UNIVERSITY OF OXFORD

Machine Learning
Nando de Freitas
January 22, 2015

Exercise Sheet 1

1 Probability revision 1: Student-t as an infinite mixture of Gaussians

Show that an infinite mixture of Gaussian distributions, with Gamma distributions as mixing weights in the following manner:

$$p(x) = \int_0^\infty \mathcal{N}(x|0, 1/\lambda)\text{Ga}(\lambda|\nu/2, \nu/2)d\lambda \tag{1}$$

is equal to a t-distribution:

$$p(x) = (\frac{2\pi\nu}{2})^{-\frac{1}{2}}\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\nu/2)}(1+\frac{x^2}{\nu})^{-\frac{\nu+1}{2}} = \mathcal{T}(x|0,1,\nu) \tag{2}$$

I recommend that you look at section 2.4.2 of Kevin's book for visualization of these distributions. You can also look them up in Wikipedia. **Hint:** Consider a change of variable $z = \lambda\Delta$.

This popular integration trick appears in the formulation of numerous statistical and neuroscience models, for example:

- Martin Wainwright and Eero Simoncelli. "Scale mixtures of Gaussians and the statistics of natural images." NIPS. 1999.
- Gavin Cawley, Nicola Talbot, and Mark Girolami. "Sparse multinomial logistic regression via Bayesian L1 regularisation." NIPS. 2007.

2 Reducing the cost of linear regression for large d small n

The ridge method is a regularized version of least squares, with objective function:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \delta^2\|\boldsymbol{\theta}\|_2^2.$$

Here, δ is a scalar, the input matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and the output vector $\mathbf{y} \in \mathbb{R}^n$. The parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$ is obtained by differentiating the above cost function, yielding the *normal equations*

$$(\mathbf{X}^T\mathbf{X} + \delta^2\mathbf{I}_d)\boldsymbol{\theta} = \mathbf{X}^T\mathbf{y},$$

where $\mathbf{I}_d$ is the $d\times d$ identity matrix. The predictions $\hat{\mathbf{y}} = \hat{\mathbf{y}}(\mathbf{X}_\star)$ for new test points $\mathbf{X}_\star \in \mathbb{R}^{n_\star\times d}$ are obtained by evaluating the hyperplane

$$\hat{\mathbf{y}} = \mathbf{X}_\star\boldsymbol{\theta} = \mathbf{X}_\star(\mathbf{X}^T\mathbf{X} + \delta^2\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y}. \tag{3}$$

The matrix $\mathbf{H}$ is known as the *hat matrix* because it puts a "hat" on $\mathbf{y}$.

1. Show that the solution can be written as $\boldsymbol{\theta} = \mathbf{X}^T\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = \delta^{-2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$.

2. Show that $\boldsymbol{\alpha}$ can also be written as follows: $\boldsymbol{\alpha} = (\mathbf{X}\mathbf{X}^T + \delta^2\mathbf{I}_n)^{-1}\mathbf{y}$ and, hence, the predictions can be written as follows:

$$\hat{\mathbf{y}} = \mathbf{X}_\star\boldsymbol{\theta} = \mathbf{X}_\star\mathbf{X}^T\boldsymbol{\alpha} = [\mathbf{X}_\star\mathbf{X}^T]([\mathbf{X}\mathbf{X}^T] + \delta^2\mathbf{I}_n)^{-1}\mathbf{y} \tag{4}$$

   (This is an *awesome trick* because if $n = 20$ patients with $d = 10,000$ gene measurements, the computation of $\boldsymbol{\alpha}$ only requires inverting the $n \times n$ matrix, while the direct computation of $\boldsymbol{\theta}$ would have required the inversion of a $d \times d$ matrix.)

3 Linear algebra revision: Eigen-decompositions

Once you start looking at raw data, one of the first things you notice is how redundant it often is. In images, it's often not necessary to keep track of the exact value of every pixel; in text, you don't always need the counts of every word. Correlations among variables also create redundancy. For example, if every time a gene, say A, is expressed another gene B is also expressed, then to build a tool that predicts patient recovery rate from gene expression data, it seems reasonable to remove either A or B. Most situations are not as clear-cut.

In this question, we'll look at eigenvalue methods for factoring and projecting data matrices (images, document collections, image collections), with an eye to one of the most common uses: Converting a high-dimensional data matrix to a lower-dimensional one, while minimizing the loss of information.

The Singular Value Decomposition (SVD) is a matrix factorization that has many applications in information retrieval, collaborative filtering, least-squares problems and image processing.

Let $\mathbf{X}$ be an $n \times n$ matrix of real numbers; that is $\mathbf{X} \in \mathbb{R}^{n\times n}$. Assume that $\mathbf{X}$ has n eigenvalue-eigenvector pairs $(\lambda_i, \mathbf{q}_i)$:

$$\mathbf{X}\mathbf{q}_i = \lambda_i\mathbf{q}_i \qquad i = 1, \ldots, n$$

If we place the eigenvalues $\lambda_i \in \mathbb{R}$ into a diagonal matrix $\boldsymbol{\Lambda}$ and gather the eigenvectors $\mathbf{q}_i \in \mathbb{R}^n$ into a matrix $\mathbf{Q}$, then the eigenvalue decomposition of $\mathbf{X}$ is given by

$$\mathbf{X}\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix}\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} \tag{5}$$

or, equivalently,

$$\mathbf{X} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1}.$$

For a symmetric matrix, i.e. $\mathbf{X} = \mathbf{X}^T$, one can show that $\mathbf{X} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$. But what if $\mathbf{X}$ is not a square matrix? Then the SVD comes to the rescue. Given $\mathbf{X} \in \mathbb{R}^{m\times n}$, the SVD of $\mathbf{X}$ is a factorization of the form

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T.$$

These matrices have some interesting properties:

- $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal with positive entries (singular values σ in the diagonal).
- $\mathbf{U} \in \mathbb{R}^{m \times n}$ has orthonormal columns: $\mathbf{u}_i^T \mathbf{u}_j = 1$ only when $i = j$ and 0 otherwise.
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ has orthonormal columns and rows. That is, $\mathbf{V}$ is an orthogonal matrix, so $\mathbf{V}^{-1} = \mathbf{V}^T$.

Often, $\mathbf{U}$ is m-by-m, not m-by-n. The extra columns are added by a process of orthogonalization. To ensure that dimensions still match, a block of zeros is added to $\mathbf{\Sigma}$. For our purposes, however, we will only consider the version where $\mathbf{U}$ is m-by-n, which is known as the *thin-SVD*.

It will turn out useful to introduce the vector notation:

$$\mathbf{X}\mathbf{v}_j = \sigma_j \mathbf{u}_j \qquad j = 1, 2, \ldots, n$$

where $\mathbf{u} \in \mathbb{R}^m$ are the left *singular vectors*, $\sigma \in [0, \infty)$ are the *singular values* and $\mathbf{v} \in \mathbb{R}^n$ are the right singular vectors. That is,

$$\mathbf{X} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} \tag{6}$$

or $\mathbf{XV} = \mathbf{U\Sigma}$. Note that there is no assumption that $m \geq n$ or that $\mathbf{X}$ has full rank. In addition, all diagonal elements of $\mathbf{\Sigma}$ are non-negative and in non-increasing order:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$$

where $p = \min(m, n)$.

Question: Outline a procedure for computing the SVD of a matrix $\mathbf{X}$. Hint: assume you can find the eigenvalue decompositions of the symmetric matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$.

4 Representing data with eigen-decompositions

It is instructive to think of the SVD as a sum of rank-one matrices:

$$\mathbf{X}_r = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T,$$

with $r \leq \min(m, n)$. What is so useful about this expansion is that the $k - th$ partial sum captures as much of the "energy" of $\mathbf{X}$ as possible. In this case, "energy" is defined by the *2-norm of a matrix*:

$$\|\mathbf{X}\|_2 = \sigma_1,$$

which can also be written as follows:

$$\|\mathbf{X}\|_2 = \max_{\|\mathbf{y}\|_2=1} \|\mathbf{X}\mathbf{y}\|_2,$$

This definition of the norm of a matrix builds upon our standard definition for the 2-norm of a vector $\|\mathbf{y}\|_2 = \sqrt{\mathbf{y}^T\mathbf{y}}$. Essensiatily, you can think of the norm of a matrix as how much it expands the unit-circle $\|\mathbf{y}\|_2 = 1$, when applied to it.

Question: Using the SVD, prove that the two definitions of the *2-norm of a matrix* are equivalent.

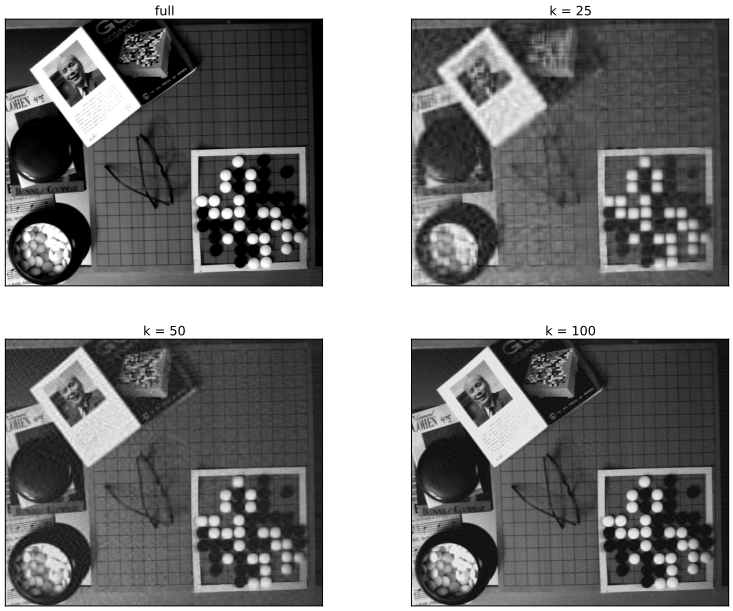

Figure 1: Image compression with SVD, showing reconstructions with different values of k.

If we only use $k \leq r$ terms in the expansion:

$$\mathbf{X}_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T,$$

a measure of the error in approximating $\mathbf{X}$ is:

$$\|\mathbf{X} - \mathbf{X}_k\|_2 = \sigma_{k+1}.$$

We will illustrate the value of this with an image compression example. Figure 1 shows the original image $\mathbf{X}$, and three SVD compressions $\mathbf{X}_k$. We can see that when k is low, we get a coarse version of the image. As more eigen-components are added, we get more high-frequency details – text becomes legible and pieces of the go game become circular instead of blocky.

The memory savings are dramatic as we only need to store the eigenvalues and eigen-vectors up to order k in order to generate the compressed image $\mathbf{X}_k$. Too see this, let the original image be of size 961 pixels by 1143 pixels, stored as a 961-by-1143 array of 32-bit floats. The total size of the data is over 17 MB (the number of entries in the array times 4 bytes per entry). Using SVD, we can compress and display the image to obtain the following numbers:

compression	**bytes**	**reduction**
original	17 574 768	0%
$k = 100$	842 000	95.2%
$k = 50$	?	97.6%
$k = 25$	?	98.8%

Question: Replace the question marks in the tables above with actual numbers of bytes.

Figure 2 shows another image example. The figure illustrates how the image of the clown can be reconstructed using the first rank-one components of the the SVD decomposition.

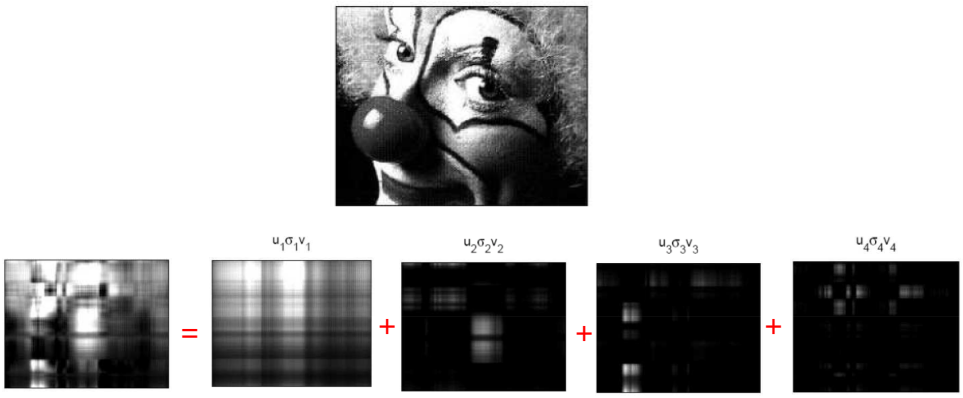

Figure 2: SVD reconstruction. Large eigenvalues are associated with low-frequencies and small eigenvalues with high-frequency components of the image.