High Dimensional Data

- So far we've considered scalar data values f_i (or interpolated/approximated each component of vector values individually)
- In many applications, data is itself in high dimensional space
  - Or there's no real distinction between dependent (f) and independent (x) -- we just have data points
- Assumption: data is actually organized along a smaller dimension manifold
  - generated from smaller set of parameters than number of output variables
- Huge topic: machine learning
- Simplest: Principal Components Analysis (PCA)

PCA

- We have n data points from m dimensions: store as columns of an mxn matrix A
- We're looking for linear correlations between dimensions
  - Roughly speaking, fitting lines or planes or hyperplanes through the origin to the data
  - May want to subtract off the mean value along each dimension for this to make sense

Reduction to 1D

- Assume data points fit through a line through the origin (1D subspace)
- In this case, say line is along unit vector u. (m-dimensional vector)
- Each data point should be a multiple of u (call the scalar multiples w_i):

$$A_{*i} = uw_i$$

- That is, A would be rank-1: $A=uw^T$
- Problem in general: find rank-1 matrix that best approximates A

The rank-1 problem

- Use Least-Squares formulation again:

$$\min_{\substack{u\in\mathbf{R}^m, \|u\|=1 \\ w\in\mathbf{R}^n}} \left\| A - uw^T \right\|_F^2$$

- Clean it up: take $w=\sigma v$ with $\sigma\geq 0$ and $|v|=1$

$$\min_{\substack{u\in\mathbf{R}^m, \|u\|=1 \\ v\in\mathbf{R}^n, \|v\|=1 \\ \sigma\geq 0}} \left\| A - u\sigma v^T \right\|_F^2$$

- u and v are the first principal components of A

Solving the rank-1 problem

- Remember trace version of Frobenius norm:

$$\begin{aligned}\left\|A-u\sigma v^T\right\|_F^2 &= \mathrm{tr}\left(A-u\sigma v^T\right)^T\left(A-u\sigma v^T\right)\\ &= \mathrm{tr}\left(A^TA\right)-\mathrm{tr}\left(A^Tu\sigma v^T\right)-\mathrm{tr}\left(v\sigma u^TA\right)+\mathrm{tr}\left(v\sigma u^Tu\sigma v^T\right)\\ &= \mathrm{tr}\left(A^TA\right)-2u^TAv\sigma+\sigma^2\end{aligned}$$

- Minimize with respect to sigma first:

$$\frac{\partial}{\partial\sigma}\left\|A-u\sigma v^T\right\|_F^2 = 0$$

$$-2u^TAv+2\sigma=0$$

$$\sigma=u^TAv$$

- Then plug in to get a problem for u and v:

$$\min -\left(u^TAv\right)^2 \quad \Leftrightarrow \quad \max\left(u^TAv\right)^2$$

Finding u

- First look at u: $$\begin{aligned}\left(u^TAv\right)^2 &= u^TAvv^TA^Tu\\ &= u^T\left(AA^T\right)u\end{aligned}$$
- AA^T is symmetric, thus has a complete set of orthonormal eigenvectors X, eigenvectors mu
- Write u in this basis: $u=\sum_{i=1}^m \hat{u}_iX_i$
- Then maximizing:

$$u^TAA^Tu = \left(\sum_{i=1}^m \hat{u}_iX_i\right)^T\left(\sum_{i=1}^m \mu_i\hat{u}_iX_i\right) = \sum_{i=1}^m \mu_i\hat{u}_i^2$$

- Obviously pick u to be the eigenvector with largest eigenvalue

Finding v

- Write the thing we're maximizing as:

$$\left(u^T A v\right)^2 = v^T A^T u u^T A v$$
$$= v^T \left(A^T A\right) v$$

- Same argument gives v the eigenvector corresponding to max eigenvalue of $A^T A$
- Note we also have

$$\sigma^2 = \left(u^T A v\right)^2 = \max \lambda\left(AA^T\right) = \max \lambda\left(A^T A\right) = \|A\|_2^2$$

Generalizing

- In general, if we expect problem to have subspace dimension k, we want the closest rank-k matrix to A
  - That is, express the data points as linear combinations of a set of k basis vectors (plus error)
  - We want the optimal set of basis vectors and the optimal linear combinations:

$$\min_{\substack{U \in \mathbf{R}^{m\times k}, U^T U = I \\ W \in \mathbf{R}^{n \times k}}} \left\| A - UW^T \right\|_F^2$$

Finding W

- Take the same approach as before:

$$\left\|A - UW^T\right\|_F^2 = \operatorname{tr}\left(A - UW^T\right)^T \left(A - UW^T\right)$$
$$= \operatorname{tr} A^T A - 2 \operatorname{tr} W U^T A + \operatorname{tr} W U^T U W^T$$
$$= \|A\|_F^2 - 2 \operatorname{tr} W U^T A + \|W\|_F^2$$

- Set gradient w.r.t. W equal to zero:

$$-2A^T U + 2W = 0$$
$$W = A^T U$$

Finding U

- Plugging in $W=A^T U$ we get

$$\min \left\|A - UW^T\right\|_F^2$$
$$\Leftrightarrow \min -2 \operatorname{tr} A^T U U^T A + \operatorname{tr} A^T U U^T A$$
$$\Leftrightarrow \max \operatorname{tr} U^T A A^T U$$

- AA^T is symmetric, hence has a complete set of orthogonormal eigenvectors, say columns of X, and eigenvalues along the diagonal of M (sorted in decreasing order):

$$AA^T = XMX^T$$

Finding U cont'd

- Our problem is now:

$$\max \operatorname{tr} U^T XMX^T U$$

- Note X and U are both orthogonal, so is $X^T U$, which we can call Z:

$$\max_{Z^T Z = I} \operatorname{tr} Z^T M Z$$
$$\Leftrightarrow \max_{Z^T Z = I} \sum_{i=1}^{k} \sum_{j=1}^{m} \mu_j Z_{ji}^2$$

- Simplest solution: set $Z=(I\ 0)^T$ which means that U is the first k columns of X (first k eigenvectors of AA^T)

Back to W

- We can write $W=V\Sigma^T$ for an orthogonal V, and square kxk Σ
- Same argument as for U gives that V should be the first k eigenvectors of $A^T A$
- What is Σ?
- From earlier rank-1 case we know

$$\Sigma_{11} = \sigma = \|A\|_2 = \left\|A^T\right\|_2$$

- Since U_{*1} and V_{*1} are unit vectors that achieve the 2-norm of A^T and A, we can derive that first row and column of Σ is zero except for diagonal.

What is Σ

- Subtract rank-1 matrix $U_{*1}\Sigma_{11}V_{*1}^T$ from A
  - zeros matching eigenvalue of A^TA or AA^T
- Then we can understand the next part of Σ
- End up with Σ a diagonal matrix, containing the squareroots of the first k eigenvalues of AA^T or A^TA (they're equal)

The Singular Value Decomposition

- Going all the way to k=m (or n) we get the Singular Value Decomposition (SVD) of A
- $A=U\Sigma V^T$
- The diagonal entries of Σ are called the singular values
- The columns of U (eigenvectors of AA^T) are the left singular vectors
- The columns of V (eigenvectors of A^TA) are the right singular vectors
- Gives a formula for A as a sum of rank-1 matrices: $A = \sum_i \sigma_i u_i v_i^T$

Cool things about the SVD

- 2-norm: $\|A\|_2 = \sigma_1$
- Frobenius norm: $\|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_n^2$
- Rank(A)= # nonzero singular values
  - Can make a sensible numerical estimate
- Null(A) spanned by columns of U for zero singular values
- Range(A) spanned by columns of V for nonzero singular values
- For invertible A: $A^{-1} = V\Sigma^{-1}U^T$

$$= \sum_{i=1}^{n} \frac{v_i u_i^T}{\sigma_i}$$

Least Squares with SVD

- Define pseudo-inverse for a general A:

$$A^+ = V\Sigma^+U^T = \sum_{\substack{i=1 \\ \sigma_i>0}}^{n} \frac{v_i u_i^T}{\sigma_i}$$

- Note if A^TA is invertible, $A^+=(A^TA)^{-1}A^T$
  - I.e. solves the least squares problem]
- If A^TA is singular, pseudo-inverse defined: A^+b is the x that minimizes $\|b-Ax\|_2$ and of all those that do so, has smallest $\|x\|_2$

Solving Eigenproblems

- Computing the SVD is another matter!
- We can get U and V by solving the **symmetric eigenproblem** for AA^T or A^TA, but more specialized methods are more accurate
- The **unsymmetric eigenproblem** is another related computation, with complications:
  - May involve complex numbers even if A is real
  - If A is not normal ($AA^T \neq A^TA$), it doesn't have a full basis of eigenvectors
  - Eigenvectors may not be orthogonal… Schur decomp
- Generalized problem: $Ax=\lambda Bx$
- LAPACK provides routines for all these
- We'll examine symmetric problem in more detail

The Symmetric Eigenproblem

- Assume A is symmetric and real
- Find orthogonal matrix V and diagonal matrix D s.t. AV=VD
  - Diagonal entries of D are the eigenvalues, corresponding columns of V are the eigenvectors
- Also put: $A=VDV^T$ or $V^TAV=D$
- There are a few strategies
  - More if you only care about a few eigenpairs, not the complete set…
- Also: finding eigenvalues of an nxn matrix is equivalent to solving a degree n polynomial
  - No "analytic" solution in general for n≥5
  - Thus general algorithms are **iterative**