Reinforcement Learning: Tutorial 1

Introduction to RL

Week 1
University of Amsterdam

Milena Kapralova
September 2024

Hello, I'm Milena

Third year MSc AI @ UvA
First year MSc QCS @ UvA
Followed RL last year

My quirk: I furnished my house all in beige without realising

What about you?

Outline

1. Admin

2. Tutorial 1
   - Set up programming environment
   - Play around in the environment
   - Prior knowledge self-test

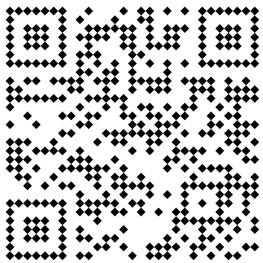
3. Tips n Tricks for the Course

Admin

- Questions
  1. During contact hours: lectures, tutorials
  2. When working outside of contact hours $\rightarrow$ *Ed*
- Tutorials
- Grading
- Deadlines
- Submissions
- Any questions?

Admin

- Questions
- Tutorials
  - Mondays 11-13 @ G0.18B, Tuesdays 11-13 @ G3.10
  - 1 exercise sheet = 1 week = 2 tutorials, find on Canvas
  - After each tutorial I will post the slides here:



github.com/milenakapralova/rlcourse-2024

- Grading
- Deadlines
- Submissions
- Any questions?

Admin

- Questions
- Tutorials
- Grading
  1. Homework (35%)
     - week 2-6 (5x): theory (4%) + coding (2%)
     - week 7: empirical RL (report on a paper) (5%)
  2. Exam (65%)
     - to pass, ≥ 5.0
- Deadlines
- Submissions
- Any questions?

Admin

- Questions
- Tutorials
- Grading
- Deadlines - Weekly, starting next week
  - HW1: 11th Sep @ 17
  - HW2: 18th Sep @ 17
  - HW3: 25th Sep @ 17
  - HW4: 2nd Oct @ 17
  - HW5: 9th Oct @ 17
  - ERL: 16th Oct @ 17
  - Exam: 22nd Oct @ 9-12
  - Resit: 10th Jan @ 9-12
- Submissions
- Any questions?

Admin

- Questions
- Tutorials
- Grading
- Deadlines
- Submissions
  - Coding submissions: Codegra.de
  - Other submissions: Canvas
- Any questions?

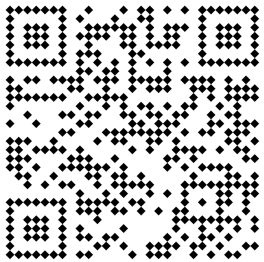
Admin

- Questions
- Tutorials
- Grading
- Deadlines
- Submissions
- Any questions?

Tutorial 1 Overview

This week is supposed to get you started on the course well

1. Set up programming environment



github.com/milenakapralova/rlcourse-2024

2. Play around in the environment
3. Prior knowledge self-test

Tutorial 1 Overview

This week is supposed to get you started on the course well

1. Set up programming environment
2. Play around in the environment
   - Question 0.1
3. Prior knowledge self-test

Q 0.1 Introduction lab (ungraded)

The Multi-Armed Bandit Problem

5 machines, can pull 100 times, how to maximise reward?

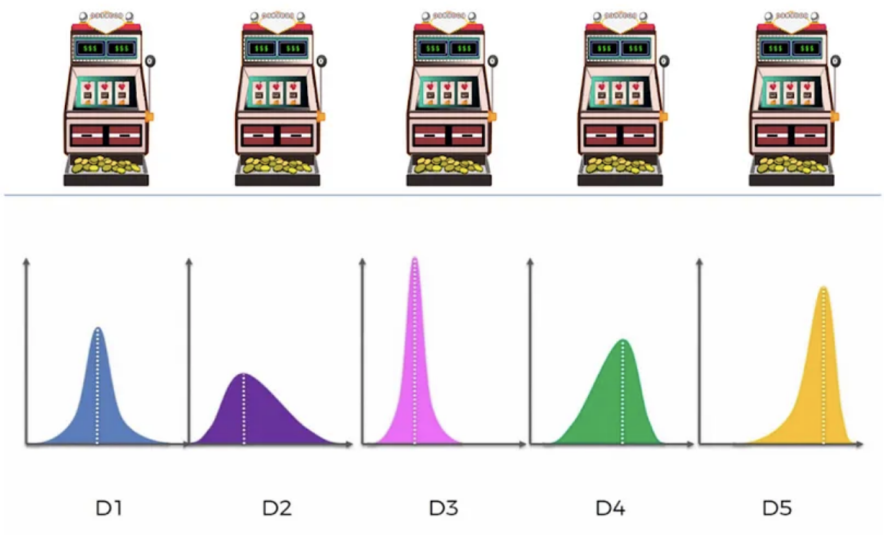


Figure: A Multi-Armed Bandit (Medium).

- Download the notebook *RL_WC1_bandit.ipynb* from Canvas
- Exploration vs exploitation
- Regret = optimal reward - obtained reward

Tutorial 1 Overview

This week is supposed to get you started on the course well

1. Set up programming environment
2. Play around in the environment
3. Prior knowledge self-test
   - Question 0.2

Q 0.2.1: Linear algebra and multivariable derivatives

$$A = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \quad d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

1. Compute AB, AB^T, and $d^T Bd$.

Q 0.2.1: Linear algebra and multivariable derivatives

$$A = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \quad d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

1. Compute AB, AB^T, and $d^T Bd$.

Two matrices in the form $A = \left[a_{nk}\right]_{N*K}$ and $B = \left[b_{km}\right]_{K*M}$:

$$A * B = \left[c_{nm}\right]_{N*M}, \quad c_{nm} = \sum_{k} a_{nk} * b_{km}.$$

Quadratic Matrix. For a N-dimensional vector d and a matrix $B = \left[b_{ij}\right]_{N*N}$, the quadratic form of $d^T Bd$ is computed as follows.

$$d^T Bd = \sum_{j=1}^{N} \sum_{j=1}^{N} b_{ij} d_i d_j \Rightarrow$$

$$AB = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} \\ a_{22}b_{21} & a_{22}b_{22} \end{pmatrix} \quad AB^T = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{21} \\ a_{22}b_{12} & a_{22}b_{22} \end{pmatrix}$$

$$d^T Bd = d_1 b_{11} d_1 + d_1 b_{12} d_2 + d_2 b_{21} d_1 + d_2 b_{22} d_2$$

Q 0.2.1: Linear algebra and multivariable derivatives

$$A = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

2. Find the inverses of A and B.

Q 0.2.1: Linear algebra and multivariable derivatives

$$A = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

2. Find the inverses of A and B.

The inverse of a diagonal matrix is a diagonal matrix whose entries are the reciprocal of the original diagonal. The inverse of a more general 2x2 matrix M is given as $\frac{1}{\det M} M*$, where det(.) is the determinant and $M*$ the adjugate matrix of M.

$$A^{-1} = \begin{pmatrix} 1/a_{11} & 0 \\ 0 & 1/a_{22} \end{pmatrix}, \quad B^{-1} = \frac{1}{b_{11}b_{22} - b_{12}b_{21}} \begin{pmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{pmatrix}.$$

Q 0.2.1: Linear algebra and multivariable derivatives

$$c = \begin{pmatrix} y - x^2 \\ \frac{\ln x}{y} \end{pmatrix} \quad e = \begin{pmatrix} x \\ y \end{pmatrix}$$

3. Compute $\frac{\partial c}{\partial x}$ and $\frac{\partial c}{\partial e}$.

Q 0.2.1: Linear algebra and multivariable derivatives

$$c = \begin{pmatrix} y - x^2 \\ \frac{\ln x}{y} \end{pmatrix} \quad e = \begin{pmatrix} x \\ y \end{pmatrix}$$

3. Compute $\frac{\partial c}{\partial x}$ and $\frac{\partial c}{\partial e}$.

Given a vector of multivariable $x = \left[x_j\right]_{N*1}$ and a multivariate function $F = \left[f_i(x_{1:N})\right]_{M*1}$, the derivative of F towards x is computed by the rule:

$$\frac{\partial F}{\partial x} = \left[\frac{F_i(x_{1:N})}{\partial x_j}\right]_{M*N}$$

where the (i, j)-th derivative value is $\frac{F_i(x_{1:N})}{\partial x_j}$.

Q 0.2.1: Linear algebra and multivariable derivatives

$$c = \begin{pmatrix} y - x^2 \\ \frac{\ln x}{y} \end{pmatrix} \quad e = \begin{pmatrix} x \\ y \end{pmatrix}$$

3. Compute $\frac{\partial c}{\partial x}$ and $\frac{\partial c}{\partial e}$.

Matrix / vector derivatives are defined similarly to scalar derivatives, although we have to choose a layout. We choose the numerator layout (also called Jacobian formulation), in which we treat the matrix / vector in the numerator as being transposed. I.e. if v is an n-vector and w is an m-vector, then $\frac{\partial v}{\partial w}$ will be an $(n \times m)$ matrix. So,

$$\frac{\partial c}{\partial x} = \begin{pmatrix} -2x \\ \frac{1}{xy} \end{pmatrix}, \qquad \frac{\partial c}{\partial e} = \begin{pmatrix} \frac{\partial (y - x^2)}{\partial x} & \frac{\partial (y - x^2)}{\partial y} \\ \frac{\partial (\frac{\ln x}{y})}{\partial x} & \frac{\partial (\frac{\ln x}{y})}{\partial y} \end{pmatrix} = \begin{pmatrix} -2x & 1 \\ \frac{1}{xy} & -\frac{\ln x}{y^2} \end{pmatrix}.$$

Q 0.2.1: Linear algebra and multivariable derivatives

4. Consider the function $f(x) = \sum_{i=1}^{N} ix_i$, which maps an N-dimensional vector of real numbers x to a real number. Find an expression for $\frac{\partial f}{\partial x}$ in terms of integers 1 to N..

Q 0.2.1: Linear algebra and multivariable derivatives

4. Consider the function $f(x) = \sum_{i=1}^{N} ix_i$, which maps an N-dimensional vector of real numbers x to a real number. Find an expression for $\frac{\partial f}{\partial x}$ in terms of integers 1 to N..

Again, we treat x as if transposed (i.e. a row vector). Thus our derivative:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial (x_1, x_2, ..., x_N)} = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_N} \right)$$

Evaluating a single term by inspection of $f(x) = \sum_i^N ix_i$, or explicitly:

$$\frac{\partial f}{\partial x_j} = \frac{\partial \left(\sum_i^N ix_i \right)}{\partial x_j} = \sum_i^N \frac{\partial (ix_i)}{\partial x_j} = \sum_i^N i\delta_{ij} = j$$

where δ_{ij} is the Kronecker delta, which equals one if $i = j$ and zero otherwise. Plugging the result into our expression for $\frac{\partial f}{\partial x}$ we obtain: $\frac{\partial f}{\partial x} = (1, 2, ..., N)$ (i.e. the row-vector of integers 1 to N).

Q 0.2.2: Probability theory

Assume X and Y are two independent random variables, respectively with mean μ, ν, and variance σ^2, τ^2.

1. What is the expected value of $X + \alpha Y$, where α is some constant?

Q 0.2.2: Probability theory

Assume X and Y are two independent random variables, respectively with mean μ, ν, and variance σ^2, τ^2.

1. What is the expected value of $X + \alpha Y$, where α is some constant?

From the linearity of expectation we have

$$\mathrm{E}[X + aY] = \mathrm{E}[X] + a\mathrm{E}[Y] = \mu + a\nu.$$

Q 0.2.2: Probability theory

Assume X and Y are two independent random variables, respectively with mean μ, ν, and variance σ^2, τ^2.

2. What is the variance of $X + \alpha Y$?

Q 0.2.2: Probability theory

Assume X and Y are two independent random variables, respectively with mean μ, ν, and variance σ^2, τ^2.

2. What is the variance of $X + \alpha Y$?

We know that for independent variables

$$\mathrm{Var}[aX + bY] = a^2\mathrm{Var}[X] + b^2\mathrm{Var}[Y].$$

So, $\mathrm{Var}[X + \alpha Y] = \sigma^2 + \alpha^2\tau^2$.

Q 0.2.2: Probability theory

We want a function $\hat{f}$ that minimises the squared error $(y - \hat{f}(x))^2$. The bias-variance decomposition of this error on an unseen sample x:

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = \left(\mathrm{Bias}[\hat{f}(x)]\right)^2 + \mathrm{Var}[\hat{f}(x)] + \sigma^2 \qquad (1)$$

where

$$\mathrm{Bias}[\hat{f}(x)] = \mathrm{E}[\hat{f}(x)] - f(x) \qquad \mathrm{Var}[\hat{f}(x)] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[\hat{f}(x)]^2$$

and the expectation $\mathrm{E}[.]$ ranges over different choices of the training set $(x_1, x_2, ..., x_n; y_1, y_2, ..., y_n)$, all sampled from the same joint distribution $P(X, Y)$.

3. Explain what errors the various terms in equation (1) represent. For instance, when is a particular term large or small?

Q 0.2.2: Probability theory

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = \left(\mathrm{Bias}[\hat{f}(x)]\right)^2 + \mathrm{Var}[\hat{f}(x)] + \sigma^2 \tag{1}$$

3. Explain what errors the various terms in equation (1) represent. For instance, when is a particular term large or small?

- The bias term = the error caused by the simplifying assumptions in our model. It is large when the model is too simple to represent our training data, and small when our model is complex enough to fit the training data well.
- The variance term = the sensitivity of our model to different sets of training data. It is high when the model is complex enough to fit many potential sets of training data well (probably fitting some of the noise), and small when our model is too simple.
- The final term = the irreducible error. Whether it is high or low depends on how precisely our dataset was collected.

Q 0.2.2: Probability theory

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = \left(\mathrm{Bias}[\hat{f}(x)]\right)^2 + \mathrm{Var}[\hat{f}(x)] + \sigma^2 \tag{1}$$

4. Explain why this decomposition is also known as the 'bias-variance trade-off'.

Q 0.2.2: Probability theory

$$\mathrm{E}\left[(y - \hat{f}(x))^2\right] = \left(\mathrm{Bias}[\hat{f}(x)]\right)^2 + \mathrm{Var}[\hat{f}(x)] + \sigma^2 \tag{1}$$

4. Explain why this decomposition is also known as the 'bias-variance trade-off'.

- Typically, changes to our model that reduce bias will increase variance, and vice-versa.
- A model with small bias is strong enough to fit the training data, but this generally means it will also find some spurious correlations resulting from the precise training dataset drawn from $P(X, Y)$, which results in high variance.
- Similarly, a model with low variance does not fit these spurious correlations, but then generally will be worse at fitting different instances of the training data.

Q 0.2.3: OLS, linear projection and gradient descent

Suppose now that we have a training set X consisting of n vectors (datapoints) of size m (features). Associated to this we have an n-dimensional vector of real values y. Suppose we want to regress a linear model to this data using ordinary least-squares (OLS). That is, we fit linear model $f_\beta(X) = X\beta$, such that we minimise $||y - f_\beta(X)||_2^2$ over the parameters β.

1. What is the dimensionality of the parameter vector β?

Q 0.2.3: OLS, linear projection and gradient descent

Suppose now that we have a training set X consisting of n vectors (datapoints) of size m (features). Associated to this we have an n-dimensional vector of real values y. Suppose we want to regress a linear model to this data using ordinary least-squares (OLS). That is, we fit linear model $f_\beta(X) = X\beta$, such that we minimise $||y - f_\beta(X)||_2^2$ over the parameters β.

1. What is the dimensionality of the parameter vector β?

The dimensionality of β is m.

Q 0.2.3: OLS, linear projection and gradient descent

Suppose now that we have a training set X consisting of n vectors (datapoints) of size m (features). Associated to this we have an n-dimensional vector of real values y. Suppose we want to regress a linear model to this data using ordinary least-squares (OLS). That is, we fit linear model $f_\beta(X) = X\beta$, such that we minimise $||y - f_\beta(X)||_2^2$ over the parameters β.

2. Show by differentiation that the OLS estimator $\hat{\beta}$ equals $(X^T X)^{-1} X^T y$.

Q 0.2.3: OLS, linear projection and gradient descent

Suppose now that we have a training set X consisting of n vectors (datapoints) of size m (features). Associated to this we have an n-dimensional vector of real values y. Suppose we want to regress a linear model to this data using ordinary least-squares (OLS). That is, we fit linear model $f_\beta(X) = X\beta$, such that we minimise $||y - f_\beta(X)||_2^2$ over the parameters β.

2. Show by differentiation that the OLS estimator $\hat{\beta}$ equals $(X^T X)^{-1} X^T y$.

We set $L = ||y - X\beta||_2^2 = (y - X\beta)^T (y - X\beta)$, and set $\frac{\partial L}{\partial \beta} = 0$.

$$0 = \frac{\partial L}{\partial \beta}$$

$$0 = -2X^T (y - X\beta)$$

$$X^T y = X^T X\beta$$

$$(X^T X)^{-1} X^T y = \beta$$

Q 0.2.3: OLS, linear projection and gradient descent

Another method for deriving the OLS estimator is by thinking geometrically. Imagine y and $X\beta$ as vectors in an n-dimensional vector space. Further note that the regressors $X\beta$ actually span an m-dimensional subspace of this larger vector space (the column space of X):

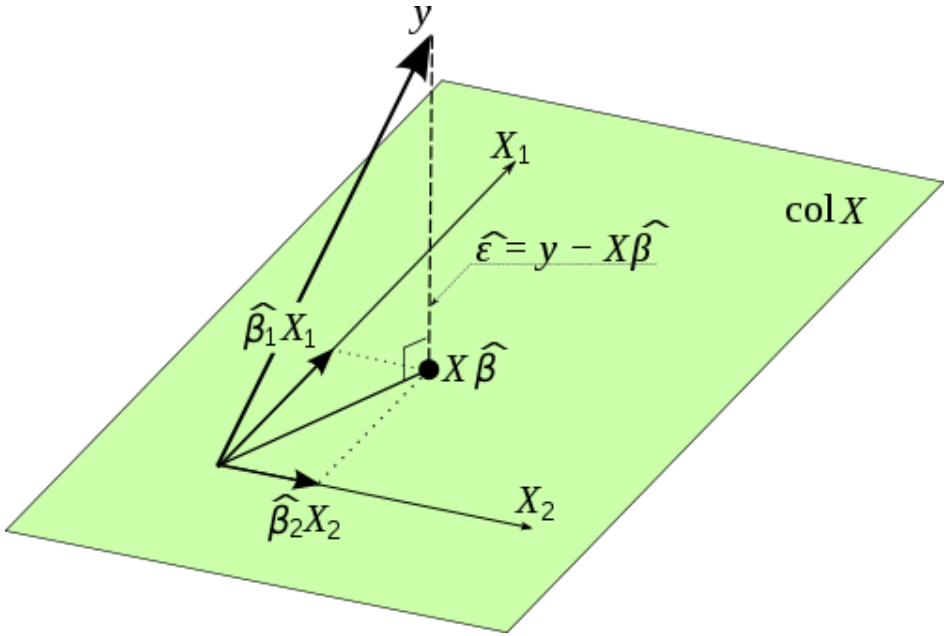
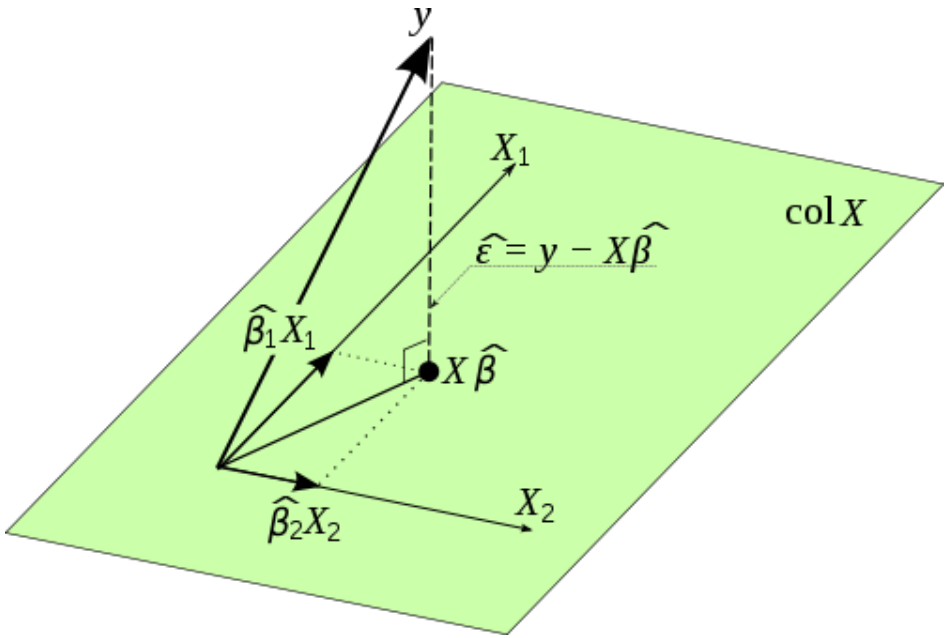


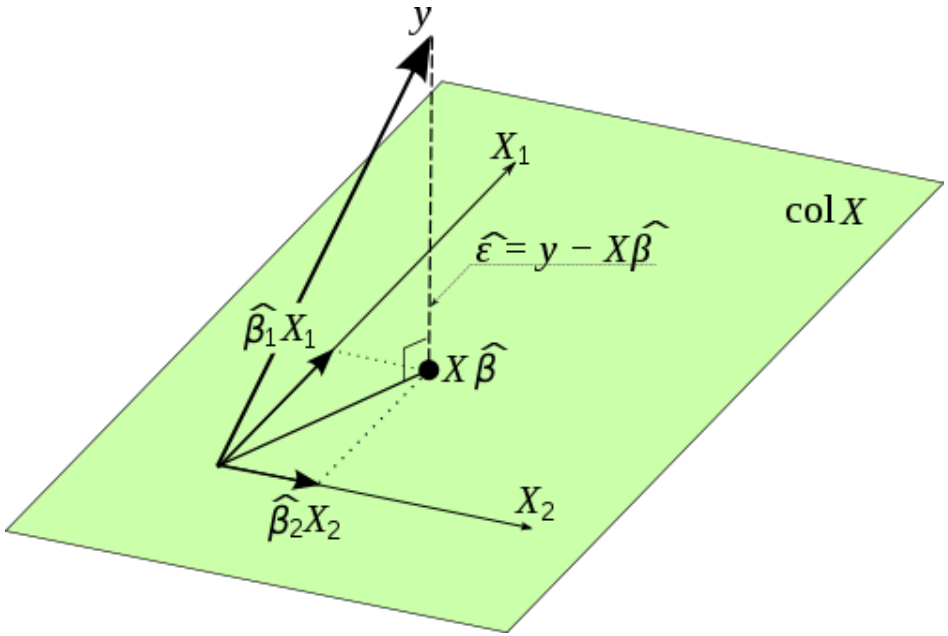
Figure: Geometric representation of OLS (Wikipedia). The X_i are columns of X.

Q 0.2.3: OLS, linear projection and gradient descent



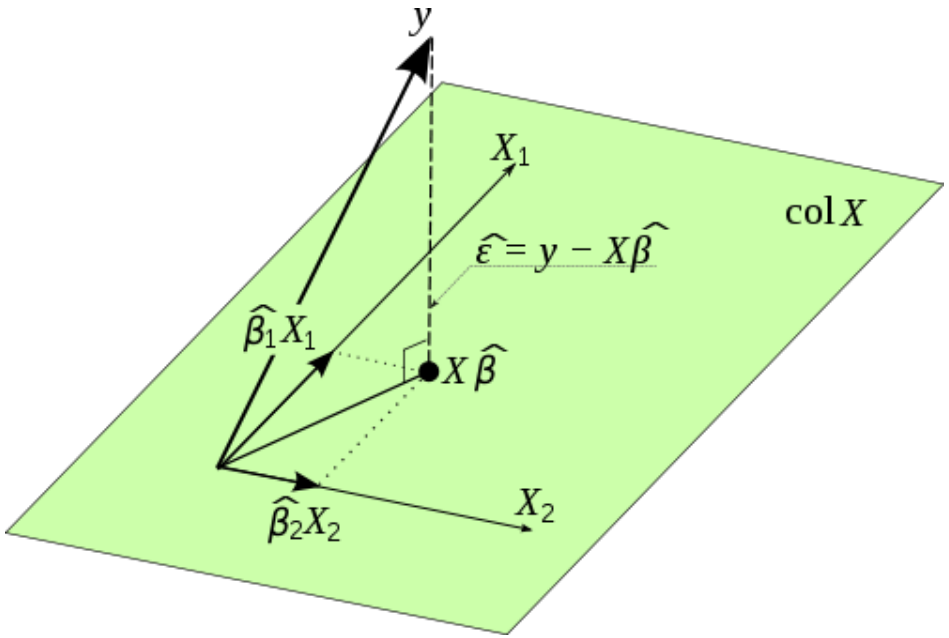
3. Define as ϵ_β the residual vector $(y - X\beta)$. Argue that minimising the L_2 norm of this vector over β (as we do in OLS) is equivalent to choosing β such that $X\beta = P(y)$, where $P(.)$ orthogonally projects y onto the linear subspace spanned by the regressors.

Q 0.2.3: OLS, linear projection and gradient descent



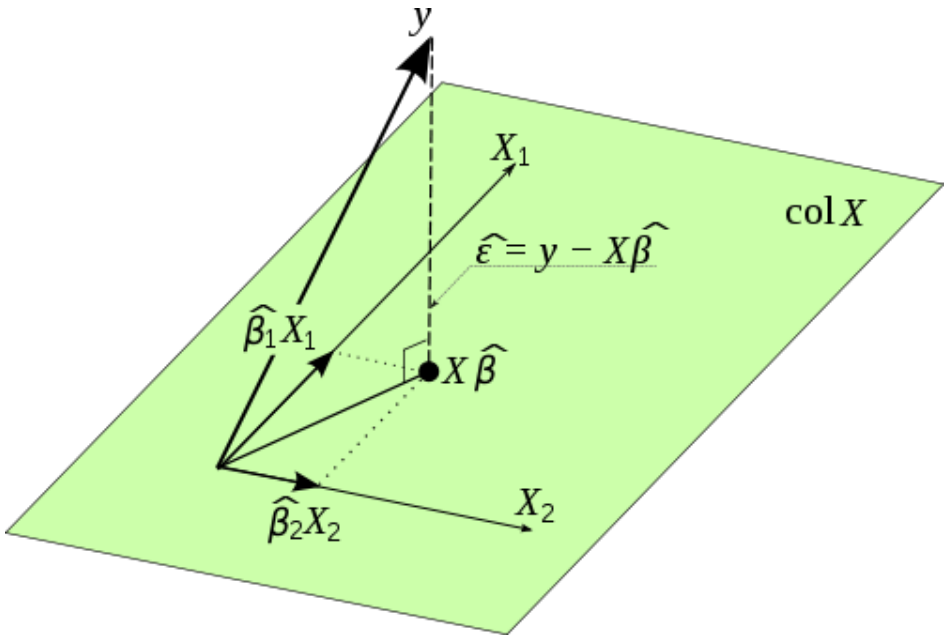
- We can imagine moving $X\beta$ around and seeing what happens to the residual. There is no way to move $X\beta$ such that the residual is smaller than at the pictured $X\hat{\beta}$, where the residual is exactly orthogonal to the column space of $X \equiv X\hat{\beta}$ is the orthogonal projection $P(y)$ of y onto this subspace.
- Note that all the vectors in the figure are n-dimensional, and that col$X$ is an m-dimensional subspace spanned by m of these vectors.

Q 0.2.3: OLS, linear projection and gradient descent



4. Argue that this is equivalent to setting $X^T \epsilon_\beta = 0$.

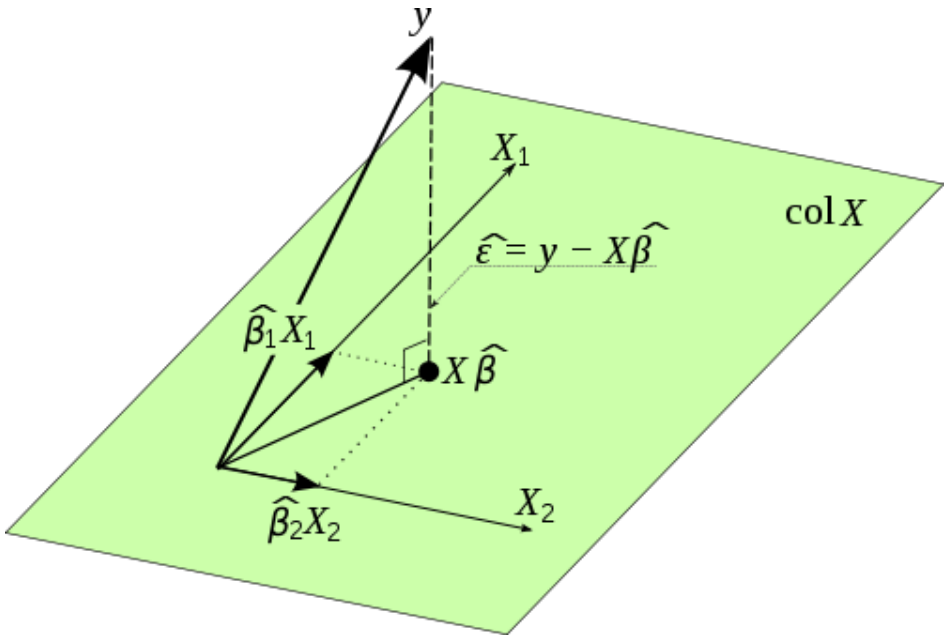
Q 0.2.3: OLS, linear projection and gradient descent



4. Argue that this is equivalent to setting $X^T \epsilon_\beta = 0$.

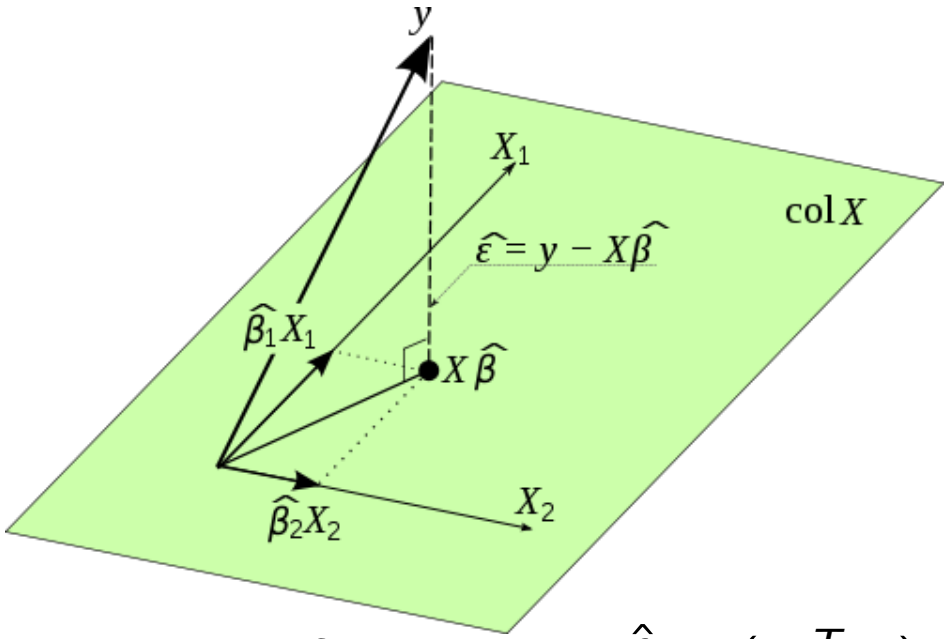
The residual ϵ_β is orthogonal to the column space of X. This means that the inner product of the residual with any of the columns of X is zero, thus $X^T \epsilon_\beta = 0$. Equivalently we can write $\epsilon_\beta^T X = 0$.

Q 0.2.3: OLS, linear projection and gradient descent



5. Show that from this also follows that $\hat{\beta} = (X^T X)^{-1} X^T y$.

Q 0.2.3: OLS, linear projection and gradient descent



5. Show that from this also follows that $\hat{\beta} = (X^T X)^{-1} X^T y$.

Using that $\epsilon_\beta = (y - X\beta)$:

$$0 = X^T \epsilon_\beta \Leftrightarrow 0 = X^T (y - X\beta)$$

$$0 = X^T y - X^T X\beta \Leftrightarrow X^T X\beta = X^T y$$

$$\beta = (X^T X)^{-1} X^T y$$

Q 0.2.3: OLS, linear projection and gradient descent

When our models get more complicated, we quickly lose the ability to optimise them analytically. Instead, we use numerical optimisation methods such as gradient descent. Although we do not need gradient descent to fit the model in this case, we can still do so.

6. In gradient descent we minimise a loss function $L_\beta(y, X)$ over the parameter values β. What is the loss function corresponding to the OLS description from before?

Q 0.2.3: OLS, linear projection and gradient descent

When our models get more complicated, we quickly lose the ability to optimise them analytically. Instead, we use numerical optimisation methods such as gradient descent. Although we do not need gradient descent to fit the model in this case, we can still do so.

6. In gradient descent we minimise a loss function $L_\beta(y, X)$ over the parameter values β. What is the loss function corresponding to the OLS description from before?

$L_\beta(y, X) = ||y - f_\beta(X)||^2$, where $||...||^2$ is the squared L_2 norm.

Q 0.2.3: OLS, linear projection and gradient descent

When our models get more complicated, we quickly lose the ability to optimise them analytically. Instead, we use numerical optimisation methods such as gradient descent. Although we do not need gradient descent to fit the model in this case, we can still do so.

7. Assume we use a learning rate α. Write the update rule for a single gradient descent step on our parameters β?

Q 0.2.3: OLS, linear projection and gradient descent

When our models get more complicated, we quickly lose the ability to optimise them analytically. Instead, we use numerical optimisation methods such as gradient descent. Although we do not need gradient descent to fit the model in this case, we can still do so.

7. Assume we use a learning rate α. Write the update rule for a single gradient descent step on our parameters β?

We move in opposite direction of the gradient of our loss function (since we want to reduce loss), so:

$$\beta_{t+1} = \beta_t - \alpha \frac{\partial L}{\partial \beta_t}$$

$$\beta_{t+1} = \beta_t + 2\alpha X^T (y - X\beta_t)$$


WELL DONE!

Tips n Tricks for the Course

- Implementations are not complex, they may be unclear at first: all you need to know is either in lecture/material referenced in the notebook
- *Textbook* (Sutton & Barto, 2nd edition) is a good source when there are confusions
- Another *textbook* used less
- Visualising helps with debugging - e.g. rewards per epoch
- A good *summary* of the course from a couple years ago by Phillip Lippe (good to get an overview and revise for the exam)

That's it!

See you tomorrow