# Ch2. Mathematical Toolbox

ST4240, 2016/2017

Version $0.1$

Alexandre Thiéry

Department of Statistics and Applied Probability

# Outline

# Linear algebra: why do we care?

- For linear models!
- We will be doing multivariate calculus.
- Will need to understand calculus in high-dimensions
- Gaussian random variables in high-dimensions

# Scalar product

- For $u, v \in \mathbb{R}^d$ the scalar product is defined as

$$\langle u, v \rangle = \sum_{i=1}^{d} u_i \, v_i = u^\top \, v$$

- For $u, v \in \mathbb{R}^d$ and matrix $A \in \mathbb{R}^{d \times d}$

$$\langle u, A \, v \rangle = \sum_{1 \leq i,j \leq d} A_{i,j} u_i v_j = u^\top \, A \, v$$

- For $u \in \mathbb{R}^p$, $v \in \mathbb{R}^q$, and matrix $A \in \mathbb{R}^{p \times q}$

$$\langle u, Av \rangle = \left\langle A^\top u, v \right\rangle$$

# Inverse

- For two matrices $A, B$ we have

$$(A\,B)^\top = B^\top\,A^\top \qquad \text{and} \qquad (A\,B)^{-1} = B^{-1}\,A^{-1}$$

- **[Exercise]** transpose and inverse interact as follows

$$\left(A^\top\right)^{-1} = \left(A^{-1}\right)^\top$$

# Determinant

- The determinant satisfies the identity

$$\det(A\,B) = \det(A)\,\det(B) \qquad \text{and} \qquad \det(A^{-1}) = (\det(A))^{-1}\,.$$

- The determinant of a **triangular matrix** is the product of the elements on the diagonal

$$\det \begin{pmatrix} 2 & & \\ 6 & 1 & \\ -8 & 5 & 3 \end{pmatrix} = 2 \times 1 \times 3 = 6$$

# Orthogonal matrices

- A matrix $O \in \mathbb{R}^{d \times d}$ is called orthogonal if $O\, O^\top = I$.
- Equivalently, a matrix $O \in \mathbb{R}^{d \times d}$ is orthogonal if $O^{-1} = O^\top$.
- Equivalently, a matrix $O \in \mathbb{R}^{d \times d}$ is orthogonal if it rows (or column) have norm $1$ and are perpendicular.
- If $O$ is orthogonal, for any vector $v$ we have

$$\|O\, v\| = \|v\|.$$

- [**Exercise**] prove that the following is orthogonal

$$O = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

# Symmetric matrices

- A matrix $S \in \mathbb{R}^{d \times d}$ is called **symmetric** if $M_{i,j} = M_{j,i}$ for all $1 \leq i, j \leq d$.
- A symmetric matrix $M$ is diagonalizable in an orthogonal basis,

$$M = O\, D\, O^\top$$

for an **orthogonal** matrix $O$ and a diagonal matrix

$$D = \mathrm{Diag}(\lambda_1, \ldots, \lambda_d).$$

The $\{\lambda_i\}_{i=1}^d$ are the **eigenvalues** of $M$.

# Positive Symmetric matrices

- A symmetric matrix $S \in \mathbb{R}^{d \times d}$ is positive semi-definite if for any vector $v \in \mathbb{R}^d$ we have

$$\langle x, Sx \rangle \geq 0.$$

  **Caution:** elements of $S$ do not need to be non-negative.

- [**Exercise**] a covariance matrix is positive semi-definite.
- [**Exercise**] the sum of two positive definite matrices is positive definite
- A symmetric matrix $S \in \mathbb{R}^{d \times d}$ is positive definite if for any vector $v \neq 0$ we have

$$\langle x, Sx \rangle > 0.$$

- Symmetric matrix $S$ is positive definite if $\lambda_i > 0$ for all $i$.

# Example

- [**Exercise**] is the matrix

$$M = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

positive definite? Is it positive semi-definite?

# Cholesky Decomposition

- A symmetric positive definite matrix $S$ can always be decomposed as

$$S = L\,L^{\top}$$

  where $L$ is a lower triangular matrix. This is the Cholesky decomposition of $S$.

- For example,

$$S = \begin{pmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{pmatrix} = \begin{pmatrix} 2 & & \\ 6 & 1 & \\ -8 & 5 & 3 \end{pmatrix} \begin{pmatrix} 2 & 6 & -8 \\ & 1 & 5 \\ & & 3 \end{pmatrix}$$

- **[Exercise]** compute the determinant of $S$.

# Outline

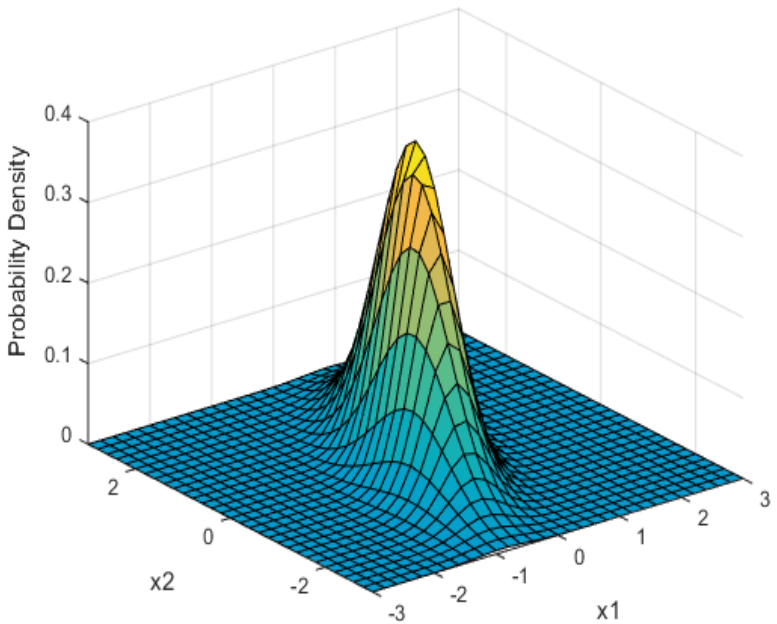
# Multivariate Gaussian Distribution: why do we care?

- A very good model in many situations
- The simplest distribution in high-dimension
- Tractable computations
- The basis of many more advanced models

# Multivariate Gaussian Density

- A Gaussian distribution in $\mathbb{R}^d$ with mean $\mu \in \mathbb{R}^d$ and covariance $\Gamma \in \mathbb{R}^{d \times d}$ has density, if $\Gamma$ is invertible, given by

$$\frac{1}{(2\,\pi)^{d/2}\ \det(\Gamma)^{1/2}} \times \exp\left\{-\frac{1}{2}\left\langle [x-\mu], \Gamma^{-1}\,[x-\mu]\right\rangle\right\}$$



# Multivariate Gaussian Density

- If $\xi$ is a Gaussian random variable in $\mathbb{R}^d$ with mean $\mu \in \mathbb{R}^d$ and covariance $\Gamma \in \mathbb{R}^{d \times d}$ then for a matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$ the random variable $Y = A\,\xi + b$ has means

$$\mathrm{Mean}(\,Y) = A\,\mu + b$$

and covariance

$$\mathrm{Cov}(\,Y) = A\,\Gamma\,A^\top$$

# Gaussian simulation

- Let $\Gamma \in \mathbb{R}^{d \times d}$ be a covariance matrix and $\mu \in \mathbb{R}^d$. Consider the Cholesky decomposition

$$\Gamma = L\, L^\top.$$

- **[Exercise]** the random variable $\xi = L\,\zeta + \mu$, with $\zeta = \mathbf{N}\,(0, I_d)$ is a Gaussian random variable with mean $\mu$ and covariance $\Gamma$.

# Outline

# Multivariate Calculus: why do we care?

- Data Mining is all about exploring high-dimensional data
- We need to understand how to describe functions in high-dimensional spaces
- Most importantly: we will need to optimize high-dimensional functions:

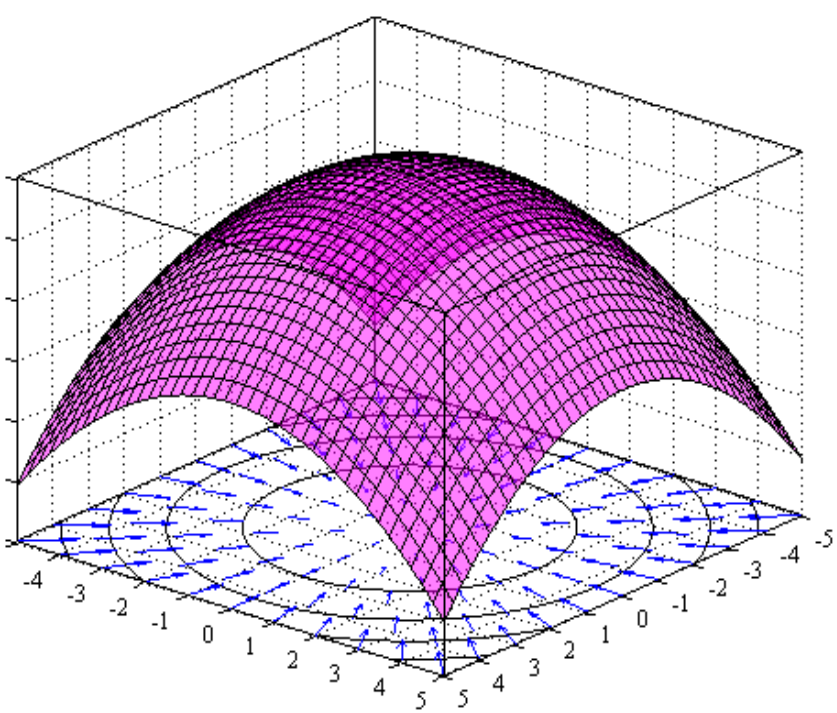
$$F : \mathbb{R}^d \to \mathbb{R}$$

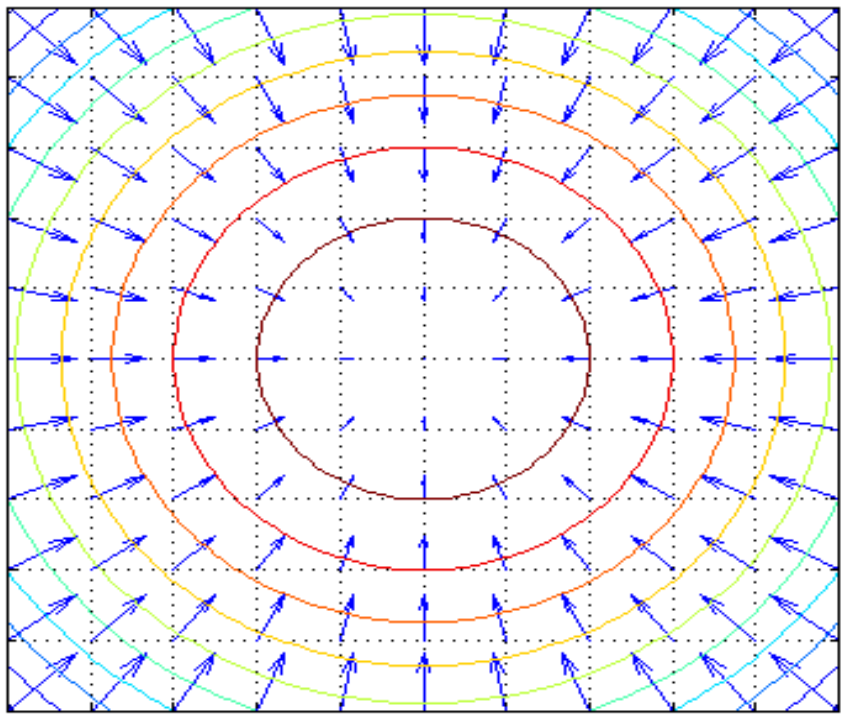
# Gradient

- Consider a function $f : \mathbb{R}^d \to \mathbb{R}$. The gradient is defined as

$$\nabla f(x) = (\partial_{x_1} f(x), \ldots, \partial_{x_d} f(x))$$

- Intuitively: $f(x + \varepsilon) \approx f(x) + \langle \nabla f(x), \varepsilon \rangle$
- Example: $f(x) = -\frac{1}{2}\ (x_1^2 + \ldots + x_d^2)$.



# Gradient: important examples

- **[Exercise]** Let $v \in \mathbb{R}^d$ and $f(x) = \langle v, x \rangle$,

$$\nabla f(x) = \dots$$

- **[Exercise]** Let $M \in \mathbb{R}^{d \times d}$ and $g(x) = \frac{1}{2} \langle x, Mx \rangle$,

$$\nabla g(x) = \dots$$

# Hessian matrix

- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function
- The Hessian of $f$ at $x$ is the symmetric matrix $\mathrm{Hess}(x) \in \mathbb{R}^{d \times d}$ defined as

$$\mathrm{Hess}(x)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x)$$

- Intuitively: $f(x+\varepsilon) \approx f(x) + \langle \nabla f(x), \varepsilon \rangle + \frac{1}{2} \langle \varepsilon, \mathrm{Hess}(x)\, \varepsilon \rangle$
- **[Exercise]** what is the Hessian of $f(x) = \langle v, x \rangle$ ?
- **[Exercise]** what is the Hessian of $g(x) = \frac{1}{2} \langle x, Mx \rangle$ ?

# Outline

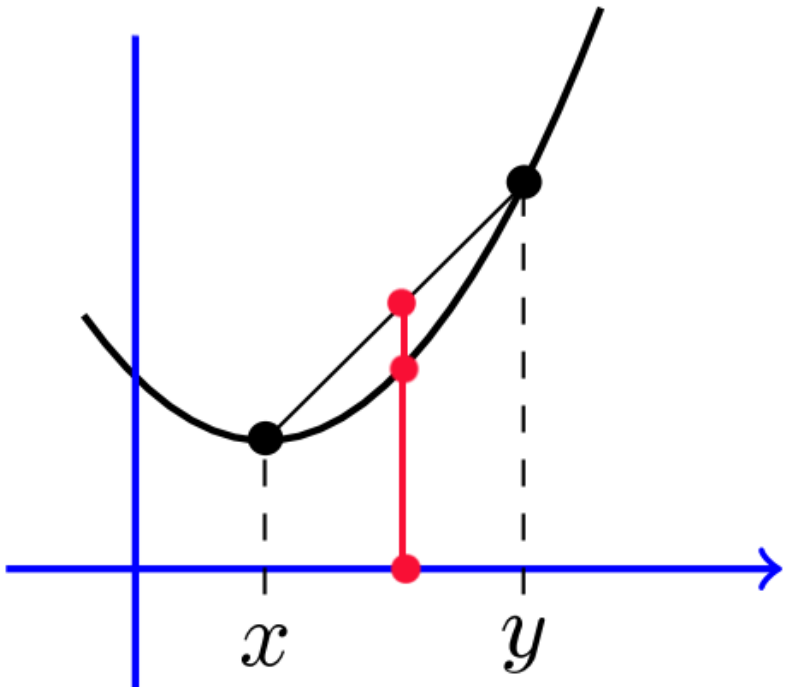
# Optimization: why do we care?

- Data Mining is a mix of data visualization, building appropriate models and efficiently fit them to (large) datasets
- Fitting statistical models to data can be formulated, in many cases, as an optimization problem.

# Convex function

- A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for any $x, y \in \mathbb{R}^d$ and $0 < \lambda < 1$ we have

$$f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y).$$



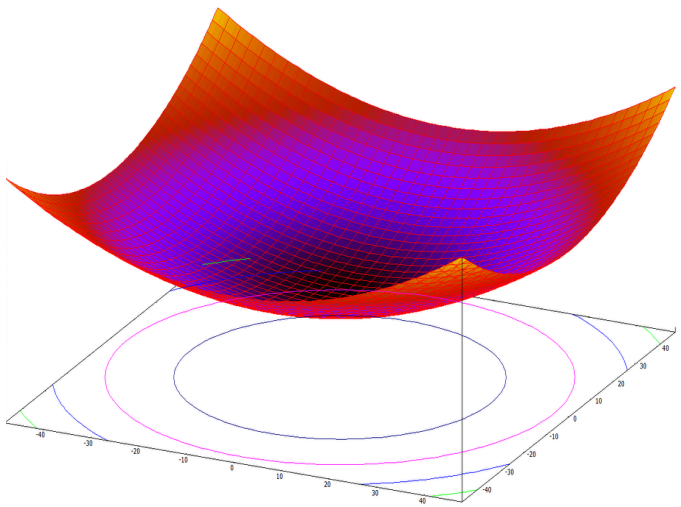
# Strict Convexity

- A function $f : \mathbb{R}^d \to \mathbb{R}$ is strictly convex if for any $x \neq y \in \mathbb{R}^d$ and $0 < \lambda < 1$ we have

$$f\left(\lambda\, x + (1-\lambda)\, y\right) < \lambda\, f\left(x\right) + (1-\lambda)\, f\left(y\right).$$

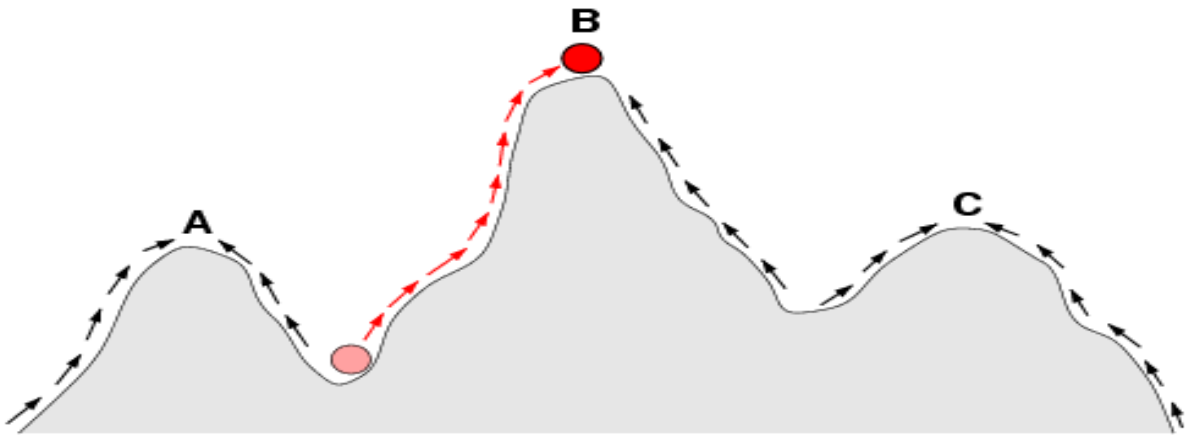
# Convex function: second derivative criterion

- Consider a function $f : \mathbb{R}^d \to \mathbb{R}$.
- If the Hessian matrix $\mathrm{Hess}(x)$ is positive semi-definite for every $x \in \mathbb{R}^d$ then the function is convex.
- If the Hessian matrix $\mathrm{Hess}(x)$ is positive definite for every $x \in \mathbb{R}^d$ then the function is strictly convex.

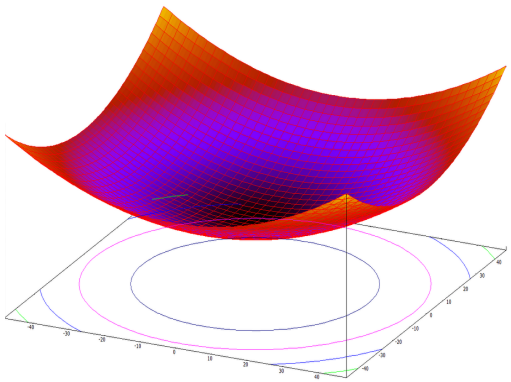


- **[Exercise]** the sum of two convex functions is convex

# Optimization and Convexity



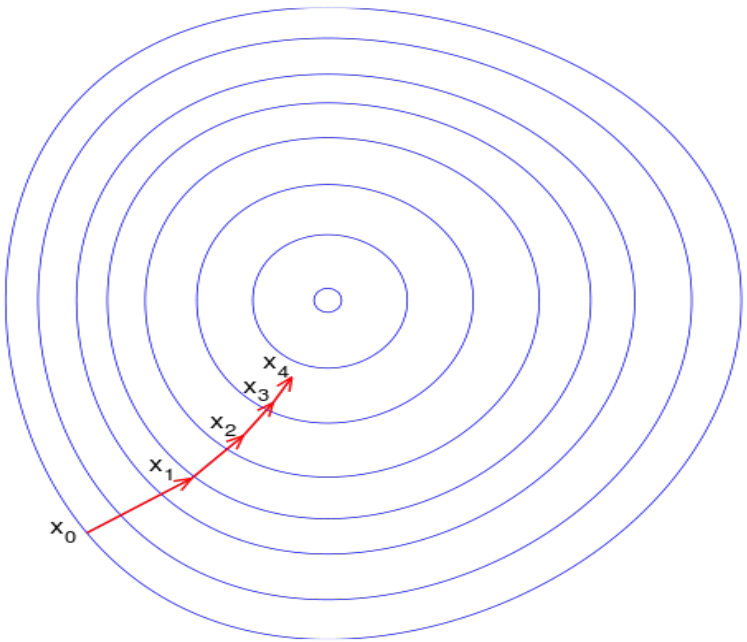

- If a strictly convex function has a local minimum, this minimum is the unique global minimum.

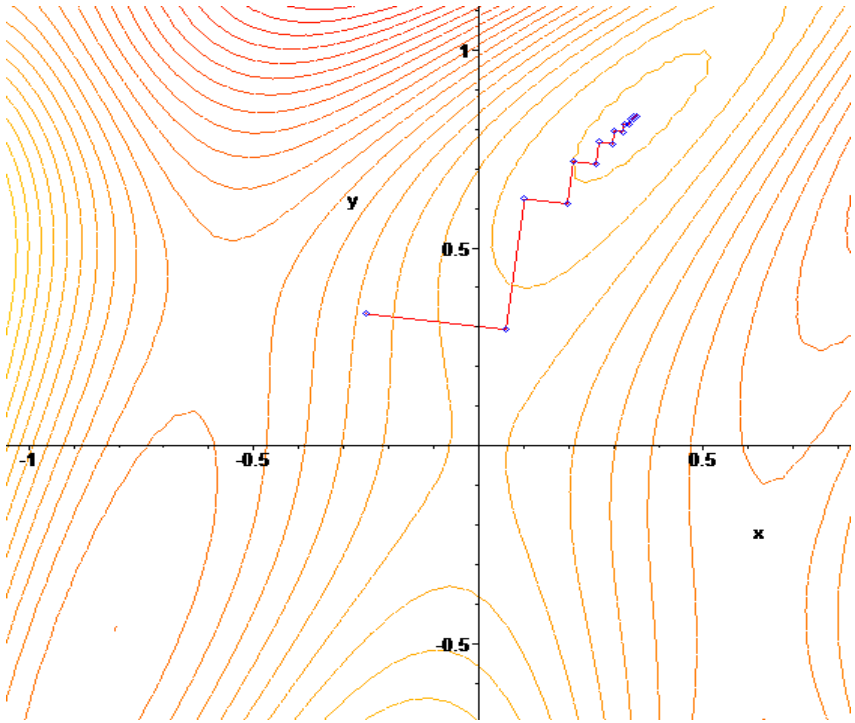


# Gradient descent

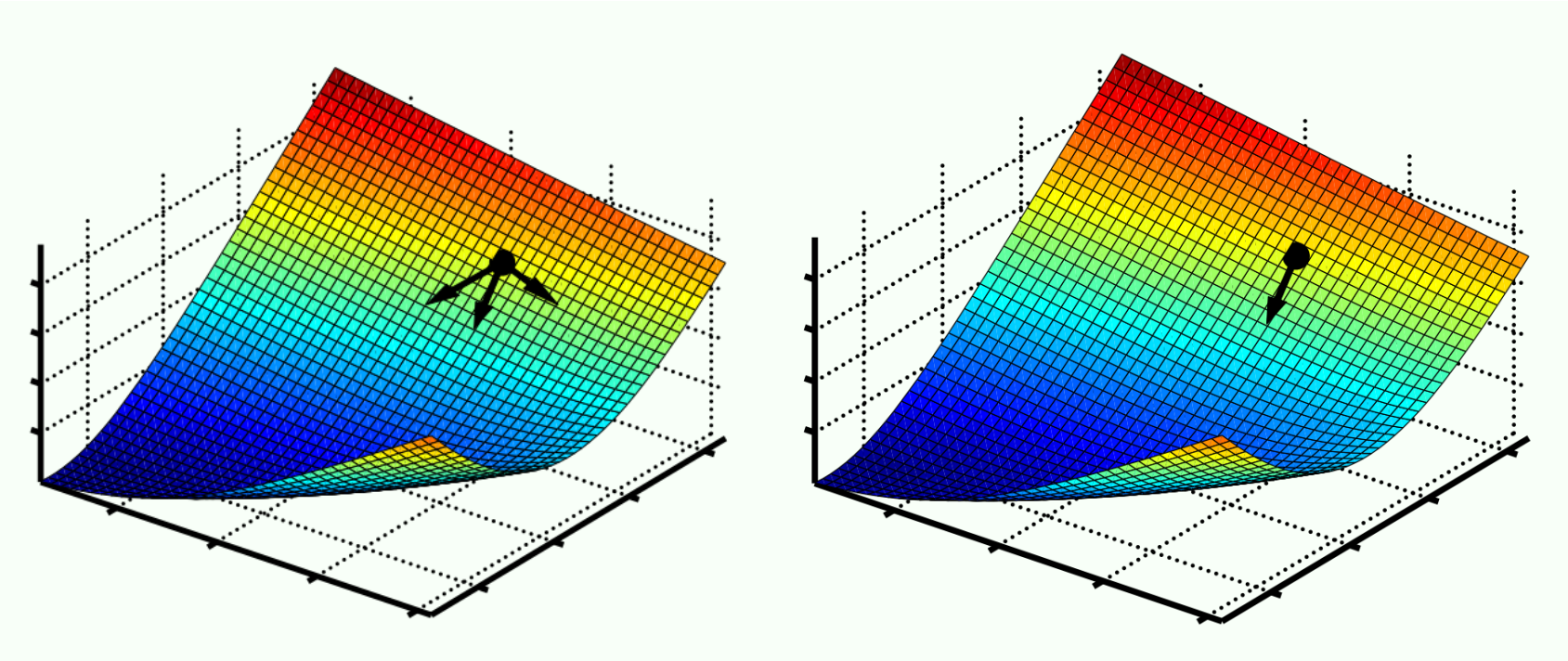
- Consider $f : \mathbb{R}^d \to \mathbb{R}$ a function to minimize.
- For a learning rate $\eta > 0$ (or step-size) the gradient descent algorithm is as follows,

$$x_{n+1} = x_n - \eta \nabla f(x_n)$$

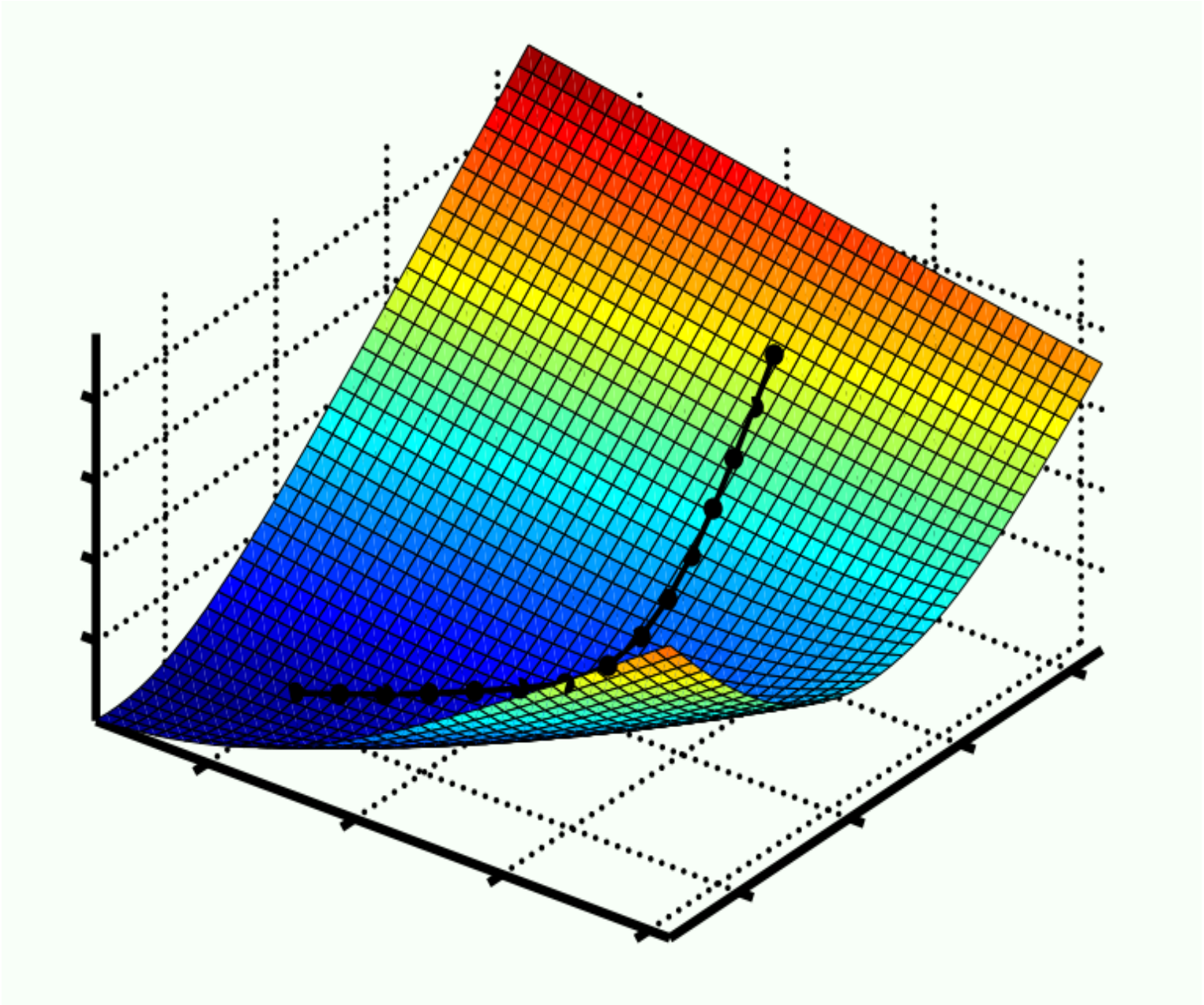


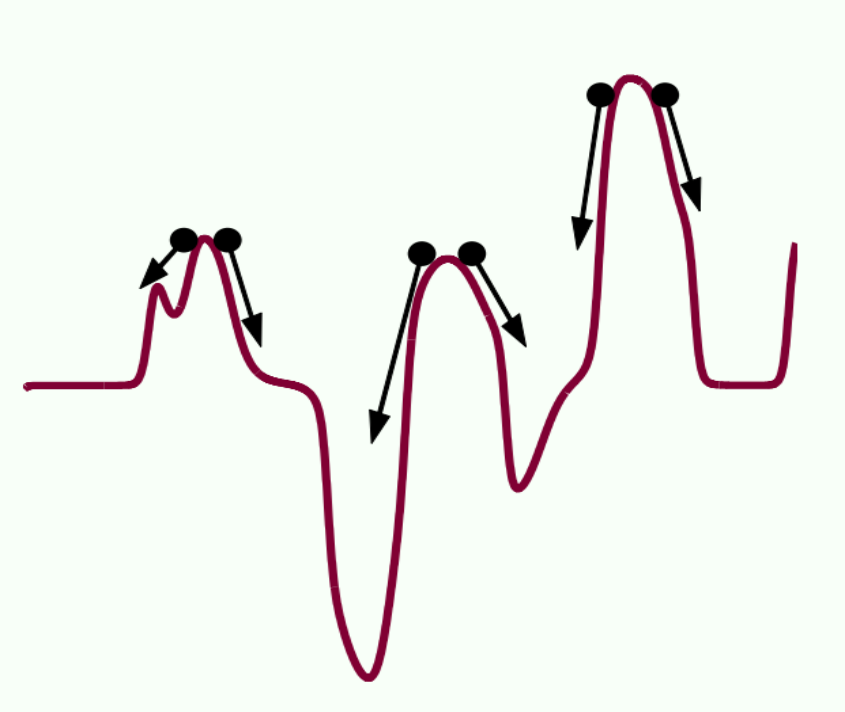
# Gradient descent



# Gradient descent



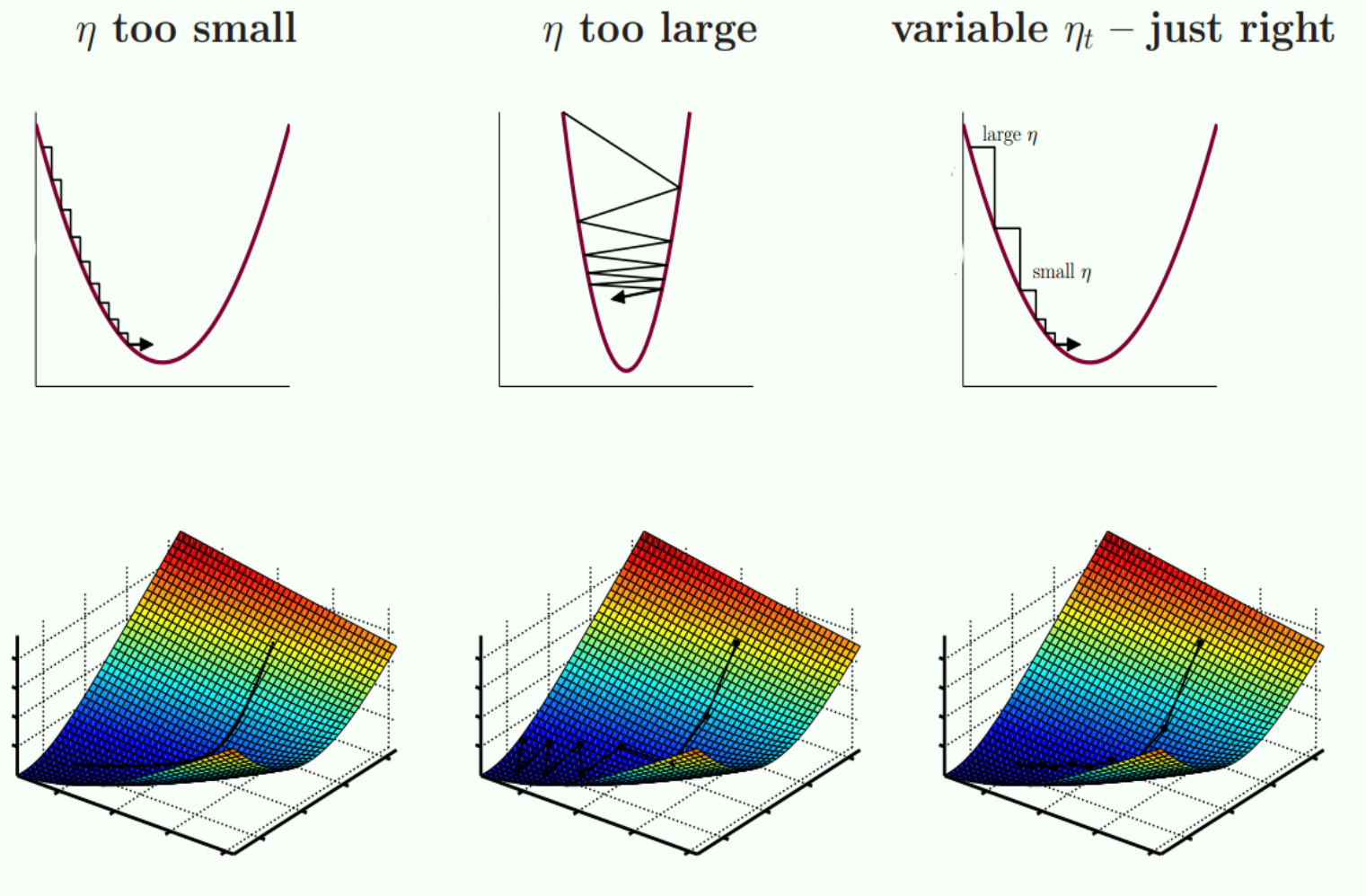
# Gradient descent can get stuck!



This cannot happen when optimizing a convex function!

# On the Learning Rate

- Gradient descent reads $x_{n+1} = x_n - \eta \nabla f(x_n)$.
- [**Exercise**] Gradient descent for $f(x) = \frac{1}{2}x^2$: influence of $\eta$?



# Gradient descent

- We would like to solve the equation $A\,x = b$ for a positive definite matrix $A \in \mathbb{R}^d$.
- Consider the

$$f(x) = \frac{1}{2}\left\langle A\,x - b, A^{-1}(Ax - b)\right\rangle$$

- **[Exercise]** prove that the function $f$ is strictly convex.
- **[Exercise]** prove that the solution $x^\star$ of $A\,x = b$ is the unique global minimum of $f$.
- **[Exercise]** write the gradient descent algorithm for minimizing the function $f$.

# Outline

# Performance and Loss function

- Performance of a parameter $\theta$ can be quantified by

$$F(\theta) = \frac{1}{N} \sum_{i=1}^{N} \textbf{Loss}(\theta, x_i, y_i) + \Omega(\theta)$$

  where
  - **Loss**$(\,\cdot\,)$ quantifies the fit to the data
  - $\{x_i, y_i\}_{i=1}^{N}$ are training examples
  - $\Omega(\theta)$ is a regularisation term
- One would to minimise the function $F$,

$$\widehat{\theta} = \textbf{argmin}\ F(\theta)$$

# Regularized Linear Regression

- In linear regression, the least square estimate $\widehat{\beta}$ minimises the function

$$F(\beta) = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - \langle x_i, \beta \rangle \right)^2 .$$

  Note that there is no regularisation term.

- For ridge regression, the estimate $\widehat{\beta}_{\text{ridge}}$ minimises

$$F_{\text{ridge}}(\beta) = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - \langle x_i, \beta \rangle \right)^2 + \lambda \sum_{i=1}^{d} \beta_i^2$$

  [**Exercise**] find $\nabla F_{\text{ridge}}(\beta)$.

- For LASSO regression, the estimate $\widehat{\beta}_{\text{lasso}}$ minimises

$$F_{\text{ridge}}(\beta) = \frac{1}{N} \sum_{i=1}^{N} \left( y_i - \langle x_i, \beta \rangle \right)^2 + \lambda \sum_{i=1}^{d} |\beta_i|$$

# Regularized Logistic Regression

- Training examples: $\{x_i, y_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$
- $y_i \in \{+1, -1\}$
- Regularized logistic regression: $\widehat{\beta} \in \mathbb{R}^d$ minimises

$$F(\beta) = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + e^{-y_i \langle x_i, \beta \rangle}\right) + \lambda \sum_{i=1}^{d} \beta_i^2$$

- **[Exercise]** write down the gradient descent update for fitting a regularized logistic regression.
- **[Exercise]** the function $\beta \mapsto F(\beta)$ is convex.

# Stochastic Gradient Descent

- If $N \gg 1$, computing the gradient of $F$ can be expensive,

$$\nabla F(\beta) = \frac{1}{N} \sum_{i=1}^{N} \nabla \textbf{Loss}(\beta, x_i, y_i) \; + \; \nabla \Omega(\beta)$$

- stochastic gradient: replace $\frac{1}{N} \sum_{i=1}^{N}$ by $\frac{1}{n} \sum_{i=1}^{n}$ for $n \ll N$,

$$\widetilde{\nabla F}(\beta) = \frac{1}{n} \sum_{i \in I_n} \nabla \textbf{Loss}(\beta, x_i, y_i) \; + \; \nabla \Omega(\beta)$$

where $I_n$ is a (random) subset with $n$ elements.

# Stochastic Gradient descent (SGD)

- For a learning rate $\eta > 0$ the SGD algorithm is as follows,

$$x_{n+1} = x_n - \eta \widetilde{\nabla F}(x_n)$$