Computational Intelligence Laboratory

Joachim M. Buhmann

Mario Frank
Morteza Haghir Chehreghani
Ekaterina Lomakina
Cheng Soon Ong
Patrick Pletscher
Sharon Wulff

http://ml2.inf.ethz.ch/courses/cil

February 21, 2011

Machine Learning Laboratory

You find us in the south wing of the CAB building, on F/G floor.

Table of Contents

Learning Goals

- Acquire fundamental theoretical concepts of unsupervised machine learning related to **matrix factorization**.
- Implement and compare several techniques to solve the same application problem:

Role Mining

Collaborative Filtering

Inpainting

Compression

- Combine and extend techniques to find a novel solution to an application problem, and compare it to baselines developed during the course.
- Learn how to write up your findings in the form of a scientific short paper.

Course Structure

Method–Application matrix structure: The methods are often useful for several applications when they are moderately adapted and you are invited to explore these options in a class project!

Application (columns) / *Theory* (rows)

	RBAC	Collaborative Filtering	Inpainting	Image Compression
Combinatorial Optimization	✓		✓	
Dimension Reduction		✓	✓	✓
Clustering	✓	✓	✓	✓
Generative Models	✓	✓	✓	
Sparse Coding			✓	✓
...				

Organizational Form

Weekly Schedule: 2 + 2 + 1

- *Lecture*: Tue 08-10, HG F 3
- *Exercises*: Thu 16-18, CAB G 52 x**or** Fri 08-10 CAB G 61
  - Both exercise sessions are identical
  - First hour: pen-and-paper exercise, immediate discussion
  - Second hour: group work on programming assignment
- Voluntary presence time: Mo 11-12, CAB H53
  - Assistants help with completing the programming assignments

Organizational Form

Programming Assignments

- Joint work in groups of two or three students
- Solving a problem by applying a technique from the lecture
- Submitting solution to the online evaluation and ranking system

First Week

The topic of the first exercise session on the 24^{th} and 25^{th} will be efficient matlab programming.

- Please bring laptops running Matlab with you
- "Hands on" exercises in groups of 2-3 students

Reading material

- Linear algebra background - on the course website

Grading Criteria

Semester Work

- Develop a novel solution for one of the application problems
- Comparison with two baseline techniques already implemented in the weekly programming assignments
- Write up in the form of a short paper
- **Competitive criteria:** speed, memory efficiency and accuracy
- **Non-competitive criteria:** paper review, creativity of solution, quality of implementation

Written Exam

- 120 minutes written exam
- 5 problems in the spirit of the pen-and-paper exercises

Grading Formula

You need to satisfy two **requirements** to pass this course:

1. Your average grade (60 % written exam, 40 % group project) is greater or equal to 4.
2. Your written exam grade is greater or equal to 3.5.

Therefore, your **final grade** will be:

1. Your average grade, if your written exam grade is greater or equal to 3.5.
2. Your written exam grade if it is below 3.5.

Matrix Factorization

Core problem: matrix factorization (MF) of a given data matrix into two or three unknown matrices, e.g.

$$\underset{D \times N}{\boxed{\mathbf{X}}} \approx \underset{D \times K}{\boxed{\mathbf{A}}} \cdot \underset{K \times N}{\boxed{\mathbf{B}}}$$

Many important data analysis techniques can be written as MF problems by specifying:

- **Type of data:** e.g. Boolean: $x_{d,n} \in \{0, 1\}$ or non-negative real: $x_{d,n} \in \mathbb{R}_+$, i.e., $x_{d,n} \geq 0$
- **Approximation:** e.g. exact: $\mathbf{X} = \mathbf{A} \cdot \mathbf{B}$ or minimal Frobenius norm: $\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{A} \cdot \mathbf{B}\|_F^2$
- **Constraints on unknown matrices:** e.g. $\mathbf{A}$ has to be a basis or $\mathbf{B}$ must be sparse.

Matrix Factorization

For many variants, MF is computationally NP-hard, therefore we need efficient and low-error approximation algorithms.

Techniques studied in this course:

1. Dimension reduction
2. Single-assignment clustering
3. Multi-assignment clustering
4. Sparse coding

Theory Part I: Dimension Reduction

Given N vectors in D dimensions, find the K most important axes to project them.

Why dimensionality reduction?

- Some features may be irrelevant.
- We want to visualize high dimensional data, e.g., in 2 or 3 dimensions.
- "Intrinsic" dimensionality may be smaller than the number of features recorded in the data.

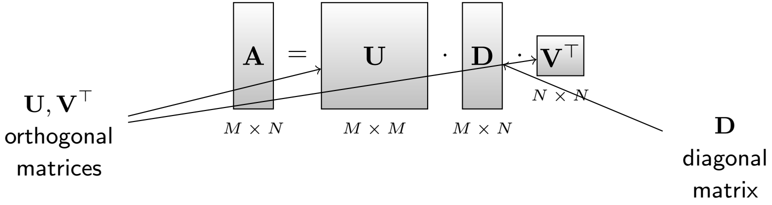
Many important dimensionality reduction techniques are based on the eigen-decomposition of the data matrix.

In this course we review the singular value decomposition (SVD) and principal component analysis (PCA).

Theory Part I: Dimension Reduction

SVD

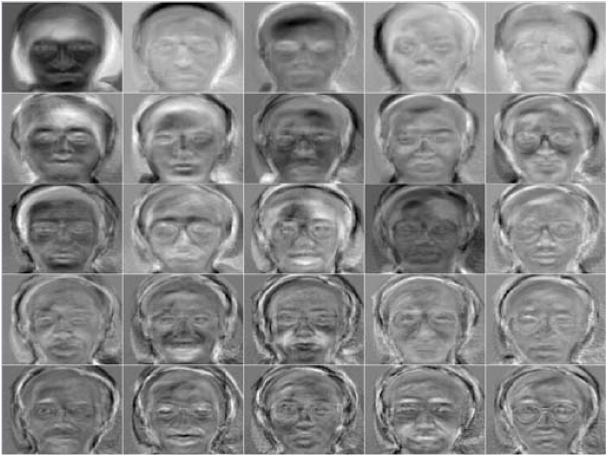
Any matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ can be decomposed uniquely as $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$:



K-closest approximation

Let $\mathbf{A}$ be a matrix of rank R, if we wish to approximate $\mathbf{A}$ using a matrix of a lower rank K then, $\tilde{\mathbf{A}} = \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}_k^\top$ is the closest matrix in the matrix L_2 norm.

Theory Part I: Dimension Reduction

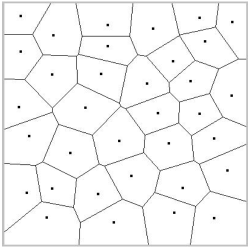


Christopher DeCoro http://www.cs.princeton.edu/~cdecoro/eigenfaces

"Eigenfaces": basis of vectors produced using the SVD of a set of training faces.

Theory Part II: Clustering

Quantization of the space $\mathbb{R}^D$



A data point $\mathbf{x} \in \mathbb{R}^D$ is represented by one of K centroids $\mathbf{u}_k \in \mathbb{R}^D,\ k = 1, \ldots, K$. We assign $\mathbf{x}$ to the centroid $\mathbf{u}_{k^*}$ that has minimal distance.

Theory Part II: Clustering

Figure: Top: original image, Bottom: same image where each pixel is represented by one of 10 colors.

Theory Part II: Clustering

K-means algorithm

Alternate between two steps until convergence:

1. Update assignments $z_{k,n}$ of data points to centroids:

$$z_{k,n} = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \|\mathbf{x}_n - \mathbf{u}_j\|_2 \\ 0 & \text{otherwise.} \end{cases}$$

2. Update centroid positions:

$$\mathbf{u}_k = \frac{\sum_n z_{k,n}\mathbf{x}_n}{\sum_n z_{k,n}}.$$

Theory Part II: Clustering

K-means clustering formulated as matrix factorization

Assign each datapoint (column of $\mathbf{X}$) to one of K clusters:

$$\underset{D \times N}{\boxed{\mathbf{X}}} \approx \underset{D \times K}{\boxed{\mathbf{U}}} \cdot \underset{K \times N}{\boxed{\mathbf{Z}}}$$

with $\mathbf{X} \in \mathbb{R}^{D \times N}$, $\mathbf{U} \in \mathbb{R}^{D \times K}$ and $\mathbf{Z} \in \{0, 1\}^{K \times N}$. In addition we enforce the unique assignment constraint $\sum_k z_{k,n} = 1 \ \forall n$.

The centroids define the columns of $\mathbf{U}$.

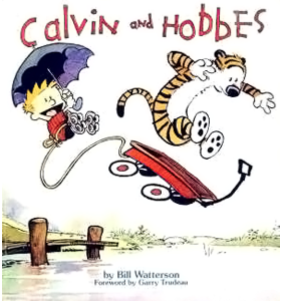
Theory Part III: Multi-Assignment Clustering

Cluster A: Schoolkids

- afraid of the teacher
- like to play tricks
- homework is just one way to spend the afternoon...

Cluster B: Comic Characters

- have less than 10 colors
- never get older
- speak in balloons
- make you smile



To which cluster does Calvin belong to?

- only A?
- only B?
- 37% A, 63% B?
- **both** A and B!

Theory Part III: Multi-Assignment Clustering

Key Features:

- Every data point can belong to one or more clusters.
- Each cluster is represented by one centroid.
- For every object, choose a set of clusters such that the representation is "closest".

Formulation as Matrix Factorization: Given a data matrix $\mathbf{X}$, find centroids $\mathbf{U}$ and assignments $\mathbf{Z}$ such that

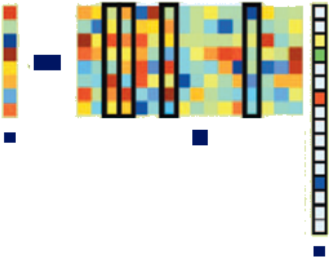
$$\mathbf{X} \approx \mathbf{U} \cdot \mathbf{Z} \ ,$$

with $\mathbf{X} \in \mathbb{R}^{D \times N}$, $\mathbf{U} \in \mathbb{R}^{D \times K}$ and $\mathbf{Z} \in \{0, 1\}^{K \times N}$

Now $\sum_k z_{k,n} > 1$ is admissible! (unlike in standard clustering)

Theory Part IV: Sparse Coding

Approximate representation of a signal $\mathbf{x} \in \mathbb{R}^D$ by a sparse linear combination $\mathbf{z} \in \mathbb{R}^L$ of K atoms of a dictionary $\mathbf{U} \in \mathbb{R}^{D \times L}$:



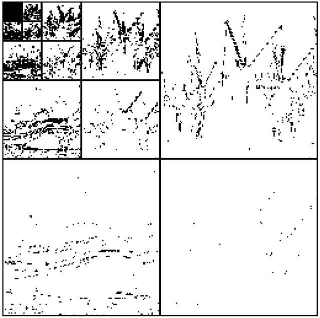
Formally, minimize the number of non-zero coefficients

$$\begin{aligned} &\min_{\mathbf{z}} \|\mathbf{z}\|_0 \\ \text{s.t.} \quad &\|\mathbf{U}\mathbf{z} - \mathbf{x}\|_2 < \sigma, \end{aligned}$$

while keeping the approximation error below σ.

Theory Part IV: Sparse Coding

Representation of grayscale image in a wavelet dictionary:



Only few coefficients are above a certain magnitude threshold (black dots in right figure). Efficient compression approach: keep large coefficients, disregard small ones.

Theory Part IV: Dictionary Learning as MF

Can we find even sparser representation? Yes, by adapting $\mathbf{U}$ to the image.

Dictionary learning: Find dictionary $\mathbf{U}$ and coding $\mathbf{Z}$ by solving

$$\min_{\mathbf{U},\mathbf{Z}} \|\mathbf{X} - \mathbf{U} \cdot \mathbf{Z}\|_F^2 \,,$$

where $\mathbf{Z}$ must be sparse.

- Problem can be made convex in $\mathbf{U}$ or $\mathbf{Z}$, but not in both arguments.
- Iterate between optimizing $\mathbf{U}$ and $\mathbf{Z}$.

Theory and Application

Group Project Example

	Role Mining	CF	Inpainting	Compression
Dim. Reduction		**o**		**x**
Clustering				**x**
MAC				
Sparse Coding				

Programming assignments: You implemented several solutions for each application, based on different techniques, e.g. image compression by dimension reduction and clustering (**x** in the matrix).

Group project: Develop a novel solution for one application, e.g. by transferring a technique to this new application domain (**o** in the matrix).

Lab Application I: Role Mining for RBAC

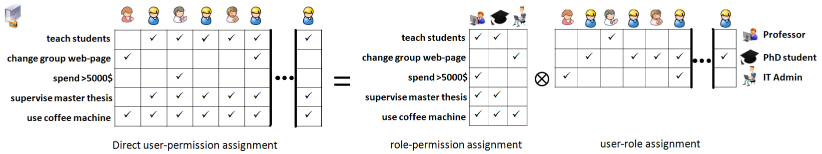
Access control can be managed by directly assigning users to their permissions (e.g. via Access Control Lists ACL's)

- Problem: 10^4 users and 10^3 permissions are hard to administrate this way!
- Preferred alternative: Role-Based Access Control (RBAC). Users are assigned to roles. Roles are sets of permissions.
- Migration step: Migrating a direct access control system to RBAC is non-trivial (has been identified as the costliest aspect of using RBAC). High demand for automated approaches.

Role Mining aims at finding and assigning roles based on a given user-permission matrix $\mathbf{X}$. The RBAC configuration should ideally capture the inherent structure of the access control system without exceptions and erroneous assignments.

Lab Application I: Role Mining for RBAC

A small example:



Note that users can have more than one role (i.e. PhD student and IT-coordinator). Technically, role mining is a matrix factorization problem or a multi-assignment clustering problem on data of Boolean nature.

Lab Application II: Collaborative Filtering

A recommender system is concerned with presenting items that are likely to interest the user.

- Books : Amazon
- Movies : MovieLens, IMDB
- Music : LastFM

In collaborative filtering (CF), we base our recommendations:

- On the (known) preference of the user towards other items,
- Take into account the preferences of other users.

Lab Application II: Collaborative Filtering

Viewers were asked to rate the some movies (items):

	Ben	Tom	John	Fred	Jack
Star Wars	?	?	1	?	4
WallE	5	?	3	4	?
Avatar	3	4	?	4	4
Trainspotting	?	1	5	?	?
Shrek	5	?	?	5	?
Ice Age	5	?	4	?	1

- Not all viewers rated all movies.
- Unrated user-movie pairs are missing values: we want to predict them.
- Should we recommend Fred to watch "Ice Age"?

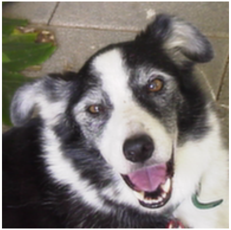
Lab Application III: Inpainting

Inpainting seeks the restoration of missing pixels, based on the intact surroundings:

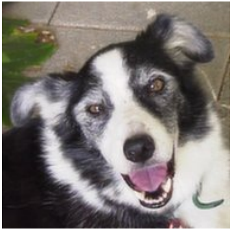
As with collaborative filtering, the missing values are predicted based on the statistics of known pixels of the target image, and possibly other images, too.

Lab Application IV: Compression

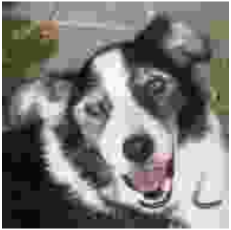
- Encode information using fewer bits than an unencoded representation.
- Lossy compression: compressed data is different from the original, but is close enough to be useful in some way.
- Achieve this by neglecting irrelevant part of information.
- Here: Compression of images



Original (61 KB)



Low Compression (10 KB)



High Compression (2 KB)

Course Structure

Methods vs. Applications: Stimulate your fantasy to fill in one of the empty checkmark boxes.

Application

Theory

	RBAC	Collaborative Filtering	Inpainting	Image Compression
Combinatorial Optimization	✓		✓	
Dimension Reduction		✓	✓	✓
Clustering	✓	✓	✓	✓
Generative Models	✓	✓	✓	
Sparse Coding			✓	✓
...				

Table of Contents

Notation

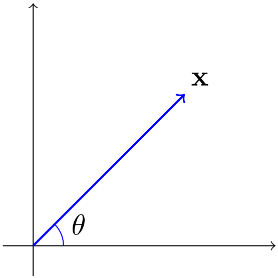
- $\mathbf{x}$ is a column vector
- $^\top$ denotes the transpose operator, so $\mathbf{x}^\top$ is a row vector
- The elements of a vector are denoted as $\mathbf{x} = (x_1, x_2, \ldots, x_D)^\top$
- $\mathbf{U}$ is a matrix:
  - $\mathbf{u}_k$ or $\mathbf{u}_{\cdot,k}$ is the k-th column of $\mathbf{U}$
  - $\mathbf{u}_d^\top$ or $\mathbf{u}_{d,\cdot}$ is the d-th row of $\mathbf{U}$
  - $u_{d,k}$ is the element in the d-th row and k-th column of $\mathbf{U}$

Vectors

- n-th data sample $\mathbf{x}_n$ is a D-dimensional vector containing D features:

$$\mathbf{x}_n \in \mathbb{R}^{D \times 1}$$

- Vectors are basic geometric entities that allow you to talk about magnitude and direction.



Matrices

Matrices:

- Represent a dataset: N datapoints $\mathbf{x}_1, ..., \mathbf{x}_N$ in a D-dimensional space $\mathbb{R}^D$:

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = \begin{pmatrix} x_{1,1} & \dots & x_{1,N} \\ \vdots & & \vdots \\ x_{D,1} & \dots & x_{D,N} \end{pmatrix}$$

- Matrices can also be used to apply geometric transformations (rotations, scaling) to vectors.
- In particular: express dataset in a different coordinate system (using projections).

Example

Viewers were asked to rate the some movies (items):

	Ben	Tom	John	Fred	Jack
Star Wars	0	0	1	0	4
WallE	5	0	3	4	0
Avatar	3	4	0	4	4
Trainspotting	0	1	5	0	0
Shrek	5	0	0	5	0
Ice Age	5	0	4	0	1

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 1 & 0 & 4 \\ 5 & 0 & 3 & 4 & 0 \\ 3 & 4 & 0 & 4 & 4 \\ 0 & 1 & 5 & 0 & 0 \\ 5 & 0 & 0 & 5 & 0 \\ 5 & 0 & 4 & 0 & 1 \end{pmatrix}$$

$\mathbf{X}$: Users are the datasamples, movies are the features.
$\mathbf{X}^\top$: Movies are the datasamples, user ratings are the features.

Matrix Transpose

Transpose: reflect vector/matrix on a diagonal line

Two examples:

$$\begin{pmatrix} a \\ b \end{pmatrix}^\top = (a\ b) \qquad \begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}^\top = \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix}$$

Note that $(\mathbf{A}\mathbf{x})^\top = \mathbf{x}^\top \mathbf{A}^\top$.

Special Matrices

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{I}$$ - identity matrix

$$\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$$ diagonal matrix

$$\begin{pmatrix} a & d & f \\ 0 & b & e \\ 0 & 0 & c \end{pmatrix}$$ upper triangular

$$\begin{pmatrix} a & 0 & 0 \\ d & b & 0 \\ f & e & c \end{pmatrix}$$ lower triangular

$$\begin{pmatrix} a & d & f \\ d & b & e \\ f & e & c \end{pmatrix}$$ symmetric: $\mathbf{A}^\top = \mathbf{A}$

Scalar Product

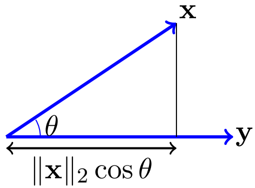
Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{D \times 1}$:

$$\begin{aligned} < \mathbf{x}, \mathbf{y} > \;=\; \mathbf{x}^\top \mathbf{y} &= (x_1, x_2, \dots, x_D) \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_D \end{pmatrix} \\ &= x_1 y_1 + x_2 y_2 + \dots x_D y_D \end{aligned}$$

- $\mathbf{x}^\top \mathbf{x}$ gives the squared Euclidean length of $\mathbf{x}$.
- L_2 Norm: $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$
- Generally: $\mathbf{x}^\top \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos\theta$

Scalar Product

- If $\|\mathbf{y}\|_2 = 1$ (unit vector), then $\mathbf{x}^\top \mathbf{y} = \|\mathbf{x}\|_2 \cos\theta$ gives the magnitude of the projection of $\mathbf{x}$ onto $\mathbf{y}$. This is the scalar component of $\mathbf{x}$ in the direction of $\mathbf{y}$:



Question: When is $\mathbf{x}^\top \mathbf{y} = 0$?

Vector Matrix Multiplications

Let $\mathbf{x}$ be a D-dimensional column vector and $\mathbf{A}$ be a $M \times D$ matrix. Then $\mathbf{b} = \mathbf{A}\mathbf{x}$ is an M vector defined as

$$b_m = \sum_{d=1}^{D} a_{m,d} x_d, \; m = 1, \ldots, M.$$

We interpret this by saying that $\mathbf{A}$ acts on $\mathbf{x}$ to produce $\mathbf{b}$.

This product can be rewritten as $\mathbf{b} = \mathbf{A}\mathbf{x} = \sum_{d=1}^{D} x_d \mathbf{a}_d$ and visualized by

$$\begin{bmatrix} \\ b \\ \\ \end{bmatrix} = \left[\begin{array}{c|c|c|c} \\ a_1 & a_2 & \cdots & a_n \\ \\ \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} \\ a_1 \\ \\ \end{bmatrix} + x_2 \begin{bmatrix} \\ a_2 \\ \\ \end{bmatrix} + \cdots + x_n \begin{bmatrix} \\ a_n \\ \\ \end{bmatrix}.$$

Here, the interpretation is that $\mathbf{x}$ acts on $\mathbf{A}$ to produce $\mathbf{b}$.

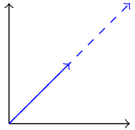
Matrices As Linear Transformations

We can transfrom $\mathbf{x}$ into $\mathbf{y} \in \mathbb{R}^{M\times 1}$ using a transformation matrix $\mathbf{A} \in \mathbb{R}^{M\times D}$

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

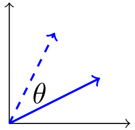
Some special cases (2-D):

$$\mathbf{A} = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix}$$



scaling in all directions by α

$$\mathbf{A} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$



counterclockwise rotation by θ

Vector Spaces

- Formally, a vector space is a set of vectors which is closed under addition and multiplication by real numbers.
- A subspace is a subset of a vector space which is a vector space itself, e.g. the plane $z = 0$ is a subspace of $\mathbb{R}^3$ (It is essentially $\mathbb{R}^2$).
- Note: subspaces must include the origin (zero vector).

Vector Spaces

$$\begin{pmatrix} 1 & 0 \\ 2 & 3 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$x_1 \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

- Linear systems define certain subspaces
- $\mathbf{Ax} = \mathbf{b}$ is solvable iff $\mathbf{b}$ can be written as a linear combination of the columns of $\mathbf{A}$.
- The set of possible vectors $\mathbf{b}$ forms a subspace called the column space of $\mathbf{A}$.

Range and Nullspace

Every linear mapping from $\mathbb{R}^D$ to $\mathbb{R}^M$ can be represented by an appropriate $\mathbf{A} \in \mathbb{R}^{M \times D}$.

- range$(\mathbf{A})$ is the set of vectors $\{\mathbf{b} \,|\, \exists \mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$.
  Furthermore, range$(\mathbf{A})$ is the space spanned by the columns of $\mathbf{A}$.

- null$(\mathbf{A})$ is the set of vectors $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{0}\}$.
  The entries of each $\mathbf{x} \in$ null$(\mathbf{A})$ form an expansion of zero as a linear combination of the columns of $\mathbf{A}$:

$$\mathbf{0} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_D\mathbf{a}_D.$$

Nullspace

$$\begin{pmatrix} 1 & 0 \\ 2 & 3 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad \Longrightarrow \text{null space: } \{(0,0)\}$$

$$\begin{pmatrix} 1 & 0 & 1 \\ 2 & 3 & 4 \\ 1 & 3 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Longrightarrow \text{null space: } \{(c, \frac{2}{3}c, -c)\}$$

Linear Independence

The vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are called linearly independent if

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_n\mathbf{v}_n = 0 \quad \text{implies}$$
$$c_1 = c_2 = \ldots = c_n = 0$$

Note that in this case the nullspace of the matrix is the origin.

$$\begin{pmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

In the previous example, the nullspace contained only $(0,0)$ therefore the columns of the matrix are linearly independent

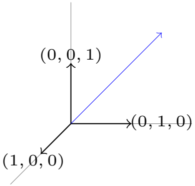
$$\begin{pmatrix} 1 & 0 \\ 2 & 3 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

null space: $\{(0,0)\}$

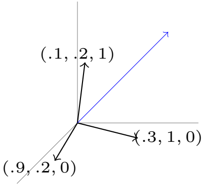
Spanning the Space

If all vectors in a vector space can be expressed as linear combinations of the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, then $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ span the space.

$$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



$$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} = 1.57 \begin{pmatrix} .9 \\ .2 \\ 0 \end{pmatrix} + 1.29 \begin{pmatrix} .3 \\ 1 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} .1 \\ .2 \\ 1 \end{pmatrix}$$



Basis

A basis B of a vector space V is a **linearly independent** subset of V that **spans** V.

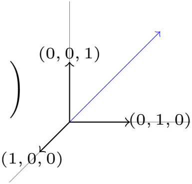
- The dimension of a space is the number of vectors in any basis for the space.
- There can be multiple different bases for the space.
- A basis is a maximal set of linearly independent vectors and a minimal set of spanning vectors.

Basis Transformation

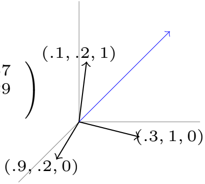
We may write $\mathbf{x} = (2, 2, 2)$ in terms of an alternate basis:

$$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$



$$\begin{pmatrix} .9 & .3 & .1 \\ .2 & 1 & .2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1.57 \\ 1.29 \\ 2 \end{pmatrix}$$



* The components of $(1.57, 1.29, 2)$ are projections of $\mathbf{x}$ onto the new vectors basis, normalized such that the new $\mathbf{x}$ still has the same length.

Exercise

Consider:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

1. What is the range?
2. What is the rank?
3. What is the null space?

Theorem

Fundamental Theorem of Linear Algebra:

If $\mathbf{A}$ is $M \times N$ with rank R, then

- range($\mathbf{A}$) has dimension R
- null($\mathbf{A}$) has dimension $N - R$ (= nullity of $\mathbf{A}$)
- row space of $\mathbf{A}$ = range($\mathbf{A}^\top$), has dimension R

Rank-Nullity Theorem: rank + nullity = N

Inverse

A nonsingular or invertible matrix is a square $M \times M$ matrix of full rank. Its columns form a basis for the whole space $\mathbb{R}^M$.

Any vector can be uniquely expressed as a linear combination of the basis vectors.

In particular, the canonical unit vectors $\mathbf{e}_m,\ m = 1, \ldots, M$ can be expanded as

$$\mathbf{e}_j = \sum_{m=1}^{M} z_{m,j} \mathbf{a}_m.$$

This fact can be compactly written as

$$\left[\begin{array}{c|c|c} \mathbf{e}_1 & \cdots & \mathbf{e}_M \end{array} \right] = \mathbf{I} = \left[\begin{array}{c|c|c} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{array} \right] \mathbf{Z},$$

where $\mathbf{Z}$ is the matrix with entries $z_{m,j}$. $\mathbf{Z}$ is the inverse $\mathbf{A}^{-1}$ of $\mathbf{A}$, satisfying $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$.

Eigenvalues and Eigenvectors

Given a square matrix $\mathbf{A}$,

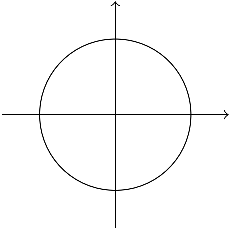
- The set of solutions to $\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$ is in the form of eigen-pairs $(\lambda, \mathbf{u})$ = (eigenvalue,eigenvector) where $\mathbf{u}$ is non-zero.

$$\mathbf{A}\mathbf{u} = \overbrace{\lambda}^{\text{eigenvalue}} \underbrace{\mathbf{u}}_{\text{eigenvector}}$$

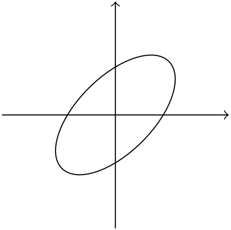
Eigenvector $\mathbf{u}$: Direction of $\mathbf{u}$ is not changed by transformation $\mathbf{A}$. It is only scaled by a factor λ (eigenvalue)

Eigendecomposition: $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues on the diagonal.

Eigenvalues and Eigenvectors



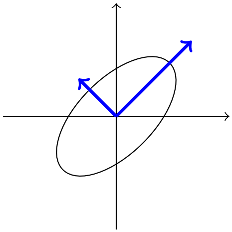
$$\Longrightarrow$$
$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$



What are the (geometric) eigenvectors of this projection?

Solution

The axes of the ellipse do not change their directions:



Equivalent Conditions

For $\mathbf{A} \in \mathbb{R}^{M \times M}$, the following conditions are equivalent:

- $\mathbf{A}$ has an inverse $\mathbf{A}^{-1}$,
- $\text{rank}(\mathbf{A}) = M$,
- $\text{range}(\mathbf{A}) = \mathbb{R}^M$,
- $\text{null}(\mathbf{A}) = \{\mathbf{0}\}$,
- 0 is not an eigenvalue of $\mathbf{A}$,
- 0 is not a singular value of $\mathbf{A}$,