Some Notes on Applied Mathematics for Machine Learning

Christopher J.C. Burges

Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA
cburges@microsoft.com
http://research.microsoft.com/~cburges

Abstract. This chapter describes Lagrange multipliers and some selected subtopics from matrix analysis from a machine learning perspective. The goal is to give a detailed description of a number of mathematical constructions that are widely used in applied machine learning.

1 Introduction

The topics discussed in this chapter are ones that I felt are often assumed in applied machine learning (and elsewhere), but that are seldom explained in detail. This work is aimed at the student who's taken some coursework in linear methods and analysis, but who'd like to see some of the tricks used by researchers discussed in a little more detail. The mathematics described here is a small fraction of that used in machine learning in general (a treatment of machine learning theory would include the mathematics underlying generalization error bounds, for example)[1], although it's a largely self-contained selection, in that derived results are often used downstream. I include two kinds of homework, 'exercises' and 'puzzles'. Exercises start out easy, and are otherwise as you'd expect; the puzzles are exercises with an added dose of mildly Machiavellian mischief.

Notation: vectors appear in bold font, and vector components and matrices in normal font, so that for example $v_i^{(a)}$ denotes the i'th component of the a'th vector $\mathbf{v}^{(a)}$. The symbol $A \succ 0$ ($\succeq$) means that the matrix A is positive (semi)definite. The transpose of the matrix A is denoted A^T, while that of the vector $\mathbf{x}$ is denoted $\mathbf{x}'$.

2 Lagrange Multipliers

Lagrange multipliers are a mathematical incarnation of one of the pillars of diplomacy (see the historical notes at the end of this section): sometimes an indirect approach will work beautifully when the direct approach fails.

[1] My original lectures also contained material on functional analysis and convex optimization, which is not included here.

O. Bousquet et al. (Eds.): Machine Learning 2003, LNAI 3176, pp. 21–40, 2004.

2.1 One Equality Constraint

Suppose that you wish to minimize some function $f(\mathbf{x})$, $\mathbf{x} \in \mathcal{R}^d$, subject to the constraint $c(\mathbf{x}) = 0$. A direct approach is to find a parameterization of the constraint such that f, expressed in terms of those parameters, becomes an unconstrained function. For example, if $c(\mathbf{x}) = \mathbf{x}'A\mathbf{x} - 1$, $\mathbf{x} \in \mathcal{R}^d$, and if $A \succ 0$, you could rotate to a coordinate system and rescale to diagonalize the constraints to the form $\mathbf{y}'\mathbf{y} = 1$, and then substitute with a parameterization that encodes the constraint that $\mathbf{y}$ lives on the $(d-1)$-sphere, for example

$$\begin{aligned} y_1 &= \sin\theta_1 \sin\theta_2 \cdots \sin\theta_{d-2} \sin\theta_{d-1} \\ y_2 &= \sin\theta_1 \sin\theta_2 \cdots \sin\theta_{d-2} \cos\theta_{d-1} \\ y_3 &= \sin\theta_1 \sin\theta_2 \cdots \cos\theta_{d-2} \\ &\cdots \end{aligned}$$

Unfortunately, for general constraints (for example, when c is a general polynomial in the d variables) this is not possible, and even when it is, the above example shows that things can get complicated quickly. The geometry of the general situation is shown schematically in Figure 1.

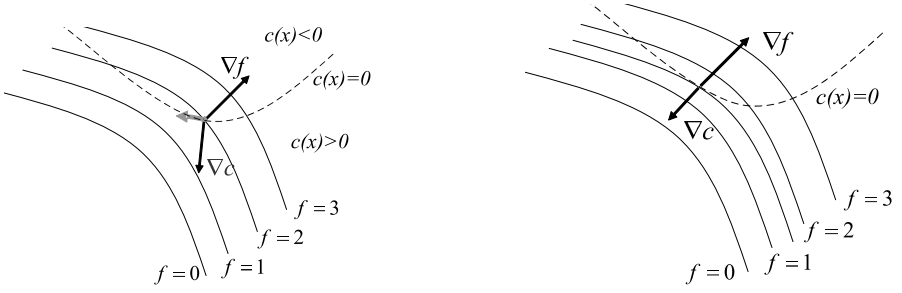

Fig. 1. At the constrained optimum, the gradient of the constraint must be parallel to that of the function

On the left, the gradient of the constraint is not parallel to that of the function; it's therefore possible to move along the constraint surface (thick arrow) so as to further reduce f. On the right, the two gradients are parallel, and any motion along $c(x) = 0$ will increase f, or leave it unchanged. Hence, at the solution, we must have $\nabla f = \lambda \nabla c$ for some constant λ; λ is called an *(undetermined) Lagrange multiplier*, where 'undetermined' arises from the fact that for some problems, the value of λ itself need never be computed.

2.2 Multiple Equality Constraints

How does this extend to multiple equality constraints, $c_i(x) = 0$, $i = 1, \ldots, n$? Let $\mathbf{g}_i \equiv \nabla c_i$. At any solution $\mathbf{x}_*$, it must be true that the gradient of f has no components that are perpendicular to all of the $\mathbf{g}_i$, because otherwise you could

move $\mathbf{x}_*$ a little in that direction (or in the opposite direction) to increase (decrease) f without changing any of the c_i, i.e. without violating any constraints. Hence for multiple equality constraints, it must be true that at the solution $\mathbf{x}_*$, the space spanned by the $\mathbf{g}_i$ contains the vector ∇f, i.e. there are some constants λ_i such that $\nabla f(\mathbf{x}_*) = \sum_i \lambda_i \mathbf{g}_i(\mathbf{x}_*)$. Note that this is not sufficient, however - we also need to impose that the solution is on the correct constraint surface (i.e. $c_i = 0 \; \forall i$). A neat way to encapsulate this is to introduce the Lagrangian $L \equiv f(\mathbf{x}) - \sum_i \lambda_i c_i(\mathbf{x})$, whose gradient with respect to the $\mathbf{x}$, and with respect to all the λ_i, vanishes at the solution.

Puzzle 1: *A single constraint gave us one Lagrangian; more constraints must give us more information about the solution; so why don't multiple constraints give us multiple Lagrangians?*

Exercise 1. *Suppose you are given a parallelogram whose side lengths you can choose but whose perimeter is fixed. What shaped parallelogram gives the largest area? (This is a case where the Lagrange multiplier can remain undetermined.) Now, your enterprising uncle has a business proposition: to provide cheap storage in floating containers that are moored at sea. He wants to build a given storage facility out of a fixed area of sheet metal which he can shape as necessary. He wants to keep construction simple and so desires that the facility be a closed parallelepiped (it has to be protected from the rain and from the occasional wave). What dimensions should you choose in order to maximize the weight that can be stored without sinking?*

Exercise 2. *Prove that the distance between two points that are constrained to lie on the n-sphere is extremized when they are either antipodal, or equal.*

2.3 Inequality Constraints

Suppose that instead of the constraint $c(\mathbf{x}) = 0$ we have the single constraint $c(\mathbf{x}) \leq 0$. Now the entire region labeled $c(\mathbf{x}) < 0$ in Figure 1 has become feasible. At the solution, if the constraint is active ($c(\mathbf{x}) = 0$), we again must have that ∇f is parallel to ∇c, by the same argument. In fact we have a stronger condition, namely that if the Lagrangian is written $L = f + \lambda c$, then since we are minimizing f, we must have $\lambda \geq 0$, since the two gradients must point in opposite directions (otherwise a move away from the surface $c = 0$ and into the feasible region would further reduce f). Thus for an inequality constraint, the sign of λ matters, and so here $\lambda \geq 0$ itself becomes a constraint (it's useful to remember that if you're minimizing, and you write your Lagrangian with the multiplier appearing with a positive coefficient, then the constraint is $\lambda \geq 0$). If the constraint is *not* active, then at the solution $\nabla f(\mathbf{x}_*) = 0$, and if $\nabla c(\mathbf{x}_*) \neq 0$, then in order that $\nabla L(\mathbf{x}_*) = 0$ we must set $\lambda = 0$ (and if in fact if $\nabla c(x_*) = 0$, we can still set $\lambda = 0$). Thus in either case (active or inactive), we can find the solution by requiring that the gradients of the Lagrangian vanish, and we also have $\lambda c(\mathbf{x}_*) = 0$. This latter condition is one of the important Karush-Kuhn-Tucker conditions of convex optimization theory [15, 4], and can facilitate the search for the solution, as the next exercise shows.

For multiple inequality constraints, again at the solution ∇f must lie in the space spanned by the ∇c_i, and again if the Lagrangian is $L = f + \sum_i \lambda_i c_i$, then we must in addition have $\lambda_i \geq 0 \ \forall i$ (since otherwise f could be reduced by moving into the feasible region); and for inactive constraints, again we (can, usually must, and so might as well) set $\lambda_i = 0$. Thus the above KKT condition generalizes to $\lambda_i c_i(\mathbf{x}_*) = 0 \ \forall i$. Finally, a simple and often useful trick is to solve ignoring one or more of the constraints, and then check that the solution satisfies those constraints, in which case you have solved the problem; we'll call this the *free constraint gambit* below.

Exercise 3. *Find the* $\mathbf{x} \in \mathcal{R}^d$ *that minimizes* $\sum_i x_i^2$ *subject to* $\sum_i x_i = 1$. *Find the* $\mathbf{x} \in \mathcal{R}^d$ *that maximizes* $\sum_i x_i^2$ *subject to* $\sum_i x_i = 1$ *and* $x_i \geq 0$ *(hint: use* $\lambda_i x_{*i} = 0$*).*

2.4 Cost Benefit Curves

Here's an example from channel coding. Suppose that you are in charge of four fiber optic communications systems. As you pump more bits down a given channel, the error rate increases for that channel, but this behavior is slightly different for each channel. Figure 2 show a graph of the bit rate for each channel versus the 'distortion' (error rate). Your goal is to send the maximum possible number of bits per second at a given, fixed total distortion rate D. Let D_i be the number

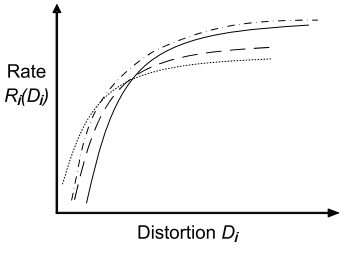

Fig. 2. Total bit rate versus distortion for each system

of errored bits sent down the i'th channel. Given a particular error rate, we'd like to find the maximum overall bit rate; that is, we must maximize the total rate $R \equiv \sum_{i=1} R_i$ subject to the constraint $D = \sum_{i=1} D_i$. Introducing a Lagrange multiplier λ, we wish to maximize the objective function

$$L = \sum_{i=1}^{4} R_i(D_i) + \lambda(D - \sum_{i=1}^{4} D_i) \tag{1}$$

Setting $\partial L / \partial D_i = 0$ gives $\partial R_i / \partial D_i = \lambda$, that is, each fiber should be operated at a point on its rate/distortion curve such that its slope is the same for all fibers. Thus we've found the general rule for resource allocation, for benefit/cost

curves like those shown[2] in Figure 2: whatever operating point is chosen for each system, in order to maximize the benefit at a given cost, the slope of the graph at that point should be the same for each curve. For the example shown, the slope of each graph decreases monotonically, and we can start by choosing a single large value of the slope λ for all curves, and decrease it until the condition $\sum_{i=1} D_i = D$ is met, so in general for m fibers, an m dimensional search problem has been reduced to a one dimensional search problem. We can get the same result informally as follows: suppose you had just two fibers, and were at an operating point where the slope s_1 of the rate/distortion graph for fiber 1 was greater than the slope s_2 for fiber 2. Suppose you then adjusted things so that fiber 1 sent one more errored bit every second, and fiber 2 sent one fewer. The extra number of bits you can now send down fiber 1 more than offsets the fewer number of bits you must send down fiber 2. This will hold whenever the slopes are different. For an arbitrary number of fibers, we can apply this argument to any pair of fibers, so the optimal point is for all fibers to be operating at the same slope.

Puzzle 2: *Suppose that instead of fibers, you have four factories making widgets, that the y-axis in Figure 2 represents the total cost for making n_i widgets, and that the x-axis represents the number n_i of widgets made by the i'th factory. The curves have the same shape (they drop off at larger n_i due to the economies of scale). Does the above argument mean that, to produce a total, fixed number of widgets, in order to minimize the cost, each factory should be operated at the same slope on its curve as all the other factories?*

2.5 An Isoperimetric Problem

Isoperimetric problems - problems for which a quantity is extremized while a perimeter is held fixed - were considered in ancient times, but serious work on them began only towards the end of the seventeenth century, with a minor battle between the Bernoulli brothers [14]. It is a fitting example for us, since the general isoperimetric problem had been discussed for fifty years before Lagrange solved it in his first venture into mathematics [1], and it provides an introduction to functional derivatives, which we'll need. Let's consider a classic isoperimetric problem: to find the plane figure with maximum area, given fixed perimeter. Consider a curve with fixed endpoints $\{x = 0, y = 0\}$ and $\{x = 1, y = 0\}$, and fixed length ρ. We will assume that the curve defines a function, that is, that for a given $x \in [0, 1]$, there corresponds just one y. We wish to maximize the area between the curve and the x axis, $A = \int_0^1 y dx$, subject to the constraint that the length, $\rho = \int_0^1 \sqrt{1 + y'^2} dx$, is fixed (here, prime denotes differentiation with respect to x). The Lagrangian is therefore

$$L = \int_0^1 y dx + \lambda \left(\int_0^1 \sqrt{1 + y'^2} dx - \rho \right) \tag{2}$$

[2] This seemingly innocuous statement is actually a hint for the puzzle that follows.

Two new properties of the problem appear here: first, integrals appear in the Lagrangian, and second, we are looking for a solution which is a function, not a point. To solve this we will use the calculus of variations, introduced by Lagrange and Euler. Denote a small variation of a function[3] f by δf: that is, replace $f(x)$ by $f(x) + \delta f(x)$ everywhere, where δf is chosen to vanish at the boundaries, that is, $\delta f(0) = \delta f(1) = 0$ (note that δf is also a function of x). Here, y is the variable function, so the change in L is

$$\delta L = \int_0^1 \delta y dx + \lambda \int_0^1 (1 + y'^2)^{-1/2} y' \delta y' dx$$

By using the facts that $\delta y' = \delta \frac{dy}{dx} = \frac{d}{dx} \delta y$ and that the variation in y vanishes at the endpoints, integrating by parts then gives:

$$\begin{aligned} \delta L &= \int_0^1 \left(1 - \lambda y''(1 + y'^2)^{-3/2}\right) \delta y dx \\ \Rightarrow \quad 1 \; &- \lambda y''(1 + y'^2)^{-3/2} \equiv 1 - \lambda\kappa = 0 \end{aligned}$$

where κ is the local curvature, and where the second step results from our being able to choose δy arbitrarily on $(0, 1)$, so the quantity multiplying δy in the integrand must vanish (imagine choosing δy to be zero everywhere except over an arbitrarily small interval around some point $x \in [0, 1]$). Since the only plane curves with constant curvature are the straight line and the arc of circle, we find the result (which holds even if the diameter of the circle is greater than one). Note that, as often happens in physical problems, λ here has a physical interpretation (as the inverse curvature); λ is always the ratio of the norms of ∇f and ∇c at the solution, and in this sense the size of λ measures the influence of the constraint on the solution.

2.6 Which Univariate Distribution has Maximum Entropy?

Here we use differential entropy, with the understanding that the bin width is sufficiently small that the usual sums can be approximated by integrals, but fixed, so that comparing the differential entropy of two distributions is equivalent to comparing their entropies. We wish to find the function f that minimizes

$$\int_{-\infty}^{\infty} f(x) \log_2 f(x) dx, \quad x \in \mathcal{R} \tag{3}$$

subject to the four constraints

$$f(x) \geq 0 \; \forall x, \quad \int_{-\infty}^{\infty} f(x) = 1, \quad \int_{-\infty}^{\infty} x f(x) = c_1 \quad \int_{-\infty}^{\infty} x^2 f(x) = c_2$$

[3] In fact Lagrange first suggested the use of the symbol δ to denote the variation of a whole function, rather than that at a point, in 1755 [14].

Note that the last two constraints, which specify the first and second moments, is equivalent to specifying the mean and variance. Our Lagrangian is therefore:

$$\mathcal{L} = \int_{-\infty}^{\infty} f(x) \log_2 f(x) dx + \lambda \left(1 - \int_{-\infty}^{\infty} f(x)\right) + \beta_1 \left(c_1 - \int_{-\infty}^{\infty} x f(x) dx\right) + \beta_2 \left(c_2 - \int_{-\infty}^{\infty} x^2 f(x) dx\right)$$

where we'll try the free constraint gambit and skip the positivity constraint. In this problem we again need the calculus of variations. In modern terms we use the *functional derivative*, which is just a shorthand for capturing the rules of the calculus of variations, one of which is:

$$\frac{\delta g(x)}{\delta g(y)} = \delta(x - y) \tag{4}$$

where the right hand side is the Dirac delta function. Taking the functional derivative of the Lagrangian with respect to $f(y)$ and integrating with respect to x then gives

$$\log_2 f(y) + \log_2(e) - \lambda - \beta_1 y - \beta_2 y^2 = 0 \tag{5}$$

which shows that f must have the functional form

$$f(y) = C \exp^{(\lambda + \beta_1 y + \beta_2 y^2)} \tag{6}$$

where C is a constant. The values for the Lagrange multipliers λ, β_1 and β_2 then follow from the three equality constraints above, giving the result that the Gaussian is the desired distribution. Finally, choosing $C > 0$ makes the result positive everywhere, so the free constraint gambit worked.

Puzzle 3: *For a given event space, say with N possible outcomes, the maximum entropy is attained when $p_i = 1/N \ \forall i$, that is, by the uniform distribution. That doesn't look very Gaussian. What gives?*

Exercise 4. *What distribution maximizes the entropy for the class of univariate distributions whose argument is assumed to be positive, if only the mean is fixed? How about univariate distributions whose argument is arbitrary, but which have specified, finite support, and where no constraints are imposed on the mean or the variance?*

Puzzle 4: *The differential entropy for a uniform distribution with support in $[-C, C]$ is*

$$\begin{aligned} h(P_U) &= -\int_{-C}^{C} (1/2C) \log_2(1/2C) dx \\ &= -\log_2(1/2C) \end{aligned} \tag{7}$$

This tends to ∞ as $C \to \infty$. How should we interpret this? Find the variance for any fixed C, and show that the univariate Gaussian with that variance has differential entropy greater than h.

2.7 Maximum Entropy with Linear Constraints

Suppose that you have a discrete probability distribution P_i, $\sum_i^n P_i = 1$, and suppose further that the only information that you have about the distribution is that it must satisfy a set of linear constraints:

$$\sum_i \alpha_{ji} P_i = C_j, \ j = 1, \ldots, m \tag{8}$$

The *maximum entropy* approach (see [5], for example) posits that, subject to the known constraints, our uncertainty about the set of events described by the distribution should be as large as possible, or specifically, that the mean number of bits required to describe an event generated from the constrained probability distribution be as large as possible. Maximum entropy provides a principled way to encode our uncertainty in a model, and it is the precursor to modern Bayesian techniques [13]. Since the mean number of bits is just the entropy of the distribution, we wish to find that distribution that maximizes[4]

$$-\sum_i P_i \log P_i + \sum_j \lambda_j (C_j - \sum_i \alpha_{ji} P_i) + \mu(\sum_i P_i - 1) - \sum_i \delta_i P_i \tag{9}$$

where the sum constraint on the P_i is imposed with μ, and the positivity of each P_i with δ_i (so $\delta_i \geq 0$ and at the maximum, $\delta_i P_i = 0 \ \forall i$)[5]. Differentiating with respect to P_k gives

$$P_k = \exp(-1 + \mu - \delta_k - \sum_j \lambda_j \alpha_{jk}) \tag{10}$$

Since this is guaranteed to be positive we have $\delta_k = 0 \ \forall k$. Imposing the sum constraint then gives $P_k = \frac{1}{Z} \exp(-\sum_j \lambda_j \alpha_{jk})$ where the "partition function" Z is just a normalizing factor. Note that the Lagrange multipliers have shown us the form that the solution must take, but that form does not automatically satisfy the constraints - they must still be imposed as a condition on the solution. The problem of maximizing the entropy subject to linear constraints therefore gives the widely used logistic regression model, where the parameters of the model are the Lagrange multipliers λ_i, which are themselves constrained by Eq. (8). For an example from the document classification task of how imposing linear constraints on the probabilities can arise in practice, see [16].

2.8 Some Algorithm Examples

Lagrange multipliers are ubiquitous for imposing constraints in algorithms. Here we list their use in a few modern machine learning algorithms; in all of these applications, the free constraint gambit proves useful. For support vector machines, the Lagrange multipliers have a physical force interpretation, and can be used to

[4] The factor $\log_2 e$ can be absorbed into the Lagrange multipliers.
[5] Actually the free constraint gambit would work here, too.

find the exact solution to the problem of separating points in a symmetric simplex in arbitrary dimensions [6]. For the remaining algorithms mentioned here, see [7] for details on the underlying mathematics. In showing that the principal PCA directions give minimal reconstruction error, one requires that the projection directions being sought after are orthogonal, and this can be imposed by introducing a matrix of multipliers. In locally linear embedding [17], the translation invariance constraint is imposed for each local patch by a multiplier, and the constraint that a solution matrix in the reconstruction algorithm be orthogonal is again imposed by a matrix of multipliers. In the Laplacian eigenmaps dimensional reduction algorithm [2], in order to prevent the collapse to trivial solutions, the dimension of the target space is enforced to be $d > 0$ by requiring that the rank of the projected data matrix be d, and again this imposed using a matrix of Lagrange multipliers.

Historical Notes. Joseph Louis Lagrange was born in 1736 in Turin. He was one of only two of eleven siblings to survive infancy; he spent most of his life in Turin, Berlin and Paris. He started teaching in Turin, where he organized a research society, and was apparently responsible for much fine mathematics that was published from that society under the names of other mathematicians [3, 1]. He '*believed that a mathematician has not thoroughly understood his own work till he has made it so clear that he can go out and explain it effectively to the first man he meets on the street*' [3][6]. His contributions lay in the subjects of mechanics, calculus[7], the calculus of variations[8], astronomy, probability, group theory, and number theory [14]. Lagrange is at least partly responsible for the choice of base 10 for the metric system, rather than 12. He was supported academically by Euler and d'Alembert, financed by Frederick and Louis XIV, and was close to Lavoisier (who saved him from being arrested and having his property confiscated, as a foreigner living in Paris during the Revolution), Marie Antoinette and the Abbé Marie. He survived the Revolution, although Lavoisier did not. His work continued to be fruitful until his death in 1813, in Paris.

3 Some Notes on Matrices

This section touches on some useful results in the theory of matrices that are rarely emphasized in coursework. For a complete treatment, see for example [12] and [11]. Following [12], the set of p by q matrices is denoted M_{pq}, the set of (square) p by p matrices by M_p, and the set of symmetric p by p matrices by S_p. We work only with real matrices - the generalization of the results to the complex field is straightforward. In this section only, we will use the notation in which repeated indices are assumed to be summed over, so that for example

[6] Sadly, at that time there were very few female mathematicians.

[7] For example he was the first to state Taylor's theorem with a remainder [14].

[8] ... with which he started his career, in a letter to Euler, who then generously delayed publication of some similar work so that Lagrange could have time to finish his work [1].

$A_{ij}B_{jk}C_{kl}$ is written as shorthand for $\sum_{j,k} A_{ij}B_{jk}C_{kl}$. Let's warm up with some basic facts.

3.1 A Dual Basis

Suppose you are given a basis of d orthonormal vectors $\mathbf{e}^{(a)} \in \mathcal{R}^d,\ a = 1, \ldots, d$, and you construct a matrix $E \in M_d$ whose columns are those vectors. It is a striking fact that the rows of E then also always form an orthonormal basis. We can see this as follows. Let the $\mathbf{e}^{(a)}$ have components $e_i^{(a)},\ i = 1, \ldots, d$. Let's write the vectors constructed from the rows of E as $\hat{\mathbf{e}}$ so that $\hat{\mathbf{e}}_i^{(a)} \equiv \mathbf{e}_a^{(i)}$. Then orthonormality of the columns can be encapsulated as $E^T E = \mathbf{1}$. However since E has full rank, it has an inverse, and $E^T E E^{-1} = E^{-1} = E^T$, so $EE^T = \mathbf{1}$ (using the fundamental fact that the left and right inverses of any square matrix are the same) which shows that the rows of E are also orthonormal. The vectors $\hat{\mathbf{e}}^{(a)}$ are called the dual basis to the $\mathbf{e}^{(a)}$. This result is sometimes useful in simplifying expressions: for example $\sum_a e_i^{(a)} e_j^{(a)} \Lambda(i,j)$, where Λ is some function, can be replaced by $\Lambda(i,i)\delta_{ij}$.

3.2 Other Ways to Think About Matrix Multiplication

Suppose you have matrices $X \in M_{mn}$ and $Y \in M_{np}$ so that $XY \in M_{mp}$. The familiar way to represent matrix multiplication is $(XY)_{ab} = \sum_{i=1}^n X_{ai} Y_{ib}$, where the summands are just products of numbers. However an alternative representation is $XY = \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i'$, where $\mathbf{x}_i$ ($\mathbf{y}_i'$) is the i'th column (row) of X (Y), and where the summands are outer products of matrices. For example, we can write the product of a 2×3 and a 3×2 matrix as

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} a \\ d \end{bmatrix} \begin{bmatrix} g & h \end{bmatrix} + \begin{bmatrix} b \\ e \end{bmatrix} \begin{bmatrix} i & j \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \begin{bmatrix} k & l \end{bmatrix}$$

One immediate consequence (which we'll use in our description of singular value decomposition below) is that you can always add columns at the right of X, and rows at the bottom of Y, and get the same product XY, provided either the extra columns, or the extra rows, contain only zeros. To see why this expansion works it's helpful to expand the outer products into standard matrix form: the matrix multiplication is just

$$\left\{ \begin{pmatrix} a & 0 & 0 \\ d & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & b & 0 \\ 0 & e & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & f \end{pmatrix} + \right\} \times \left\{ \begin{pmatrix} g & h \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ i & j \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ k & l \end{pmatrix} \right\}$$

Along a similar vein, the usual way to view matrix-vector multiplication is as an operation that maps a vector $\mathbf{z} \in \mathcal{R}^n$ to another vector $\mathbf{z}' \in \mathcal{R}^m$: $\mathbf{z}' = X\mathbf{z}$. However you can also view the product as a linear combination of the columns of X: $\mathbf{z}' = \sum_{i=1}^n z_i \mathbf{x}_i$. With this view it's easy to see why the result must lie in the span of the columns of X.

3.3 The Levi-Civita Symbol

The Levi-Civita symbol[9] in d dimensions is denoted $\epsilon_{ij\cdots k}$ and takes the value 1 if its d indices are an even permutation of $1, 2, 3, \cdots, d$, the value -1 if an odd permutation, and 0 otherwise. The 3-dimensional version of this is the fastest way I know to derive vector identities in three dimensions, using the identity $\epsilon_{ijk}\epsilon_{imn} = \delta_{jm}\delta_{kn} - \delta_{jn}\delta_{km}$ (recall that repeated indices are summed).

Exercise 5. *Use the fact that* $\mathbf{a} = \mathbf{b} \wedge \mathbf{c}$ *can be written in component form as* $a_i = \epsilon_{ijk}b_j c_k$ *to derive, in one satisfying line, the vector identity in three dimensions:* $(\mathbf{a} \wedge \mathbf{b}) \cdot (\mathbf{c} \wedge \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c})$.

3.4 Characterizing the Determinant and Inverse

The determinant of a matrix $A \in M_n$ can be defined as

$$|A| \equiv \frac{1}{n!}\epsilon_{\alpha_1\alpha_2\cdots\alpha_n}\epsilon_{\beta_1\beta_2\cdots\beta_n}A_{\alpha_1\beta_1}A_{\alpha_2\beta_2}\cdots A_{\alpha_n\beta_n} \tag{11}$$

Exercise 6. *Show that also,*

$$|A| = \epsilon_{\alpha_1\alpha_2\cdots\alpha_n}A_{1\alpha_1}A_{2\alpha_2}\cdots A_{n\alpha_n} \tag{12}$$

We can use this to prove an interesting theorem linking the determinant, derivatives, and the inverse:

Lemma 1. *For any square nonsingular matrix* A*,*

$$\frac{\partial |A|}{\partial A_{ij}} = A^{-1}_{ji} \tag{13}$$

Proof.

$$\frac{\partial |A|}{\partial A_{ij}} = \epsilon_{j\alpha_2\cdots\alpha_n}\delta_{i1}A_{2\alpha_2}\cdots A_{n\alpha_n} + \epsilon_{\alpha_1 j\cdots\alpha_n}A_{1\alpha_1}\delta_{i2}A_{3\alpha_3}\cdots A_{n\alpha_n} + \cdots$$

so

$$A_{kj}\frac{\partial |A|}{\partial A_{ij}} = \epsilon_{\alpha_1\alpha_2\cdots\alpha_n}(A_{k\alpha_1}\delta_{i1}A_{2\alpha_2}\cdots A_{n\alpha_n} + A_{1\alpha_1}A_{k\alpha_2}\delta_{i2}A_{3\alpha_3}\cdots + \cdots)$$

For any value of i, one and only one term in the sum on the right survives, and for that term, we must have $k = i$ by antisymmetry of the ϵ. Thus the right hand side is just $|A|\delta_{ki}$. Multiplying both sides on the right by $(A^T)^{-1}$ gives the result. □

[9] The name 'tensor' is sometimes incorrectly applied to arbitrary objects with more than one index. In factor a tensor is a generalization of the notion of a vector and is a geometrical object (has meaning independent of the choice of coordinate system); ϵ is a pseudo-tensor (transforms as a tensor, but changes sign upon inversion).

We can also use this to write the following closed form for the inverse:

$$A^{-1}_{ij} = \frac{1}{|A|(n-1)!}\epsilon_{j\alpha_1\alpha_2\cdots\alpha_{n-1}}\epsilon_{i\beta_1\beta_2\cdots\beta_{n-1}}A_{\alpha_1\beta_1}A_{\alpha_2\beta_2}\cdots A_{\alpha_{n-1}\beta_{n-1}} \quad (14)$$

Exercise 7. *Prove this, using Eqs. (11) and (13).*

Exercise 8. *Show that, for an arbitrary non-singular square matrix A, $\frac{\partial A^{-1}_{ij}}{\partial A_{\alpha\beta}} = -A^{-1}_{i\alpha}A^{-1}_{\beta j}$. (Hint: take derivatives of $A^{-1}A = \mathbf{1}$).*

Exercise 9. *The density $p(\mathbf{x})$ for a multivariate Gaussian is proportional to $|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$. For n independent and identically distributed points, the density is $p(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n|\boldsymbol{\mu}, \Sigma) = \prod_i p(\mathbf{x}_i|\boldsymbol{\mu}, \Sigma)$. By taking derivatives with respect to $\boldsymbol{\mu}$ and Σ and using the above results, show that the maximum likelihood values for the mean and covariance matrix are just their sample estimates.*

Puzzle 5: *Suppose that in Exercise 9, $n = 2$, and that $\mathbf{x}_1 = -\mathbf{x}_2$, so that the maximum likelihood estimate for the mean is zero. Suppose that Σ is chosen to have positive determinant but such that $\mathbf{x}$ is an eigenvector with negative eigenvalue. Then the likelihood can be made as large as you like by just scaling Σ with a positive scale factor, which appears to contradict the results of Exercise 9. What's going on?*

3.5 SVD in Seven Steps

Singular value decomposition is a generalization of eigenvalue decomposition. While eigenvalue decomposition applies only to square matrices, SVD applies to rectangular; and while not all square matrices are diagonalizable, every matrix has an SVD. SVD is perhaps less familiar, but it plays important roles in everything from theorem proving to algorithm design (for example, for a classic result on applying SVD to document categorization, see [10]). The key observation is that, given $A \in M_{mn}$, although we cannot perform an eigendecomposition of A, we can do so for the two matrices $AA^T \in S_m$ and $A^TA \in S_n$. Since both of these are positive semidefinite, their eigenvalues are non-negative; if AA^T has rank k, define the 'singular values' σ_i^2 to be its k positive eigenvalues. Below we will use 'nonzero eigenvector' to mean an eigenvector with nonzero eigenvalue, will denote the diagonal matrix whose i'th diagonal component is σ_i by $\mathrm{diag}(\sigma_i)$, and will assume without loss of generality that $m \leq n$. Note that we repeatedly use the tricks mentioned in Section (3.2). Let's derive the SVD.

1. *AA^T has the same nonzero eigenvalues as A^TA.* Let $\mathbf{x}_i \in \mathcal{R}^m$ be an eigenvector of AA^T with positive eigenvalue σ_i^2, and let $\mathbf{y}_i \equiv (1/\sigma_i)(A^T\mathbf{x}_i)$, $\mathbf{y} \in \mathcal{R}^n$. Then $A^TA\mathbf{y}_i = (1/\sigma_i)A^TAA^T\mathbf{x}_i = \sigma_iA^T\mathbf{x}_i = \sigma_i^2\mathbf{y}_i$. Similarly let $\mathbf{y}_i \in \mathcal{R}^n$ be an eigenvector of A^TA with eigenvalue $\sigma_i'^2$, and let $\mathbf{z}_i \equiv (1/\sigma_i')(A\mathbf{y}_i)$. Then $AA^T\mathbf{z}_i = (1/\sigma_i')AA^TA\mathbf{y}_i = \sigma_i'A\mathbf{y}_i = \sigma_i'^2\mathbf{z}_i$. Thus there is a 1-1 correspondence between nonzero eigenvectors for the matrices A^TA and AA^T, and the corresponding eigenvalues are shared.

2. *The* $\mathbf{x}_i$ *can be chosen to be orthonormal, in which case so also are the* $\mathbf{y}_i$. The $\mathbf{x}_i$ are orthonormal, or can be so chosen, since they are eigenvectors of a symmetric matrix. Then $\mathbf{y}_i \cdot \mathbf{y}_j \propto \mathbf{x}_i' A A^T \mathbf{x}_j \propto \mathbf{x}_i \cdot \mathbf{x}_j \propto \delta_{ij}$.
3. $\text{rank}(A) = \text{rank}(A^T) = \text{rank}(AA^T) = \text{rank}(A^T A) \equiv k$ [12].
4. Let the $\mathbf{x}_i$ be the nonzero eigenvectors of AA^T and the $\mathbf{y}_i$ those of $A^T A$. Let $X \in M_{mk}$ ($Y \in M_{nk}$) be the matrix whose columns are the $\mathbf{x}_i$ ($\mathbf{y}_i$). Then $Y = A^T X \text{diag}(1/\sigma_i) \Rightarrow \text{diag}(\sigma_i) Y^T = X^T A$. Note that $m \geq k$; if $m = k$, then $A = X \text{diag}(\sigma_i) Y^T$.
5. If $m > k$, add $m - k$ rows of orthonormal null vectors of A^T to the bottom of X^T, and add $m - k$ zero rows to the bottom of $\text{diag}(\sigma_i)$; defining the latter to be $\text{diag}(\sigma_i, 0)$, then X is orthogonal and $A = X \text{diag}(\sigma_i, 0) Y^T$. Note that here, $X \in M_m$, $\text{diag}(\sigma_i, 0) \in M_{mk}$ and $Y \in M_{nk}$.
6. To get something that looks more like an eigendecomposition, add $n - k$ rows of vectors that, together with the $\mathbf{y}_i$ form an orthonormal set, to the bottom of Y^T, and add $n - k$ columns of zeros to the right of $\text{diag}(\sigma_i, 0)$; defining the latter to be $\text{diag}(\sigma_i, 0, 0)$, then the Y are also orthogonal and $A = X \text{diag}(\sigma_i, 0, 0) Y^T$. Note that here, $X \in M_m$, $\text{diag}(\sigma_i, 0, 0) \in M_{mn}$, and $Y \in M_n$.
7. To get something that looks more like a sum of outer products, just write A in step (4) as $A = \sum_{i=1}^k \sigma_i \mathbf{x}_i \mathbf{y}_i'$.

Let's put the singular value decomposition to work.

3.6 The Moore-Penrose Generalized Inverse

Suppose $B \in S_m$ has eigendecomposition $B = E \Lambda E^T$, where Λ is diagonal and E is the orthogonal matrix of column eigenvectors. Suppose further that B is nonsingular, so that $B^{-1} = E \Lambda^{-1} E^T = \sum_i (1/\lambda_i) \mathbf{e}_i \mathbf{e}_i'$. This suggests that, since SVD generalizes eigendecomposition, perhaps we can also use SVD to generalize the notion of matrix inverse to non-square matrices $A \in M_{mn}$. The Moore-Penrose generalized inverse (often called just the generalized inverse) does exactly this[10]. In outer product form, it's the SVD analog of the ordinary inverse, with the latter written in terms of outer products of eigenvectors: $A^\dagger = \sum_{i=1}^k (1/\sigma_i) \mathbf{y}_i \mathbf{x}_i' \in M_{nm}$. The generalized inverse has several special properties:

1. $AA^\dagger$ *and* $A^\dagger A$ *are Hermitian;*
2. $AA^\dagger A = A$;
3. $A^\dagger A A^\dagger = A^\dagger$.

In fact, $A^\dagger$ is uniquely determined by conditions (1), (2) and (3). Also, if A is square and nonsingular, then $A^\dagger = A^{-1}$, and more generally, if $(A^T A)^{-1}$ exists, then $A^\dagger = (A^T A)^{-1} A^T$, and if $(AA^T)^{-1}$ exists, then $A^\dagger = A^T (AA^T)^{-1}$. The generalized inverse comes in handy, for example, in characterizing the general solution to linear equations, as we'll now see.

[10] The Moore-Penrose generalized inverse is one of many pseudo inverses.

3.7 SVD, Linear Maps, Range and Null Space

If $A \in M_{mn}$, the *range* of A, $\mathcal{R}(A)$, is defined as that subspace spanned by $\mathbf{y} = A\mathbf{x}$ for all $\mathbf{x} \in \mathcal{R}^n$. A's *null space* $\mathcal{N}(A)$, on the other hand, is that subspace spanned by those $\mathbf{x} \in \mathcal{R}^n$ for which $A\mathbf{x} = 0$. Letting $A_{|i}$ denote the columns of A, recall that $A\mathbf{x} = x_1 A_{|1} + x_2 A_{|2} + \cdots + x_n A_{|n}$, so that the dimension of $\mathcal{R}(A)$ is the rank k of A, and $\mathcal{R}(A)$ is spanned by the columns of A. Also, $\mathcal{N}(A^T)$ is spanned by those vectors which are orthogonal to every row of A^T (or every column of A), so $\mathcal{R}(A)$ is the orthogonal complement of $\mathcal{N}(A^T)$. The notions of range and null space are simply expressed in terms of the SVD, $A = \sum_{i=1}^{k} \sigma_i \mathbf{x}_i \mathbf{y}_i'$, $\mathbf{x} \in \mathcal{R}^m$, $\mathbf{y} \in \mathcal{R}^n$. The null space of A is the subspace orthogonal to the k $\mathbf{y}_i$, so $\dim(\mathcal{N}(A)) = n - k$. The range of A is spanned by the $\mathbf{x}_i$, so $\dim(\mathcal{R}(A)) = k$. Thus in particular, we have $\dim(\mathcal{R}(A)) + \dim(\mathcal{N}(A)) = n$.

The SVD provides a handy way to characterize the solutions to linear systems of equations. In general the system $A\mathbf{z} = \mathbf{b}$, $A \in M_{mn}$, $\mathbf{z} \in \mathcal{R}^n$, $\mathbf{b} \in \mathcal{R}^m$ has 0, 1 or ∞ solutions (if $\mathbf{z}_1$ and $\mathbf{z}_2$ are solutions, then so is $\alpha\mathbf{z}_1 + \beta\mathbf{z}_2$, $\alpha, \beta \in \mathcal{R}$). When does a solution exist? Since $A\mathbf{z}$ is a linear combination of the columns of A, $\mathbf{b}$ must lie in the span of those columns. In fact, if $\mathbf{b} \in \mathcal{R}(A)$, then $\mathbf{z}_0 = A^\dagger \mathbf{b}$ is a solution, since $A\mathbf{z}_0 = \sum_{i=1}^{k} \sigma_i \mathbf{x}_i \mathbf{y}_i' \sum_{j=1}^{k} (1/\sigma_i) \mathbf{y}_j \mathbf{x}_j' \mathbf{b} = \sum_{i=1}^{k} \mathbf{x}_i \mathbf{x}_i' \mathbf{b} = \mathbf{b}$, and the general solution is therefore $\mathbf{z} = A^\dagger \mathbf{b} + \mathcal{N}(A)$.

Puzzle 6: *How does this argument break down if* $\mathbf{b} \notin \mathcal{R}(A)$*?*

What if $\mathbf{b} \notin \mathcal{R}(A)$, i.e. $A\mathbf{z} = \mathbf{b}$ has no solution? One reasonable step would be to find that $\mathbf{z}$ that minimizes the Euclidean norm $\|A\mathbf{z} - \mathbf{b}\|$. However, adding any vector in $\mathcal{N}(A)$ to a solution $\mathbf{z}$ would also give a solution, so a reasonable second step is to require in addition that $\|\mathbf{z}\|$ is minimized. The general solution to this is again $\mathbf{z} = A^\dagger \mathbf{b}$. This is closely related to the following unconstrained quadratic programming problem: minimize $f(\mathbf{z}) = \frac{1}{2}\mathbf{z}' A\mathbf{z} + b\mathbf{z}$, $\mathbf{x} \in \mathcal{R}^n$, $A \succeq 0$. (We need the extra condition on A since otherwise f can be made arbitrarily negative). The solution to this is at $\nabla f = 0 \rightarrow A\mathbf{z} + \mathbf{b} = 0$, so the general solution is again $\mathbf{z} = A^\dagger \mathbf{b} + \mathcal{N}(A)$.

Puzzle 7: *If* $\mathbf{b} \notin \mathcal{R}(A)$*, there is again no solution, even though* $A \succeq 0$*. What happens if you go ahead and try to minimize* f *anyway?*

3.8 Matrix Norms

A function $\|\cdot\| : M_{mn} \rightarrow \mathcal{R}$ is a *matrix norm* over a field $\mathcal{F}$ if for all A, $B \in M_{mn}$,

1. $\|A\| \geq 0$
2. $\|A\| = 0 \Leftrightarrow A = 0$
3. $\|cA\| = |c|\|A\|$ for all scalars $c \in \mathcal{F}$
4. $\|A + B\| \leq \|A\| + \|B\|$

The Frobenius norm, $\|A\|_F = \sqrt{\sum_{ij} |A_{ij}|^2}$, is often used to represent the distance between matrices A and B as $\|A - B\|_F^2$, when for example one is searching for that matrix which is as close as possible to a given matrix, given

some constraints. For example, the closest positive semidefinite matrix, in Frobenius norm, to a given symmetric matrix A, is $\hat{A} \equiv \sum_{i:\lambda_i>0} \lambda_i \mathbf{e}^{(i)} \mathbf{e}'^{(i)}$ where the λ_i, $\mathbf{e}^{(i)}$ are the eigenvalues and eigenvectors of A, respectively. The Minkowski vector p-norm also has a matrix analog: $\|A\|_p \equiv \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|_p$. There are three interesting special cases of this which are easy to compute: the maximum absolute column norm, $\|A\|_1 \equiv \max_j \sum_i^n |A_{ij}|$, the maximum absolute row norm, $\|A\|_\infty \equiv \max_i \sum_j^n |A_{ij}|$, and the spectral norm, $\|A\|_2$. Both the Frobenius and spectral norms can be written in terms of the singular values: assuming the ordering $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_k$, then $\|A\|_2 = \sigma_1$ and $\|A\|_F = \sqrt{\sum_{i=1}^k \sigma_i^2}$.

Exercise 10. *Let U and W be orthogonal matrices. Show that $\|UAW\|_F = \|A\|_F$.*

Exercise 11. *The* **submultiplicative property**, *$\|AB\| \leq \|A\|\|B\|$, is an additional property that some matrix norms satisfy [11][11]. Prove that, if $A \in M_m$ and if a submultiplicative norm exists for which $\|A\| < 1$, then $(\mathbf{1}+A)^{-1} = \mathbf{1} - A + A^2 - A^3 + \cdots$, and if A is nonsingular and a submultiplicative norm exists for which $\|A^{-1}\| < 1$, then $(\mathbf{1}+A)^{-1} = A^{-1}(\mathbf{1} - A^{-1} + A^{-2} - A^{-3} + \cdots)$. Show that for any rectangular matrix W, $W(\mathbf{1}+W'W)^{-1}W' = (\mathbf{1}+WW')^{-1}WW' = WW'(\mathbf{1}+WW')^{-1}$. (This is used, for example, in the derivation of the conditional distribution of the latent variables given the observed variables, in probabilistic PCA [19].)*

The Minkowski p norm has the important property that $\|A\mathbf{x}\|_p \leq \|A\|_p \|\mathbf{x}\|_p$. Let's use this, and the L_1 and L_∞ matrix norms, to prove a basic fact about stochastic matrices. A matrix P is stochastic if its elements can be interpreted as probabilities, that is, if all elements are real and non-negative, and each row sums to one (row-stochastic), or each column sums to one (column-stochastic), or both (doubly stochastic).

Theorem 1. *If P is a square stochastic matrix, then P has eigenvalues whose absolute values lie in the range $[0, 1]$.*

Proof. For any $p \geq 1$, and $\mathbf{x}$ any eigenvector of P, $\|P\mathbf{x}\|_p = |\lambda| \|\mathbf{x}\|_p \leq \|P\|_p \|\mathbf{x}\|_p$ so $|\lambda| \leq \|P\|_p$. Suppose that P is row-stochastic; then choose the L_∞ norm, which is the maximum absolute row norm $\|P\|_\infty = \max_i \sum_j |P_{ij}| = 1$; so $|\lambda| \leq 1$. If P is column-stochastic, choosing the 1-norm (the maximum absolute column norm) gives the same result. □

Note that stochastic matrices, if not symmetric, can have complex eigenvalues, so in this case $\mathcal{F}$ is the field of complex numbers.

3.9 Positive Semidefinite Matrices

Positive semidefinite matrices are ubiquitous in machine learning theory and algorithms (for example, every kernel matrix is positive semidefinite, for Mercer

[11] Some authors include this in the definition of matrix norm [12].

kernels). Again we restrict ourselves to real matrices. A matrix $A \in S_n$ is positive definite iff for every $\mathbf{x} \in \mathcal{R}^n$, $\mathbf{x}'A\mathbf{x} > 0$; it is positive semidefinite iff for every $\mathbf{x} \in \mathcal{R}^n$, $\mathbf{x}'A\mathbf{x} \geq 0$, and some $\mathbf{x}$ exists for which the equality is met. Recall that we denote the property of positive definiteness of a matrix A by $A \succ 0$, and positive semidefiniteness by $A \succeq 0$. Let's start by listing a few properties, the first of which relate to what positive semidefinite matrices look like (here, repeated indices are not summed):

1. If $A \succ 0$, then $A_{ii} > 0 \ \forall i$;
2. If $A \succeq 0$, then $A_{ii} \geq 0 \ \forall i$;
3. If $A \succeq 0$, then $A_{ii} = 1 \ \forall i \Rightarrow |A_{ij}| \leq 1 \ \forall i, j$;
4. If $A \in S_n$ is strictly diagonally dominant, that is, $A_{ii} > \sum_{j \neq i} |A_{ij}| \ \forall i$, then it is also positive definite;
5. If $A \succeq 0$ and $A_{ii} = 0$ for some i, then $A_{ij} = A_{ji} = 0 \ \forall j$;
6. If $A \succeq 0$ then $A_{ii}A_{jj} \geq |A_{ij}|^2 \ \forall i, j$;
7. If $A \in S_n \succeq 0$ and $B \in S_n \succeq 0$ then $AB \succeq 0$;
8. $A \in S_n$ is positive semidefinite and of rank one iff $A = \mathbf{x}\mathbf{x}'$ for some $\mathbf{x} \in \mathcal{R}^n$;
9. $A \succ 0 \Leftrightarrow A$ all of the leading minors of A are positive.

A very useful way to think of positive semidefinite matrices is in terms of Gram matrices. Let V be a vector space over some field $\mathcal{F}$, with inner product $\langle \cdot, \cdot \rangle$. The *Gram matrix* G of a set of vectors $\mathbf{v}_i \in V$ is defined by $G_{ij} \equiv \langle \mathbf{v}_i, \mathbf{v}_j \rangle$. Now let V be Euclidean space and let $\mathcal{F}$ be the reals. The key result is the following: let $A \in S_n$. Then A is positive semidefinite with rank r if and only if there exists a set of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, $\mathbf{v}_i \in V$, containing exactly r linearly independent vectors, such that $A_{ij} = \mathbf{v}_i \cdot \mathbf{v}_j$.

Note in particular that the vectors $\mathbf{v}$ can always be chosen to have dimension $r \leq n$.

Puzzle 8: *A* **kernel matrix** $K \in S_n$ *is a matrix whose elements take the form* $K_{ij} \equiv k(\mathbf{x}_i, \mathbf{x}_j)$ *for some* $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^d, \ i, j = 1, \ldots, n$ *for some* d*, where* k *is a symmetric function which satisfies Mercer's condition (see e.g. [6]). For any such function* k*, there exists an inner product space* $\mathcal{H}$ *and a map* $\Phi : \mathcal{R}^d \mapsto \mathcal{H}$ *such that* $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$*. The dimension of* $\mathcal{H}$ *can be large, or even infinite (an example of the latter is* $k(\mathbf{x}_i, \mathbf{x}_j) = \exp^{-(1/\sigma^2)\|\mathbf{x}_i - \mathbf{x}_j\|^2}$*). In particular, the dimension of the dot product space can be larger than* n*. How does this square with the claim just made about the maximum necessary dimension of the Gram vectors?*

Some properties of positive semidefinite matrices that might otherwise seem mysterious become obvious, when they are viewed as Gram matrices, as I hope the following exercise helps demonstrate.

Exercise 12. *Use the fact that every positive semidefinite matrix is a Gram matrix to prove items (2), (3), (5), and (6) in the list above. Use the definition of a positive (semi)definite matrix to prove (1), (4), (7) and (8).*

If the Gram representation is so useful, the question naturally arises: given a positive semidefinite matrix, how can you extract a set of Gram vectors for

it? (Note that the set of Gram vectors is never unique; for example, globally rotating them gives the same matrix). Let $A \in S_n \succeq 0$ and write the eigendecomposition of A in outer product form: $A = \sum_{a=1}^{n} \lambda_a \mathbf{e}^{(a)} \mathbf{e}'^{(a)}$ or $A_{ij} = \sum_{a=1}^{n} \lambda_a e_i^{(a)} e_j^{(a)}$. Written in terms of the dual eigenvectors (see Section 3.1): $A_{ij} = \sum_{a=1}^{n} \lambda_a \hat{e}_a^{(i)} \hat{e}_a^{(j)}$, the summand has become a weighted dot product; we can therefore take the set of Gram vectors to be $v_a^{(i)} = \sqrt{\lambda_a} \hat{e}_a^{(i)}$. The Gram vectors therefore are the dual basis to the scaled eigenvectors.

3.10 Distance Matrices

One well-known use of the Gram vector decomposition of positive semidefinite matrices is the following. Define a 'distance matrix' to be any matrix of the form $D_{ij} \in S_n \equiv \|\mathbf{x}_i - \mathbf{x}_j\|^2$, where $\|\cdot\|$ is the Euclidean norm (note that the entries are actually squared distances). A central goal of multidimensional scaling is the following: given a matrix which is a distance matrix, or which is approximately a distance matrix, or which can be mapped to an approximate distance matrix, find the underlying vectors $\mathbf{x}_i \in \mathcal{R}^d$, where d is chosen to be as small as possible, given the constraint that the distance matrix reconstructed from the $\mathbf{x}_i$ approximates D with acceptable accuracy [8]. d is chosen to be small essentially to remove unimportant variance from the problem (or, if sufficiently small, for data visualization). Now let $\mathbf{e}$ be the column vector of n ones, and introduce the 'centering' projection matrix $P^e \equiv \mathbf{1} - \frac{1}{n}\mathbf{e}\mathbf{e}'$.

Exercise 13. *Prove the following: (1) for any $\mathbf{x} \in \mathcal{R}^n$, $P^e\mathbf{x}$ subtracts the mean value of the components of $\mathbf{x}$ from each component of $\mathbf{x}$, (2) $P^e\mathbf{e} = 0$, (3) $\mathbf{e}$ is the only eigenvector of P^e with eigenvalue zero, and (4) for any dot product matrix $A_{ij} \in S_m \equiv \mathbf{x}_i \cdot \mathbf{x}_j$, $i, j = 1, \ldots, m$, $\mathbf{x}_i \in \mathcal{R}^n$, then $(P^e A P^e)_{ij} = (\mathbf{x}_i - \boldsymbol{\mu}) \cdot (\mathbf{x}_j - \boldsymbol{\mu})$, where $\boldsymbol{\mu}$ is the mean of the $\mathbf{x}_i$.*

The earliest form of the following theorem is due to Schoenberg [18]. For a proof of this version, see [7].

Theorem 2. *Consider the class of symmetric matrices $A \in S_n$ such that $A_{ij} \geq 0$ and $A_{ii} = 0 \;\; \forall i, j$. Then $\bar{A} \equiv -P^e A P^e$ is positive semidefinite if and only if A is a distance matrix, with embedding space $\mathcal{R}^d$ for some d. Given that A is a distance matrix, the minimal embedding dimension d is the rank of $\bar{A}$, and the embedding vectors are any set of Gram vectors of $\bar{A}$, scaled by a factor of $\frac{1}{\sqrt{2}}$.*

3.11 Computing the Inverse of an Enlarged Matrix

We end our excursion with a look at a trick for efficiently computing inverses. Suppose you have a symmetric matrix $K \in S_{n-1}$, and suppose you form a new symmetric matrix by adding a number $u \equiv K_{nn}$ and a column $\mathbf{v}$, $v_i \equiv K_{in}$ (and a corresponding row $K_{ni} \equiv K_{in}$). Denote the enlarged matrix by

$$K_+ = \begin{pmatrix} K & \mathbf{v} \\ \mathbf{v}' & u \end{pmatrix} \tag{15}$$

Now consider the inverse

$$K_+^{-1} \equiv \begin{pmatrix} A & \mathbf{b} \\ \mathbf{b}' & c \end{pmatrix} \tag{16}$$

where again $\mathbf{b}$ is a column vector and c is a scalar. It turns out that it is straightforward to compute A, $\mathbf{b}$ and c in terms of K^{-1}, $\mathbf{v}$ and u. Why is this useful? In any machine learning algorithm where the dependence on all the data is captured by a symmetric matrix $K(\mathbf{x}_i, \mathbf{x}_j)$, then in test phase, when a prediction is being made for a single point $\mathbf{x}$, the dependence on all the data is captured by K_+, where $v_i = K(\mathbf{x}_i, \mathbf{x})$ and $u = K(\mathbf{x}, \mathbf{x})$. If that algorithm in addition requires that the quantities $\mathbf{b}$ and c be computed, it's much more efficient to compute them by using the following simple lemma (and computing K^{-1} just once, for the training data), rather than by computing K_+^{-1} for each $\mathbf{x}$. This is used, for example, in Gaussian process regression and Gaussian process classification, where in Gaussian process regression, c is needed to compute the variance in the estimate of the function value $f(\mathbf{x})$ at the test point $\mathbf{x}$, and $\mathbf{b}$ and c are needed to compute the mean of $f(\mathbf{x})$ [9, 20].

Lemma 2. *Given $K \in M_{n-1}$ and $K_+ \in M_n$ as defined above, then the elements of K_+ are given by:*

$$c = \frac{1}{u - \mathbf{v}' K^{-1} \mathbf{v}} \tag{17}$$

$$\mathbf{b} = -\frac{1}{u - \mathbf{v}' K^{-1} \mathbf{v}} \mathbf{v}' K^{-1} \tag{18}$$

$$A_{ij} = K_{ij}^{-1} + \frac{1}{u - \mathbf{v}' K^{-1} \mathbf{v}} (\mathbf{v}' K^{-1})_i (\mathbf{v}' K^{-1})_j \tag{19}$$

and furthermore,

$$\frac{\det(K)}{\det(K_+)} = \frac{1}{u - \mathbf{v}' K^{-1} \mathbf{v}} = c \tag{20}$$

Proof. Since the inverse of a symmetric matrix is symmetric, K_+^{-1} can be written in the form (16). Then requiring that $K_+^{-1} K_+ = \mathbf{1}$ gives (repeated indices are summed):

$$i < n,\ j < n: \qquad A_{im} K_{mj} + b_i v_j = \delta_{ij} \tag{21}$$

$$i = n,\ j < n: \qquad b_m K_{mj} + c v_j = 0 \tag{22}$$

$$i < n,\ j = n: \qquad A_{im} v_m + b_i u = 0 \tag{23}$$

$$i = n,\ j = n: \qquad b_m v_m + c u = 1 \tag{24}$$

Eq. (22) gives $b = -c\mathbf{v}' K^{-1}$. Substituting this in (24) gives Eq. (17), and substituting it in (21) gives Eq. (19). Finally the expression for the ratio of determinants follows from the expression for the elements of an inverse matrix in terms of ratios of its cofactors. □

Exercise 14. *Verify formulae (17), (18), (19) and (20) for a matrix $K_+ \in S_2$ of your choice.*

Puzzle 9: *Why not use this result iteratively (starting at $n = 2$) to compute the inverse of an arbitrary symmetric matrix $A \in S_n$? How does the number of operations needed to do this compare with the number of operations needed by Gaussian elimination (as a function of n)? If, due to numerical problems, the first (top left) element of the first matrix is off by a factor $1 + \epsilon$, $\epsilon \ll 1$, what is the error (roughly) in the estimated value of the final (bottom right) element of S_n?*

References

1. W.W. Rouse Ball. *A Short Account of the History of Mathematics*. Dover, 4 edition, 1908.
2. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
3. E.T. Bell. *Men of Mathematics*. Simon and Schuster, Touchstone edition, 1986; first published 1937.
4. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
5. B. Buck and V. Macaualay (editors). *Maximum Entropy in Action*. Clarendon Press, 1991.
6. C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
7. C.J.C. Burges. Geometric Methods for Feature Extraction and Dimensional Reduction. In L. Rokach and O. Maimon, editors, *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*. Kluwer Academic, 2004, to appear.
8. T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall, 2001.
9. Noel A.C. Cressie. *Statistics for spatial data*. Wiley, revised edition, 1993.
10. S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.
11. G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins, third edition, 1996.
12. R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
13. E.T. Jaynes. Bayesian methods: General background. In J.H. Justice, editor, *Maximum Entropy and Bayesian Methods in Applied Statistics*, pages 1–25. Cambridge University Press, 1985.
14. Morris Kline. *Mathematical Thought from Ancient to Modern Times, Vols. 1,2,3*. Oxford University Press, 1972.
15. O.L. Mangasarian. *Nonlinear Programming*. McGraw Hill, New York, 1969.
16. K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.
17. S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(22):2323–2326, 2000.

18. I.J. Schoenberg. Remarks to maurice frechet's article *sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert. Annals of Mathematics*, 36:724–732, 1935.
19. M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61(3):611, 1999.
20. C.K.I. Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 599–621. MIT Press, 1999.