the non-zero eigenvalues of $A^T A$ are the same as the nonzero eigenvalues of AA^T and vice versa.
- If there are r non-zero eigenvalues, this means that $\lambda_1 = \mu_1, ..., \lambda_r = \mu_r$ and that all subsequent eigenvalues must be zero, i.e., $\lambda_{r+1} = ... = \lambda_n = 0$ and that $\mu_{r+1} = ... = \mu_m = 0$.
- For example, for all $k = 1, ..., r$,

$$u_k = \frac{Av_k}{\|Av_k\|} \text{ and } v_k = \frac{A^T u_k}{\|A^T u_k\|}.$$

- This happens automatically if the non-zero eigenvalues of $A^T A$ and AA^T are non-degenerate.
- Even if there are degeneracies, it is possible to choose the appropriate linear combinations in the degenerate eigenspaces so that these are true.
- The value of r is known as the rank of the matrix A.
- $r \leq m$ and $r \leq n$.

- The norms

$$u_k = \frac{Av_k}{\|Av_k\|} \text{ and } v_k = \frac{A^T u_k}{\|A^T u_k\|}.$$

may be evaluated,

$$\|Av_k\|^2 = (Av_k)^T(Av_k) = v_k^T(A^T A)v_k = \lambda_k,$$

where the last equality holds because v_k is an eigenvalue of $A^T A$ belonging to λ_k, and because v_k is normalized so that $v_k^T v_k = 1$.

- Similarly,

$$\|A^T u_k\|^2 = \mu_k^2.$$

- Since $\lambda_k = \mu_k > 0$, we may define σ_k to be the square root of the eigenvalue and write

$$\|Av_k\| = \|A^T u_k\| = \sigma_k = \sqrt{\lambda_k} = \sqrt{\mu_k},$$

or $k = 1, 2, ..., r$.

- Equation

$$u_k = \frac{Av_k}{\|Av_k\|} \text{ and } v_k = \frac{A^T u_k}{\|A^T u_k\|}.$$

then takes the simple form

$$Av_k = \sigma_k u_k$$

$$A^T u_k = \sigma_k v_k.$$

- The effect of the linear transformation A on the unit vector $v_k \in R^n$ is to take it to the vector $\sigma_k u_k \in R^m$ of length σ_k in the direction of the unit vector $u_k \in R^m$.
- The effect of the linear transformation A^T on the unit vector $u_k \in R^m$ is to take it to the vector $\sigma_k v_k \in R^n$ of length σ_k in the direction of the unit vector $v_k \in R^n$.

- On the other hand for $k > r$, the eigenvalue of $A^T A$ associated with v_k is zero and so $A^T A v_k = 0$.
- Premultiplying this by v_k^T shows that $\|Av_k\| = 0$ and hence that $Av_k = 0$.
- We thus have that $Av_k = 0$ for $k = r + 1, ..., n$
- and similarly, $A^T u_k = 0$ for $k = r + 1, ..., m$.
- Equations $Av_k = 0$ for $k = r + 1, ..., n$ and $A^T u_k = 0$ for $k = r + 1, ..., m$. together describe how A acts on the vectors in the basis vk for $k = 1, ..., n$.
- By linearity, any operator which has the same action as A on each of the vectors of the basis must be the same as A.

- Thus we may write

$$A = \sum_{k=1}^{r} \sigma_k u_k v_k^T,$$

and it is easy to check (using the orthonormality of the basis $\{v_k\}$) that the right hand side does have the same action as A on the basis.

- Taking the transpose of

$$A = \sum_{k=1}^{r} \sigma_k u_k v_k^T,$$

gives

$$A^T = \sum_{k=1}^{r} \sigma_k v_k u_k^T$$

- and it is again easy to check that this is consistent with $A^T u_k = \sigma_k v_k$ and $A^T u_k = 0$ for $k = r + 1, ..., m$.

- The orthonormal vectors $\{v_k\}$ are known as the right singular vectors, the vectors $\{u_k\}$ are known as the left singular vectors, and the scalars $\{\sigma_k\}$ are called the singular values of the matrix A.
- We may write the column vectors u_k next to each other to form an orthogonal $m \times m$ matrix U and stack the row vectors v_k^T on top of each other to form the orthogonal $n \times n$ matrix V^T.

- The equation $A = \sum_{k=1}^{r} \sigma_k u_k v_k^T$ may then be written in matrix form as

$$A = USV^T$$

  where S is an $m \times n$ matrix whose only non-zero elements are the first r entries on the diagonal, i.e., $s_{kk} = \sigma_k$.
-

we discuss the interpretation of the singular value decomposition $A = \sum_{k=1}^{r} \sigma_k u_k v_k^T$. When the matrix A acts on a vector f , we may write the product as

$$Af = \sum_{k=1}^{r} u_k \sigma_k (v_k^T f)$$

this may again be understood as the sequence:

- It resolves the input vector along each of the right singular vectors v_k, the component of the input vector along the kth singular vector being given by $v_k^T f$,
- The amount along the kth direction is multiplied by the singular value σ_k,
- The product tells us how much of the kth left singular vector u_k is present in the product Af.

- The decomposition shows how a complicated operation such as matrix multiplication can be split into r independent multiplications, each of which takes a component along a vector in R^n and converts it into a component along a vector in R^m.
- This result is all the more remarkable since $\{v_k\}_{k=1}^{r}$ can be extended to an orthonormal basis $\{v_k\}_{k=1}^{n}$ for R^n and $\{u_k\}_{k=1}^{r}$ can be extended to an orthonormal basis $\{u_k\}_{k=1}^{m}$ for R^m.

The action of the transpose of A can also be worked out using the singular value decomposition. If $y \in R^m$, we see that,

$$A^T y = \sum_{k=1}^{r} v_k \sigma_k (u_k^T y),$$

which may be understood as the sequence:

- Resolve the vector y along each of the left singular vectors u_k, the component of the input vector along the kth singular vector being given by $u_k^T y$,
- The amount along the kth direction is multiplied by the singular value σ_k,
- The product tells us how much of the kth right singular vector v_k is present in the product $A^T y$.

- Note that the singular values for A^T are the same as those for A.

Geometry of a linear transformation

- Equations $Av_k = \sigma_k$ and $A^T u_k = 0,$ for $k = r+1, ..., m$ tell us that the image of A is spanned by $u_1, ..., u_r$ and the null space of A^T is spanned by $u_{r+1}, ..., u_m$.
- Together they span all of R^m, and hence $R^m = \text{image}(A) \oplus \text{null}(A^T)$.
- Similarly, from equations $Av_k = 0,$ for $k = r+1, .., n$ and $A^T u_k = \sigma_k v_k$, the image of A^T is spanned by $v_1, ..., v_r$ and the null space of A is spanned by $v_{r+1}, ..., v_n$.

-
- Together they span all of R^n, and hence $R^n = \text{image}(A^T) \oplus \text{null}(A)$.
- If we write $C = A \oplus B$ where A, B and C are vector spaces, this means that any vector $c \in C$ can be written in a unique way as the sum of a vector $a \in A$ and a vector $b \in B$. The above direct sums are also orthogonal, o that the vectors a and b are at right angles to each other. The dimensions satisfy $\dim C = \dim A + \dim B$.

- The action of the linear transformation A is to map non-zero vectors in the space image(A^T) into non-zero vectors in the space image(A).
- All vectors which are orthogonal to image(A^T) (i.e., those orthogonal to every row of A) are necessarily in null(A) and are mapped to zero under the action of A.

- Similarly, the action of the linear transformation A^T is to map non-zero vectors in the space image(A) into non-zero vectors in the space image(A^T)
- All vectors which are orthogonal to image(A) (i.e., those orthogonal to every column of A) are necessarily in null(A^T) and are mapped to zero under the action of A^T.

- Let us now see how the singular value decomposition is useful in revealing how a linear transformation converts an image vector f into a data vector Af.
- Any image vector f may be written as the sum of right singular vectors v_k, the length of the component being given by the projection along the singular vector,

$$f = \sum_{k=1}^{n} f_k v_k \text{ where } f_k = v_k^T f.$$

- This is converted into data by multiplication by A, the result is

$$Af = \sum_{k=1}^{r} (\sigma_k f_k) u_k.$$

- Information about the projection of f along v_k is encoded in the data as the component along the direction u_k.
- The size of the projection is multiplied by σ_k to give the coefficient of u_k, namely σf_k.
- The singular value is the factor which tells us by how much each component defining the image is amplified or attenuated when it is converted into the data.

- Notice that only the projections of the image f along the first r right singular vectors v_k play a role in determining the data Af .
- If $r < n$, this means that the data are blind to certain aspects of the image: the data do not allow us to distinguish between images which have the same projections along the first r right singular vectors.
- Such images will look different, since they may differ in their projections along the remaining singular vectors.

- Equivalently, we may add to an image any element of the null space of A and the data will not be changed in any way. If we now think of solving the inverse problem of reconstructing f from a measurement of d, it is clear that at best, those components of f along the first r right singular vectors are determined by the data.
- However the data tell us nothing at all about the components of f along the remaining $n - r$ right singular vectors, and we have to use some other means for determining these components.

- The above indicates that by calculating and plotting the right singular vectors, we can get an idea of what types of structure in the image will be visible in the data and also the types of structure which are invisible in the data.

The Singular Value Decomposition in Model Fitting Problems

- In model-fitting problems, such as fitting of a straight line $y = f_0 + f_1 x$ to a collection of m data points $\{(x_k, y_k)\}_{k=1}^m$, the dimensionality of the image space is very low.
- The forward problem is

$$d = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = Af$$

so image space is two-dimensional ($n = 2$), while data space is m dimensional

- For model fitting to give well-defined answers, the rank r of the matrix A is equal to n, so that the image of A is a two-dimensional subspace of R^m.
- The data sets which lie in the image of A are those for which all the points lie exactly on some straight line.
- For most data sets, the points will not all lie on a straight line, and in these cases $d \in \text{image}(A)$.
- In the least-squares approach, the model parameters $\hat{f}$ are chosen so that the $A\hat{f}$ is as close as possible to d, i.e.,

$$\hat{f} = \text{argmin}\|d - Af\|^2$$

- Suppose we compute the singular value decomposition of A, i.e., we find the left singular vectors $\{u_k\}_{k=1}^m$, the right singular vectors $\{v_k\}_{k=1}^n$ and the singular values σ_k such that:

$$A = \sum_{k=1}^{r} \sigma_k u_k v_k^T$$

- Since $\{u_k\}_{k=1}^m$ form a basis of data space, we may write the data d as a the linear combination:

$$d = \sum_{k=1}^{m} u_k (u_k^T d).$$

- Then, given any f , we see that

$$\|d - Af\|^2 = \left\|\sum_{k=1}^{m} u_k(u_k^T d) - \sum_{k=1}^{r} \sigma_k u_k(v_k^T f)\right\|^2$$

-

$$= \left\|\sum_{k=1}^{r} u_k \left\{(u_k^T d) - \sigma_k(v_k^T f)\right\} + \sum_{k=r+1}^{m} u_k^T(u_k^T d)\right\|^2 .$$

- Using the theorem of Pythagorus (since the vectors $\{u_k\}$ are orthogonal),

$$\|d - Af\|^2 = \left\| \sum_{k=1}^{r} \left|(u_k^T d) - \sigma_k (v_k^T f)\right|^2 + \sum_{k=r+1}^{m} |(u_k^T d)|^2 \right\|^2 .$$

- Choosing $\hat{f}$ so as to minimize $\|d - Af\|^2$is now straightforward.
- The second term on the right-hand side is the square of the perpendicular distance from d to the image of A, and is completely unaffected by the choice of f.

- The first term on the right hand side can be reduced to zero (its minimum possible value) by choosing $\hat{f}$ such that

$$v_k^T \hat{f} = \frac{u_k^T d}{\sigma_k}, \text{ for }, k = 1, 2, ..., r.$$

- Whether or not this completely determines $\hat{f}$ depends on whether $r = n$ or $r < n$. For model fitting, $r = n$, and so the unique solution to the model fitting problem is:

$$\hat{f} = \sum_{k=1}^{n} v_k (v_k^T \hat{f}) = \sum_{k=1}^{n} v_k \frac{u_k^T d}{\sigma_k} = \sum_{k=1}^{n} \frac{1}{\sigma_k} \left(v_k u_k^T \right) d.$$

- Note that the two-dimensional image space is depicted in its entirety, but that image(A), which is a two dimensional subspace in data space, is only depicted schematically by a line.
- The other $m-2$ dimensions in data space are also depicted as a line perpendicular to image(A). The data d are shown as being slightly off image(A) due to noise.
- The reconstructed model parameters $\hat{f}$ are chosen so that $A\hat{f}$ is as close as possible to d in data space.
- The components of $\hat{f}$ along the right singular vectors v_k are given by $\frac{u_k^T d}{\sigma_k}$.
- The singular vectors v_1 and v_2 are at right angles to each other, but may make an angle to the f_0 and f_1 axes which represent the intercept and gradient of the fitted straight line, respectively.

Relationship to the Moore-Penrose inverse

- In order to minimize the misfit $C = \|d - Af\|^2$, we may write

$$C = \|d - Af\|^2 = \sum_{k=1}^{m} \left(d_k - \sum_{l=1}^{n} a_{kl} f_l \right)^2$$

- Then

$$\frac{\partial C}{\partial f_i} = \sum_{k=1}^{m} 2 \left(d_k - \sum_{l=1}^{n} a_{kl} f_l \right) (-a_{ki}) = 0 \text{ for } i = 1, \ldots, n$$

for an extremum.

- This may be written as

$$\sum_{l=1}^{n}\left(\sum_{k=1}^{m} a_{ki}a_{kl}\right) f_l = \sum_{k=1}^{m} a_{ki}d_k,$$

which in matrix form is

$$A^T A f = A^T d.$$

- These are known as the normal equations of the least-squares problem.
- We obtain a unique solution provided that $A^T A$ is invertible, and find the best fit parameters $\hat{f}$ using

$$\hat{f} = (A^T A)^{-1} A^T d$$

- The matrix

$$(A^T A)^{-1} A^T$$

is known as the Moore-Penrose inverse of A.

- Singular-value decomposition:

$$A^TA = \left(\sum_{k=1}^{r} \sigma_k v_k u_k^T\right)\left(\sum_{l=1}^{r} \sigma_l u_l v_l^T\right) = \sum_{k=1}^{r}\sum_{l=1}^{r} \sigma_k \sigma_l v_k (u_k^T u_l) v_l^T$$

$$= \sum_{k=1}^{r} \sigma_k^2 v_k v_k^T$$

since $u_k^T u_l = \delta_{kl}$.

- $\{\sigma_k^2\}_{k=1}^r$ are the nonzero eigenvalues of A^TA. Since A^TA is an $n \times n$ matrix, it is invertible iff $r = n$.

- If the matrix is invertible, then

$$\left(A^T A\right)^{-1} = \sum_{k=1}^{r} \frac{1}{\sigma_k^2} v_k v_k^T,$$

-

$$\left(A^T A\right)^{-1} A^T = \left(\sum_{k=1}^{r} \frac{1}{\sigma_k^2} v_k v_k^T\right) \left(\sum_{l=1}^{r} \sigma_l v_l v_l^T\right) = \sum_{k=1}^{r} \frac{1}{\sigma_k} v_k u_k^T$$

- shows that the least squares solution as calculated using the singular value decomposition is identical to that using the Moore Penrose inverse.

Effects of Noise on Model Parameter Estimates

- How well is the model parameters $\hat{f}$ are determined in a least squares fitting procedure.
- The method is based on the idea that the value of $\|d - Af\|^2$ is a measure of how unlikely f is when the measured data are d.
- The best parameter estimate $\hat{f}$ is thus the one which minimizes $\|d - Af\|^2$

$$\|d - A\hat{f}\|^2 = \min_{f \in R^n} \|d - A\hat{f}\|^2 = r_m^2$$

in where $r_{\min}$ is the distance between the data d and the image of A,

- The value of $r_{\min}$ allows us to estimate the amount of noise likely to be present.
- In order to get some idea of how confident we are about $\hat{f}$, we consider the set of probable f values for the given data and noise level.
- This is known as the feasible set,

$$F = \left\{ f : \|d - Af\|^2 = \alpha r_{\min}^2 \right\},$$

where α is chosen according to the confidence level required

- Using the singular value decomposition of A and orthogonality , we find that

$$\|d - Af\|^2 = \sum_{k=1}^{r} \left| u_k^T d - \sigma_k (v_k^T f) \right| + r_{\min}^2$$

- Substituting the values of v_k^T, we get

$$\|d - Af\|^2 = \sum_{k=1}^{r} \left| \sigma_k (v_k^T \hat{f}) - \sigma_k (v_k^T f) \right|^2 + r_{\min}^2$$

$$= \sum_{k=1}^{r} \sigma_k^2 \left| v_k^T (f - \hat{f}) \right|^2 + r_{\min}^2$$

- From the definition of the set F, we want those f which satisfy

$$\sum_{k=1}^{r} \sigma_k^2 \left| v_k^T (f - \hat{f}) \right|^2 \leq (\alpha - 1) r_{\min}^2$$

- the independent uncertainties along the singular vector directions v_k translate into correlated uncertainties along the axes f_0 and f_1 of the estimated parameters

The Singular Value Decomposition in General

- As the number n of parameters in the model becomes large, it becomes difficult to ensure that they are all independent.
- Once there are dependent parameters, it becomes possible to achieve the same data vector from more than one set of parameters.
- As n becomes large, the problem of model fitting merges into that of indirect imaging.
- In the regime of indirect imaging, the number of points in image space is chosen to be so large that it can adequately represent the object of interest.
- Since there are now many images which map to the same data, it becomes necessary to choose from among them using an additional criterion of optimality.

- In terms of the singular value decomposition, the rank r of the forward map A is less than n when the parameters of the model are not independent.
- This means that the null space of A contains some non-zero vectors.
- In fact, all vectors which are linear combinations of $v_{r+1}, ..., v_n$ are in the null space of A, so that

$$A(c_{r+1}v_{r+1} + \quad + c_n v_n) = 0$$

for every choice of coefficients $c_{r+1}, ..., c_n$.

- Let us now consider what happens if we try to use the principle of least squares to reconstruct the image, when $r < n$.
- For measured data d, and a trial image f , states that

$$\|d - Af\|^2 = \sum_{k=1}^{r} \left| u_k^T d - \sigma_k (v_k^T f) \right|^2 + \sum_{k=r+1}^{m} \left| u_k^T d \right|^2 .$$

- In order to minimize $\|d - Af\|^2$ over all possible f , the best that we can do is to ensure that the first term on the right hand side is equal to zero.

- If $r = n$, this completely defines f , but if $r < n$, we see that only the projections of f along the first r right singular vectors $v_1, ..., v_r$ are determined.
- The projections on the remaining $n - r$ are completely arbitrary. Thus, instead of a single best solution $\hat{f}$, all images of the form:

$$\sum_{k=1}^{r} v_k \left(\frac{u_k^T d}{\sigma_k} \right) + c_{r+1} v_{r+1} + ... + c_n v_n$$

for every choice of coefficients $c_{r+1}, ..., c_n$ will give the same minimum value for $\|d - Af\|^2$.

- if some of the arbitrary coefficients are large, the reconstructed image will look terrible.
- We need to select the arbitrary coefficients according to some other criterion of optimality, because the data do not determine them at all.
- If $r < m$, it is very likely that noise will cause the data d to lie outside the image of A.
- If y is the point which is closest to d in image (A) , we find that an infinite number of images f map to the point d.
- All such images differ by some vector in null (A) , and so the set S in image space which maps to y is represented schematically by a line parallel to the subspace null (A) .

- The essential goal of regularization methods is to choose a reasonable solution within the feasible set.

Classifying linear operators

- If the image of the operator A has smaller dimension than the data space, i.e., if $r < m$, there are vectors $d \in R^m$ which lie outside the range of A.
- This means that for some possible d, the equations $Af = d$ have no solution and it is necessary to use some condition such as least-squares to select a vector $\hat{f}$ which minimizes the discrepancy between Af and d.
- Whether or not this least-squares solution is unique or not depends on the next condition.

- If the image of the operator A has smaller dimension than the image space, i.e., if $r < n$, then there are an infinite number of vectors x all of which map under A to the same vector d.
- This means that for some choices of d, the equations $Af = d$ have infinitely many solutions.
- Whether or not solutions exist for all values of d depends on the previous condition.

- The only situation in which $Af = d$ has a unique solution for every d is if $r = m = n$ so that the matrix A is square and invertible.
- This situation almost never holds in practice and it is almost always a bad idea to try to force an inverse problem into a form with a square invertible matrix with a view of solving the problem by a solution of these equations.

- Whenever the second condition holds, it is necessary to use additional information over and above the data collected in order to select a good reconstruction from among the possible reconstructions.
- One way of doing this is by using the process of regularization which we shall examine in more detail later.
- In practice, it is often the case that both the first two conditions hold and r is strictly less than both m and n. When this is the case, there is no solution to $Af = d$ for some d while there are infinitely many solutions for other possible d.
- If there is no solution for some d, it makes sense to find f to minimize $\|d - Af\|^2$.
- It turns out that this minimization gives a unique f if $r = n$ but there are an infinity of vectors all of which give exactly the same minimum norm if $r < n$.

The effects of noise and small singular values

- we have drawn a sharp distinction between the eigenvalues of $A^T A$ which are non-zero and those which are zero.
- Indeed the rank r of A may be defined as the number of non-zero eigenvalues of $A^T A$.
- In practice of course, when the eigenvalues of $A^T A$ are sorted in decreasing order, there is a smooth transition from the large eigenvalues through the small eigenvalues to the tiny eigenvalues and the actual rank is always equal to the smaller of m or n.
- A more useful concept is the effective rank, which depends on the threshold below which we consider the eigenvalue to be negligible.

- For typical measurement processes, large or slowly-varying portions in the image are well represented in the data while structures on fine scales or with high frequency components tend to be poorly represented.
- This is because measurements can usually only be made with a certain spatial or temporal resolution.
- For example, if we are taking a photograph of an object with visible light, structures on the object with a scale smaller than the wavelength are invisible.
- The singular vectors in image space associated with the large singular values for such an operator will tend to be smooth, while those associated with the small singular values will tend to be highly irregular or oscillatory.

- In order to understand how the small singular values affect the reconstruction process, we consider a simple model for the measurement uncertainty or noise that is always present in data.
- Let us suppose that the measured data d may be written as the sum of the transformed image $y = Af$ and a noise vector n so that

$$d = Af + n.$$

- The vector f represents the true underlying image and y is the data that would have been obtained in the absence of noise.

- Neither of these quantities is known in practice, but the aim of reconstruction is to find a vector approximating to f.
- Substituting the singular-value decomposition of A into this yields

$$d = \sum_{k=1}^{n} \sigma_k u_k v_k^T f + n$$

where the rank has been taken to be n, the size of the image space (assumed to be smaller than m).
- The forward mapping is strictly 1-1, and so there is a unique least-squares solution which we have seen is given by:

$$\hat{f} = \sum_{k=1}^{n} \left(\frac{u_k^T d}{\sigma_k} \right) v_k.$$

- Substituting the expansion gives

$$\hat{f} = \sum_{k=1}^{n} \left(v_k^T f + \frac{u_k^T n}{\sigma_k} \right) v_k$$

$$= f + \sum_{k=1}^{n} \frac{u_k^T n}{\sigma_k} v_k$$

  where we have made use of the fact that the $\{vk\}$ form a basis for R^n.

- We see that the reconstruction is the sum of the true image and terms due to the noise.
- The error term along the direction of v_k in image space arises from the component of the noise in the direction of u_k in data space, divided by the singular value σ_k.

- If we now suppose that some of the singular values σ_k are small, this division will give a very large random component, often completely swamping the component of f in that direction.
- Another way of thinking about this is to see that the small singular values correspond to directions in image space for which the data contain very little information about the image.
- In attempting to reconstruct those aspects of f which lie along these directions, we have to amplify the small signal buried in the data by dividing by the small singular value.
- Such a scheme is risky because the noise which is inevitably present in the data is also going to be amplified by a large factor, corrupting the reconstruction.

- when there are small singular values, the least squares method can give bad reconstructions.
- It is better to consider small singular values as being effectively zero, and to regard the components along such directions as being free parameters which are not determined by the data.

- When the singular values of the measurement operator A are ranked in non-increasing order, the rate at which they decrease with index gives valuable information about how much we can hope to reconstruct from data taken using that measurement process for a given amount of noise in the data.
- The more rapid is the decrease, the less we can reconstruct reliably for a given noise level.
- Equivalently, in order to get good reconstructions when the singular values decrease rapidly, an extremely high signal-to-noise ratio in the data is required.

Regularization Methods for Linear Inverse Problems

- The primary difficulty with linear ill-posed problems is that the inverse image is undetermined due to small (or zero) singular values of A.
- Actually the situation is a little worse in practice because A depends on our model of the measurement process and that is typically not precisely known leading to a slight imprecision in the singular values.
- Usually that is not significant for the large singular values, but may lead to ambiguity in the small singular values so that we do not know if they are small or zero.
-

The Data Misfit and the Solution Semi-Norm

- We considered the linear problem

$$d = Af + n$$

  and focussed on the structure of the operator $A \in R^{m \times n}$.
- As far as the data are concerned, a reconstructed image $\hat{f}$ is good provided that it gives rise to mock data $A\hat{f}$ which are close to the observed data.
- Thus, one of the quantities for measuring the quality of $\hat{f}$ is the data misfit function or the square of the residual norm

$$C(f) = \|d - Af\|^2.$$

- We have seen that choosing $\hat{f}$ so as to minimize $C(f)$ usually gives a poor reconstruction.
- If the rank of the operator A is less than n, there are an infinite number of reconstructions, all of which minimize $C(f)$, since the data are not affected by adding to a reconstruction any vector which lies in the null space of A.
- In the presence of noise, finding the (possibly non-unique) minimum of C is undesirable as it leads to amplification of the noise in the directions of the singular vectors with small singular values. Instead, we usually regard the data as defining a feasible set of reconstructions for which $C(f) \leq C_0$ where C_0 depends on the level of the noise. Any reconstruction within the feasible set is to be thought of as being consistent with the data.

- Since the data do not give us any information about some aspects of f , it is necessary to include additional information which allows us to select from among several feasible reconstructions.
- Analytical solutions are available if we choose sufficiently simple criteria.
- One way of doing this is to introduce a second function $\Omega(f)$ representing our aversion to a particular reconstruction.
- For example, we may decide that the solution of minimum norm should be chosen from among the feasible set. This can be done by choosing

$$\Omega(f) = \|f\|^2.$$

- Sometimes, we have a preference for reconstructions which are close to some default solution f^{∞}.
- This may be appropriate if we have historical information about the quantity.
- This can be done by choosing

$$\Omega(f) = \|f - f^{\infty}\|^2.$$

More generally, it may not be the norm of $f - f^{\infty}$ which needs to be small, but some linear operator acting on this difference.
- Introducing the operator L for this purpose, we can set

$$\Omega(f) = \|L(f - f^{\infty})\|^2 = (f - f^{\infty})^t L^t L(f - f^{\infty})$$

- If the image space is n dimensional and the data space is m dimensional, the matrix A is of size $m \times n$ and the matrix L is of size $p \times n$ where $p - n$.
- Typically, L is the identity matrix or a banded matrix approximation to the $(n-p)$th derivative.

- For example, an approximation to the first derivative is given by the matrix

$$L_1 = \frac{1}{\Delta x}\begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix},$$

- while an approximation to the second derivative is

$$L_2 = \frac{1}{(\Delta x)^2}\begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}.$$

- In other cases, it may be appropriate to minimize some combination of the derivatives such as

$$\Omega(f) = \alpha_0 \|f - f^\infty\|^2 + \sum_{k=1}^{q} \alpha_k \|L_k(f - f^\infty)\|^2,$$

where L_k is a matrix which approximates the k th derivative, and α_k are non-negative constants.

- Such a quantity is also the square of a norm, called a Sobolev norm.

- It is a simple result from linear algebra that any real symmetric positive semidefinite matrix may be factorized into a product of the form $L^t L$ where L is a lower triangular matrix.
- A constructive proof of this result leads to the so-called Cholesky factorization of the square matrix.
- The Sobolev norm above may thus also be written in the form of $\Omega(f) = \|L(f - f^{\infty})\|^2$ for a suitable choice of L.
- There are many ways of balancing the often conflicting requirements of equations $\Omega(f) = \|L(f - f^{\infty})\|^2$ and $C(f) = \|d - Af\|^2$ and these lead to a variety of regularization methods.

Tikhonov regularization

- This is perhaps the most common and well-known of regularization schemes.
- We form a weighted sum of $\Omega(f)$ and $C(f)$ using a weighting factor λ^2, and find the image $\hat{f}_\lambda$ which minimizes this sum, i.e.,

$$\hat{f}_\lambda = \operatorname{argmin}\left\{\lambda^2\|L(f - f^\infty)\|^2 + \|d - Af\|^2\right\}.$$

- This is a whole family of solutions parameterized by the weighting factor λ^2.
- We call λ the regularization parameter.

- If the regularization parameter is very large, the effect of the data misfit term $C(f)$ is negligible to that of $\Omega(f)$ and we find that
$$\lim_{\lambda \to \infty} \hat{f}_\lambda = f^\infty.$$
- With a large amount of regularization, we effectively ignore the data (and any noise on the data) completely and try to minimize the solution semi-norm which is possible by choosing the default solution.
- On the other hand, if λ is small, the weighting placed on the solution seminorm is small and the value of the misfit at the solution becomes more important.
- Of course, if λ is reduced to zero, the problem reduces to the least-squares case considered earlier with its extreme sensitivity to noise on the data.

- A formal solution to the problem may readily be found.
- We set

$$\frac{\partial}{\partial f_k}\left\{\lambda^2(f-f^\infty)^t L^t L(f-f^\infty)+(d-Af)^t(d-Af)\right\}=0,$$

  for $k=1,2,\ldots,n$.
- This leads to the simultaneous equations

$$2\lambda^2 L^t L(f-f^\infty)-2A^t(d-Af)=0,$$

  or

$$(\lambda^2 L^t L+A^t A)f=\lambda^2 L^t L f^\infty+A^t d.$$

  Setting $\lambda=0$ reduces this system of equations to the normal equations associated with the usual least squares problem.
-

- For non-zero values of λ, the additional term $\lambda^2 L^t L$ in the matrix on the left-hand side alters the eigenvalues (and eigenvectors) from those of $A^t A$ alone.
- So long that

$$\lambda^2 L^t L + A^t A$$

is non-singular, there is a unique solution. The problem of image reconstruction is thus reduced to solving a (large) system of simultaneous equations with a symmetric positive definite coefficient matrix.

Truncated Singular Value Decomposition (TSVD)

- Let us suppose that the operator A has the singular value decomposition
$$A = \sum_{l=1}^{r} \sigma_l u_l^t v_l.$$
- The truncated singular value decomposition (TSVD) method is based on the observation that for the larger singular values of A, the components of the reconstruction along the corresponding singular vector is well-determined by the data, but the other components are not well-determined.

- An integer $k \leq n$ is chosen for which the singular values are deemed to be significant and the solution vector $\hat{f}$ is chosen so that
$$v_l^T \hat{f} = \frac{u_l^T d}{\sigma_l} \text{ for } l = 1, ..., k.$$
- The components along the remaining singular-vector directions $\{v_l\}$ for $l = k+1, ..., n$ are then chosen so that the total solution vector $\hat{f}$ satisfies some criterion of optimality, such as the minimization of a solution semi-norm of the form
$$\Omega(f) = \|L(f - f^{\infty})\|^2$$
as above.

- Let us denote by V_k the $n \times (n-k)$ matrix whose columns are $\{v_l\}$ for $l = k+1, \ldots, n$ so that V_k is the matrix whose columns span the effective null space of A.
- The reconstruction which has zero projection in this effective null space is

$$f' = \sum_{l=1}^{k} \frac{u_l^T d}{\sigma_l}.$$

- The desired reconstruction $\hat{f}$ must be equal to the sum of f' and a vector which is the superposition of the columns of V_k.
- This may be written as

$$\hat{f} = f' + \sum_{l=k+1}^{n} c_l v_l = f' + V_k c$$

for a $n - k$ element column vector c. The solution semi-norm of this reconstruction is

$$\Omega(\hat{f}) = \|L(\hat{f} - f^{\infty})\|^2 = \|L\left(f' + V_k c - f^{\infty}\right)\|^2$$
$$= \|L(f' - f^{\infty}) + (LV_k)c\|^2$$

- The vector c which minimizes this semi-norm is

$$c = -(LV_k)^{\dagger} \left(f' - f\right).$$

where the dagger represents the Moore-Penrose inverse. i.e., for any matrix A, we define

$$A^{\dagger} = \left(A^t A\right)^{-1} A^t.$$

- This gives an explicit expression for the truncated singular value decomposition solution, namely

$$\hat{f} = f' - V_k (LV_k)^{\dagger} L (f' - f).$$

- Note that some authors use the terminology truncated singular value decomposition to refer to the special case where L is chosen to be the identity matrix, and call the general case derived above the modified truncated singular value decomposition.

Filter factors

- In any regularization scheme, there is a regularization parameter which is a quantity that can be adjusted in order to change the degree of regularization of the solution.
- For values of this parameter at one end of its range, the solution is usually smoother, more similar to the default solution and less affected by noise on the data whereas for values of this parameter at the other end, the solution can be very sensitive to noise as it is primarily determined by the requirement of minimizing the data residual.
- In the case of Tikhonov regularization, the parameter is the quantity while in the case of the TSVD method, it is the choice of k at which the singular values are deemed to be negligible.

- It is useful to be able to look at the range of solutions which result as the regularization parameter is varied.
- This can always be done by recomputing the solution from scratch for each value of the parameter, but this is computationally very intensive as we often need to invert a large matrix for each choice of the regularization parameter.
- An advantage of studying the singular value decomposition is that it provides a convenient way of investigating the family of regularized solutions without having to reinvert large matrices.
- We shall thus re-examine the regularization methods described above in terms of the singular value decomposition.

- Tikhonov regularization can be analyzed in this way when the matrix L happens to be the identity.
- The solution to the problem is given by

$$\left(\lambda^2 I + A^T A\right)\hat{f} = \lambda^2 f^{\infty} + A^t d$$

- Let us suppose that we have computed the singular value decomposition of A in the usual form

$$A = \sum_{l=1}^{r} \sigma_l u_l v_l^t$$

- then

$$\left(\lambda^2 I + A^t A\right) \hat{f} = \lambda^2 \sum_{l=1}^{n} \hat{f}_l v_l + \sum_{l=1}^{r} \sigma_l^2 \hat{f}_l v_l$$

- $$= \sum_{l=1}^{r} \left(\lambda^2 + \sigma_l^2\right) \hat{f}_l v_l + \lambda^2 \sum_{l=r+1}^{n} \hat{f}_l v_l$$

where $\hat{f}_l = v_l^t \hat{f}$.

- $$\lambda^2 \hat{f}^\infty + A^t d = \lambda^2 \sum_{l=1}^{n} f_l^\infty v_l + \sum_{r=1}^{r} \lambda_l d_l v_l)$$

$$= \sum_{l=1}^{r} \left[\lambda^2 f_l^\infty + \sigma_l \frac{d_l}{\sigma_l} \right] v_l + \lambda^2 \sum_{l=r+1}^{n} f^\infty v_l$$

where $f^\infty = v_l^t f^\infty, d_l = u_l^t d$ and we have made use of the fact that

$$I = \sum_{l=1}^{n} v_l v_l^t$$

since the $\{v_l\}_{l=1}^n$ form an orthonormal basis of R^n.

- Equating

$$\sum_{l=1}^{r} \left(\lambda^2 + \sigma_l^2\right) \hat{f}_l v_l + \lambda^2 \sum_{l=r+1}^{n} \hat{f}_l v_l$$

and

$$\sum_{l=1}^{r} \left[\lambda^2 f_l^{\infty} + \sigma_l \frac{d_l}{\sigma_l}\right] v_l + \lambda^2 \sum_{l=r+1}^{n} f^{\infty} v_l$$

and using the linear independence of the vectors v_l we see that

- $$\hat{f}_l = \begin{cases} \frac{\lambda^2}{\lambda^2+\sigma_l^2} f_l^{\infty} + \frac{\sigma_l^2}{\lambda^2+\sigma_l^2} \frac{d_l}{\sigma_l} \text{ for } l = 1, 2, ..., r, \\ f_l^{\infty} \text{ for } l = r+1, ..., n. \end{cases}$$

- This displays the solutions to the regularization problem for all values of λ in a convenient form.
- In the directions of the singular vectors $v_{r+1}, ..., v_n$ which span the null space of A, the projections $\hat{f}_l$ of the regularized solution are equal to the projections of the default solution f_l^{∞} .
-

- This is not unreasonable as the data do not give us any information about those aspects of the image.
- On the other hand, along each of the directions $v_1, \ldots, v_r$ for which the data do give us some information, the regularized solution is a weighted linear combination of f_l^{∞} , which is what the default solution would have us take, and of $\frac{d_l}{\sigma_l}$ which is what the data alone would have suggested.
- Notice that since the weights add up to one, the regularized solution lies along the line in n space joining these points. The value of the weights is also of interest.

-
- As λ becomes large, the solution is pulled towards the default solution, as expected, but it should be noticed that where along the line the solution ends up at depends on the relative values of λ and of σ_l. The larger is the singular value σ_l, the smaller is the relative effect of the regularization parameter λ.
- In fact we need to have $\lambda = \sigma_l$ in order to pull the component of the solution to the midpoint of the line joining $f_l^{\infty} v_l$ and $(\frac{d_l}{\sigma_l})v_l$.
- This is desirable since it is precisely in the directions corresponding to the large singular values that the data give us the greatest information.

- The quantities
$$\frac{\sigma_l^2}{\lambda^2 + \sigma_l^2}$$
which multiply the data coefficient $\left(\frac{d_l}{\sigma_l}\right)$ are called filter factors.
- They show how the algorithm reduces the weighting for the data which are associated with the small singular values.
- Depending on the level of the noise on the data, we need different amounts of protection against the noise-amplifying effects of reconstruction using the small singular values.

- By contrast to the Tikhonov regularization algorithm in which the filter factors smoothly decrease to zero as the singular values gets smaller, the truncated singular value algorithms have filter factors which are equal to unity for those singular values which are deemed to be non-negligible $(l - k)$ and to zero for those singular values which are negligible $(l > k)$.
- The value of the components of the regularized solution in the directions of the significant singular values are completely determined by the data, as

$$\hat{f}_l = \frac{d_l}{\sigma_l} \text{ for } l = 1, 2, ..., k$$

- The other components $\hat{f}k + 1, ..., .\hat{f}n$ are adjusted so as to minimize the solution semi-norm.

Smoothness versus data-fitting in the deblurring example

- The regularizing parameter, λ, can be thought of as controlling the balance between minimizing the regularizing term which is often a criterion of smoothness of the reconstructed image and minimizing the term which corresponds to fitting the data.
- When λ is small, there is little weight put on the regularizing term, the data is fitted well and the reconstructed image is not smooth.
- Conversely when λ is large, the regularizer dominates the minimization and the reconstructed image is smooth at the expense of not fitting the data so well.

Choosing the regularization parameter

- We have seen that λ sets the balance between minimizing the residual norm $\|d - Af_{\lambda}\|_2$and minimizing the roughness $\|f_{\lambda} - f^{\infty}\|_2$.The big question now is how to choose λ?
-

The L-Curve

- Perhaps the most convenient graphical tool for setting λ is the $L-$curve.
- When we plot $\log \|d - Af_\lambda\|$ versus $\log \|f_\lambda - f^\infty\|$ (for a discrete problem) we get the characteristic $L-$shaped curve with a (often) distinct corner separating vertical and horizontal parts of the curve.

- The rationale for using the L curve is that regularization is a trade-off between the data misfit and the solution seminorm.
- In the vertical part of the curve the solution seminorm $\|L(f - f^{\infty})\|$ is a very sensitive function of the regularization parameter because the solution is undergoing large changes with λ in an attempt to fit the data better.
- At these low levels of filtering, there are still components in the solution which come from dividing by a small singular value, which corresponds to having inadequate regularization.
- On the horizontal part, the solution is not changing by very much as λ is changed.
- However, the data misfit is increasing sharply with more filtering.

- Along this portion, our preference for the more highly filtered solutions increases only slightly at the cost of a rapidly rising data misfit, and so it is desirable to choose a solution which lies not too far to the right of the corner.
- The following figure shows the L-curve that is associated with the deblurring.
- The curve is labelled parametrically with the values of the regularization parameter. In this example, the bend in the curve is not very sharp, and the solution which appears optimal visually lies slightly to the right of the position of largest upwards-pointing curvature.

Solving Large Systems of Simultaneous Equations for Regularization Problems

- This is a very large subject in its own right which we cannot treat in detail.
- Although the singular value decomposition (and the generalized SVD) is useful for the theoretical study of inverse problems, it is rather numerically intensive to calculate in practice.
- As a rough guide, finding the SVD is feasible when there are up to a few hundred variables, and so is of limited use for problems involving several thousand to several million components in the image and data vectors.

- The mapping from the image to the data is specified by multiplication by the matrix A.
- For large problems, it is often not feasible to store the matrix A in the computer, or to compute its action by direct multiplication.
- If for example, we take the problem of blurring an image consisting of a 256×256 pixel scene to produce a data photograph of the same size, the sizes of image and data space are each 65536 dimensional, and the matrix A has 256^4 elements.
- In the case of blurring, we know that the two-dimensional convolution corresponding to the action of A can be computed efficiently via the trick of multiplying together Fourier transforms, and so the matrix A is never formed explicitly.

- When writing algorithms for solving such problems, we cannot use methods which involve manipulating the full matrices, and we should regard even the storage of vectors in image and data space being rather costly.
- For generality, we assume that the user will provide us with a function that will calculate Av and Lv when provided with a vector $v \in R^n$.
- As we shall see, we also need to be provided with routines to calculate $A^t u$ for $u \in R^m$ and $L^t w$ for $w \in R^p$.
- These functions will usually employ some trick such as using Fourier transforms or exploiting the sparsity of the matrices in order to do the calculation in a reasonable time.
-

- In the above, we derived the simultaneous equations for Tikhonov regularization from the minimization of the function

$$\mathcal{L}(f) = \lambda^2 \| L(f - f^{\infty}) \|^2 + \| d - Af \| 2$$

- For the linear inverse problem, this is a quadratic in the image f.
- Before describing the algorithm, we need to consider some properties of such quadratic expressions

Minimizing a quadratic in many dimensions

- A quadratic expression in a vector $x \in R^n$ has the general form

$$Q(x) = x^t Hx - x^t G - G^t x + Q_0$$

where $H \in R^{n \times n}$ is a real symmetric matrix, $G \in R^n$ is a vector and $Q_0 \in R$.

- This quadratic has a stationary point whenever

$$\frac{\partial Q}{\partial x_i} = 2 \sum_j h_{ij} x_j - 2g_i = 0$$

for all $i \in \{1, ..., n\}$. i.e., stationary points x_s of Q satisfy the set of linear equations $Hx_s = G$

- Depending on the nature of the matrix H, this can have none, one or infinitely many solutions.
- A special case of interest is when H is positive definite, which means that $x^t Hx \geq 0$ for all x, and that $x^t Hx = 0$ only if $x = 0$. If H is positive definite, it is invertible and there is a unique stationary point $x_s = H^{-1}G$

- The original quadratic may be written as

$$Q(x) = (x - x_s)^t H(x - x_s) + \left(Q_0 - x^t sHx_s\right)$$

  from which it is clear that x_s is the global minimum of Q.
- When H is positive definite, we may regard solving the system of equations $Hx_s = G$ and minimizing the quadratic $Q(x) = x^t Hx - x^t G - G^t x + Q_0$ as being equivalent problems.
-

- Let us now consider the problem of minimizing $Q(x)$ when n is so large that computing H^{-1} explicitly is not feasible.
- We can consider an iterative algorithm which proceeds from an initial guess x_0 of the minimum to a point x_1 which is hopefully a better estimate of x_s.
- One way of doing this is to consider the direction of steepest descent from x_0.
- This is along the direction opposite to the gradient of Q at x_0. We see that

$$\nabla Q = 2(Hx - G)$$

and so the direction of steepest descent is along $-\nabla Q$ which is parallel to $s_1 = G - Hx_0$.

- We now proceed along the line $x_0 + c_1 s_1$ trying to find a better approximation to x_s.
- It is easy to find out how Q behaves on this line, since

$$Q(x_0+c_1s_1) = (x_0+c_1s_1)^t H(x_0+c_1s_1) - (x_0+c_1s_1)^t G - G^t(x_0+c_1s_1) +$$

$$= \left(s_1^t H s_1\right) c_1^2 - \left\{s_1^t(G - Hx_0) + (G - Hx_0)^t s_1\right\} c_1 + Q(x_0)$$

- This is a quadratic in c1 whose minimum is readily found to be at

$$c_1 = s_1^t(G - Hx_0) + (G - Hx_0)ts12(s_1^t H s_1) = \frac{s_1^t s_1}{s_1^t H s_1}$$

- We now set $x_1 = x_0 + c_1 s_1$ as our next estimate of the position of x_s.
- Notice that in order to compute c_1, all that we need to be able to do is to calculate Hs_1 and to carry out arithmetic with vector quantities.
- So long that Hs_1 can be computed efficiently, no large matrices need to be stored.
- We can proceed iteratively, finding $-\nabla Q(x_1)$ and searching along this direction from x_1 to find the point x_2 which minimizes Q along this line.
- This is known as the steepest descent algorithm for minimizing a function.

- At each iteration, a one dimensional search is carried out, and the hope is that a succession of these will ultimately lead to the minimization of the n dimensional quadratic.
- Despite its intuitive appeal, it is a very inefficient way of minimizing a function of many variables. Unless the contours of Q are spherical, it requires many more than n iterations to find the minimum due to a phenomenon called hem-stitching in which the succession of iterates slowly crawls down the walls of a long elliptical valley.
- In effect, each successive search largely undoes the work of the previous step, and the result is only a very gradual reduction in Q.
- For large n, this algorithm is essentially useless.

- Instead of starting at an initial guess and searching along a single line for the minimum of Q, we can consider searching in a larger subspace starting from an initial guess.
- For the moment, suppose that someone gives us a list of linearly independent search directions, $s_1, s_2, \ldots, s_k$ and asks us to minimize Q within the subspace

$$x_0 + c_1 s_1 + \ldots + c_n s_n = x_0 + Sc$$

where S is the matrix whose columns are the search directions and c is the column vector of the coefficients.

- Within this k dimensional affine space, we see that

$$Q(x_0+Sc) = (x_0+Sc)^t H(x_0+Sc) - (x_0+Sc)^t G - G^t(x_0+Sc) + Q_0$$

$$= c^t\left(S^t HS\right)c - c^t S^t(G-Hx_0) - (G-Hx_0)^t Sc + \left(Q_0 + x_0^t Hx_0 - x_0^t G -\right.$$

$$= c^t\tilde{H}c - c^t\tilde{G} - \tilde{G}^t c + \tilde{Q}_0$$

which is also a quadratic in c, with the matrices

$$\tilde{H} = S^t HS$$

$$\tilde{G} = S^t(G - Hx_0)$$

$$\tilde{Q}_0 = Q(x_0) = Q_0 + x_0^t Hx_0 - x_0^t G - G^t x_0$$

- Since the columns of S are linearly independent and the matrix H is positive definite, so is the matrix $\tilde{H}$.
- By the above discussion, the minimum of $Q(x_0 + Sc)$ in this affine subspace is located at

$$\hat{c} = \tilde{H}^{-1}\tilde{G}$$

- This can be readily computed since the matrix $\tilde{H}$ is only of size $k \times k$ where k is the number of search directions in the subspace.
- When using a subspace search, the next guess at the minimizing point x_s is

$$x_1 = x_0 + S\hat{c} = x0 + S\tilde{H}^{-1}\tilde{G}$$

- We can then proceed iteratively to get a succession of estimates to x_s.

- The efficiency of the resulting algorithm depends critically on making an appropriate choice for the search directions.
- One might imagine starting at x_0 and searching in the direction of $s_1 = -\nabla Q(x_0)$, thus reaching x_1.
- From there, we find $s_2 = -\nabla Q(x_1)$, but instead of simply searching along this line, we can search in the affine subspace spanned by s_1 and s_2 starting at x_1.
- Having found the minimum of Q in this subspace at x_2, we then set $s_3 = -\nabla Q(x_2)$ and search the subspace spanned by s_1, s_2 and s_3.

- In this way, we search for the minimum over larger and larger spaces, which ensures that each search does not undo the work of the previous searches.
- The spaces generated in this manner are known as the Krylov subspaces. The algorithm, as described would rapidly become unworkable since the search space increases in size on each iteration.
- However, due to a truly amazing result which we do not have time to prove, it turns out that for quadratics, it is possible to achieve the same result as searching over all the space spanned by all the previous search directions by searching firstly along $-\nabla Q(x_0)$ and subsequently over only a two-dimensional space on each iteration.

- Suppose we have reached the point x_l after the lth iteration.
- We calculate $-\nabla Q(x_l)$ as before and search within the affine space spanned by this vector and the vector $x_l - x_{l-1}$.
- It can be shown that in the absence of round-off errors, the effect is the same as if we had searched in the space spanned by $-\nabla Q(x0)$, $-\nabla Q(x_1)$, ...,$-\nabla Q(x_l)$.
- With this algorithm and perfect arithmetic, one can reach the minimum of an n dimensional quadratic in at most n steps.

Matlab Code

- The following Matlab code illustrates how a search can be carried out over an arbitrary affine subspace around the current point. The user specifies the search directions as the columns of the matrix S.
- This code was written by Sze Tan, University of Auckland, July 1998.

```
 function [xnew,Qpred] = search1(x0,res,Hfunc,Qnow,S)
% Searches in an affine subspace for the minimum of the
% quadratic
% Q(x) = x*H*x-x*G-G*x+Q0
% x0 = Initial guess of location of the minimum
% res = G-H*x0
% Hfunc = Name of function which calculates H*x for given x
% Qnow = Value of Q(x0)
% S = matrix with search directions as its columns


nsrch = size(S,2);
HS = zeros(length(x0),nsrch);
fprintf(Value of quadratic (current) = %f\n,Qnow);
for k = 1 : nsrch
HS(:,k) = feval(Hfunc,S(:,k));
end
```

```
Hq = S*HS;
Gq = S*res;
c = Hq\Gq;
Qpred = Qnow + c*Hq*c - c*Gq - Gq*c;
fprintf(Value of quadratic (predicted) = %f\n,Qpred);
xnew = x0 + S * c;
```

- In this code, the function whose name is in Hfunc is used to apply the matrix H to an arbitrary vector.
- In general, this will compute the product in some indirect way for efficiency.
- However, as an illustrative example, the following shows how we can solve the problem $Hx = G$ for a small matrix H by making the function Hfunc do an explicit matrix multiplication.
- Note that H has to be positive definite for the algorithm to work.
- In this algorithm, we predict the value that $Q(x)$ is going to have on the next iteration by examining its behavior in the subspace.
- On the next iteration the value of Q is recalculated at this point and it is a good check to see that the actual value agrees with the predicted value.

```
global H
% Set up a random positive definite matrix H and a right-ha
% of the equation
neq = 20;
H = rand(neq,neq);
H = H*H + 1e-6*eye(neq,neq);
G = randn(neq,1);
% Hmult is the name of a simple function that multiplies by
Hfunc = Hmult;
S = [];
% Random starting guess
x0 = randn(neq,1);
while 1,
Hx = feval(Hfunc,x0);
res = G - Hx;
fprintf(Norm of residual = %f\n,norm(res));
```

```
Qnow = x0*Hx - x0*G - G*x0;
S(:,1) = 2*res; % Search direction along negative gradient
[xnew,Qpred] = search1(x0,res,Hfunc,Qnow,S);
S(:,2) = xnew - x0; % Second search direction
x0 = xnew;
keyboard % Type "return" to continue to next iteration
end
```

- Notice how the matrix of search directions S is set up.
- On the first iteration, it consists of a single column containing $2(G - Hx_0)$.
- This is the negative gradient at the starting guess.
- After the first iteration, the second column of S is set to $x_1 - x_0$.
- On the second iteration, the first column is set to $2(G - Hx_1) \equiv -\nabla Q(x1)$ so that the function search1 finds the minimum in the two dimensional space spanned by $-\nabla Q(x_1)$ and $x_1 - x_0$, as required. This process continues on subsequent iterations.
- The function Hmult is simply

```
  function y = Hmult(x)
global H
y = H*x;
```

Application to Tikhonov Regularization

- For Tikhonov regularization, we need to find the minimum of the quadratic expression

$$\begin{aligned}\mathcal{L}(f) &= \lambda^2 \|L(f - f)\|^2 + \|d - Af\|^2 \\ &= \lambda^2 \Omega(f) + C(f)\end{aligned}$$

  where we shall assume that λ has been given.
- One can apply the algorithm discussed above directly to this problem by multiplying out the norms in order to find the matrices H and G, but it is convenient to use the special form of the expression and to assume that the user can provide functions which will apply the forward operator A and the operator L to any given vector.

- Starting from f_0 and searching within a subspace spanned by the columns of S as before, we wish to consider

$$\mathcal{L}(f_0 + Sc) = \lambda^2 \Omega(f_0 + Sc) + C(f_0 + Sc)$$

Substituting into the expression for , we find

$$\Omega(f0 + Sc) = \|L(f0 + Sc - f^\infty)\|^2$$

$$= (f_0 + Sc - f^\infty)^t L^t L(f_0 + Sc - f^\infty)$$

$$= c^t S^t L^t LSc + c^t StL^t L(f_0 - f^\infty) + (f_0 - f^\infty)^t L^t LSc + \Omega(f_0) (3.42)$$

$$= c^t \tilde{H}_\Omega c - c^t \tilde{G}_\Omega - \tilde{G}_\Omega^t c + \Omega(f_0)$$

- where

$$\tilde{H}_\Omega = S^t L^t L S = (LS)^t (LS)$$

$$\tilde{G}_\Omega = -S^t L^t L(f_0 - f^\infty) = -(LS)^t L(f_0 - f^\infty)$$

Similarly using the expression for C, we find

$$C(f_0 + Sc) = \|d - A(f_0 + Sc)\|^2$$

$$= d - A(f_0 + Sc)^t d - A(f_0 + Sc)$$

$$= c^t S^t A^t A S c - c^t S^t A^t (d - Af_0) - (d - Af_0)^t A S c + C(f_0)$$

$$= c^t H_C c - c^t G_C - \tilde{G}_C^t c + C(f_0)$$

- where

$$\tilde{H}_C = S^t A^t A S = (AS)^t (AS)$$

$$\tilde{G}_C = S^t A^t (d - Af_0) = (AS)^t r_0 \text{ where } r_0 = d - Af_0$$

- Thus

$$\mathcal{L}(f_0 + Sc) = c^t \left(\lambda^2 \tilde{H}_\Omega + \tilde{H}_C\right) - c^t \left(\lambda^2 \tilde{G}_\Omega + \tilde{G}_C\right) - (\lambda^2 \tilde{G}_\Omega + \tilde{G}_C)^t c + \mathcal{L}$$

- The minimum in the subspace occurs where

$$\tilde{c} = \left(\lambda^2 \tilde{H}_\Omega + \tilde{H}_C\right)^{-1} \left(\lambda^2 \tilde{G}_\Omega + \tilde{G}_C\right)$$

- and so we set

$$f_1 = f_0 + S(\lambda^2 \tilde{H}_\Omega + \tilde{H}_C)^{-1}(\lambda^2 \tilde{G}_\Omega + \tilde{G}_C)$$

- These considerations are illustrated in the Matlab program included below

```
function [fnew,cpred,wpred] =
 subsearch(f0,res,fdef,Afunc,Lfunc,lambda,S)
% Searches in subspace spanned by
% columns of S for the optimal solution
% of the regularization problem
% (lambda)^2*||L*(f-fdef)||^2 + ||d-A*f||^2
% f0 = Initial guess at location of the minimum
% res = d - A*f0, the current residual
% fdef = The default image
% Afunc = Name of function which applies A to a vector
% Lfunc = Name of function which applies L to a vector
% lambda = Weighting between solution seminorm and data mis
% S = matrix with search directions as its columns
%
% fnew = Position of minimum within subspace
% cpred = Predicted value of ||d-A*fnew||^2
```

```
% wpred = Predicted value of ||L*(fnew-fdef)||^2


  nsrch = size(S,2);
pref = feval(Lfunc,f0-fdef);
w0 = pref * pref;
c0 = res * res;
fprintf(Square of regularization semi-norm (current) = %f\n
fprintf(Square of data misfit norm (current) = %f\n,c0);
AS = zeros(length(res),nsrch);
LS = zeros(length(pref),nsrch);
for k = 1 : nsrch
AS(:,k) = feval(Afunc,S(:,k));
LS(:,k) = feval(Lfunc,S(:,k));
end
Hc = AS * AS; Hw = LS * LS;
Gc = AS * res; Gw = -LS*pref;
```

```
c = (Hc + lambda^2 * Hw) \ (Gc + lambda^2 *Gw);
cpred = c0 + c*Hc*c - c*Gc - Gc*c;
wpred = w0 + c*Hw*c - c*Gw - Gw*c;
fprintf(Square of regularization semi-norm (predicted) = %f
fprintf(Square of data misfit norm (predicted) = %f\n\n,cpr
fnew = f0 + S * c;
```

- Notice that for the minimization in the subspace, it is only necessary to be able to apply A and L to vectors.
- It only remains for us to calculate the search directions for the minimization.
- The negative gradient of $\mathcal{L}$ is

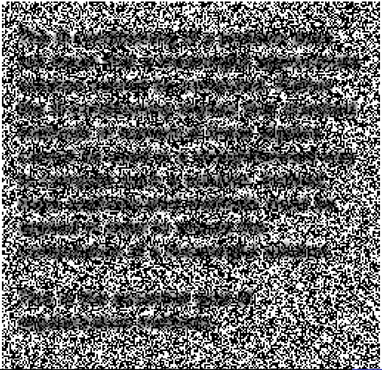
$$-\nabla\mathcal{L}(f) = -\lambda^2 L^t L(f - f^\infty) + A^t(d - Af)$$

- In order to calculate this, we also need functions which will apply L^t and A^t to a vector.
- A method which works reasonably well in practice is to calculate $L^t L(f_l - f), A^t(d - Af_l)$ as search directions on the first iteration ($l = 0$), and to append the direction $f_l - f_{l-1}$ on subsequent iterations.
- This is illustrated in the code fragment below:

```
S = [];
while 1,
pref = feval(Lfunc,f - fdef);
res = data - feval(Afunc,f);
S(:,1) = feval(Ahfunc,res); S(:,2) = -feval(Lhfunc,pref);
test = 1 - abs(S(:,1)*S(:,2)./(norm(S(:,1))*norm(S(:,2)));
fprintf(Test statistic = %f\n,test);
[fnew,cpred,spred] =
subsearch(f,res,fdef,Afunc,Lfunc,lambda,S);
S(:,3) = fnew - f;
f = fnew;
keyboard % Pause to allow user to view result
end
```

- In this example, the functions whose names are in Afunc and Lfunc apply the matrices A and L to a vector, while the functions whose names are in Ahfunc and Lhfunc apply the (conjugate) transposes (A^H and L^H) to a vector.
- The search directions are placed in the columns of S.
- The quantity test indicates whether the vectors ∇C and $\nabla\Omega$ are parallel to each other, as they should be at the optimal point.
- When the value of test is sufficiently small, the algorithm may be deemed to have converged.
- The figures below show a reconstruction based on the blurred image discussed in the first chapter, with the added complication that half the points of the blurred image are now assumed to be unmeasured or corrupted. In Figure, all unmeasured points are shown as black.
- The forward problem is as before, except that after the

Blurred image with missing data. Every black pixel indicates a data point that was unmeasured. In this image, approximately half the pixels are unmeasured.

- 

Reconstruction from the above data set

- The ill-conditioning of a problem does not mean that a meaningful approximate solution cannot be computed. Rather the ill-conditioning implies that standard methods in numerical linear algebra cannot be used in a straightforward way to compute such a solution. Instead, more sophisticated methods must be applied in order to ensure the computation of a meaningful solution.

  This is the essential goal of regularization methods.