Distributed optimization for Machine Learning

School of Electrical and Computer Engineering

University of Tehran

Erfan Darzi

Lecture 0 - Background

erfandarzi@ut.ac.ir

What is this course about?

* Useful optimization tools for machine learning
* This is NOT a machine learning course
* Don’t expect to learn detailed ML
* This is NOT a classical optimization course
* We won’t cover many classical optimization results
* We cover some basics though
* Few weeks on optimization
* Some ML examples will be explained in details

# Prerequisites

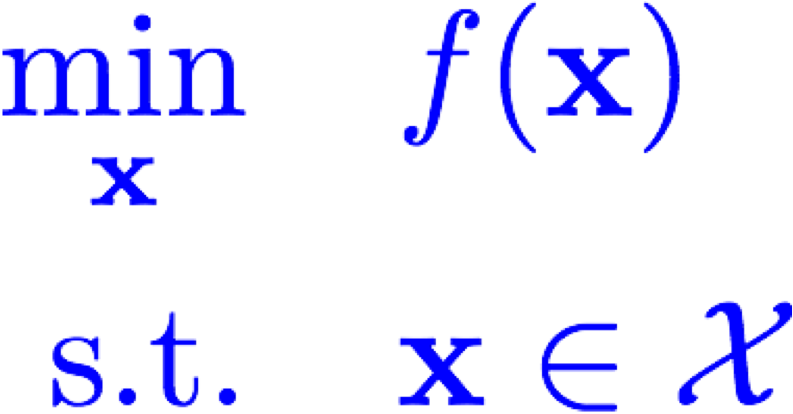
* Probability and Statistics
* Expected value, variance, statistical independence, conditional probability, maximum likelihood estimation, regression, etc.
* Linear Algebra and Mathematical Analysis
* Sets, functions, limits, liminf, limsup, derivative, gradient, subspace, linear dependence, inner product, eigenvalue, singular value, norms, etc.
* Programming skills
* Matrix/vector operations in Matlab/Python/C++
* “For, while, repeat until” loops

What is optimization?

Decision variable (Discrete/Continuous)

Objective/Cost

function



* Existence of a solution? Feasible Region
* Checking if a candidate *x* is optimal?

Why do we care?

* Many engineering problems requires optimization
* In this course, we focus mostly on machine learning applications

Can we use this dataset to predict the price of this house?

Samples/Data points

**Area**

**Crime Rate**

**Age**

**RAD**

**PTRATIO**

**Bedrooms**

**…**

**Price (K)**

600

1.05

12

2.4

10.1

1

…

500

1000

2.34

10

2.5

20.1

1

…

800

1200

1.45

3

3.1

9.7

3

…

1500

1500

1.56

30

1.7

7.2

2

…

1200

…

…

…

…

…

…

…

…

2700

1.01

20

0.9

4.3

4

…

5000

Features/independent Variables

Target/Dependent Variables/Label

**Area**

**Crime Rate**

**Age**

**RAD**

**PTRATIO**

**Bedrooms**

**…**

**Price (K)**

600

1.05

12

2.4

10.1

1

…

500

1000

2.34

10

2.5

20.1

1

…

800

1200

1.45

3

3.1

9.7

3

…

1500

1500

1.56

30

1.7

7.2

2

…

1200

…

…

…

…

…

…

…

…

2700

1.01

20

0.9

4.3

4

…

5000

Can we use this dataset to predict the price of this house?

Training Data

**Area**

**Crime Rate**

**Age**

**RAD**

**PTRATIO**

**Bedrooms**

**…**

**Price (K)**

600

1.05

12

2.4

10.1

1

…

500

1000

2.34

10

2.5

20.1

1

…

800

1200

1.45

3

3.1

9.7

3

…

1500

1500

1.56

30

1.7

7.2

2

…

1200

…

…

…

…

…

…

…

…

2700

1.01

20

0.9

4.3

4

…

5000

Training Data

Learning Algorithm

Prediction

# Learning Algorithms

* Various methods in ML
* Decision trees, deep learning, Bayes, empirical Bayes, linear regression,

logistic regression, …

* Many methods
* Model
* Minimize the loss/Maximize the likelihood

# Linear regression

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Area** | **Crime Rate** | **Age** | **RAD** | **PTRATIO** | **Bedrooms** | **…** | **Price (K)** |
| 600 | 1.05 | 12 | 2.4 | 10.1 | 1 | … | 500 |
| 1000 | 2.34 | 10 | 2.5 | 20.1 | 1 | … | 800 |
| 1200 | 1.45 | 3 | 3.1 | 9.7 | 3 | … | 1500 |
| 1500 | 1.56 | 30 | 1.7 | 7.2 | 2 | … | 1200 |
| … | … | … | … | … | … | … | … |
| 2700 | 1.01 | 20 | 0.9 | 4.3 | 4 | … | 5000 |

Can we use this dataset to predict the price of this house?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1400 | 2.2 | 3 | 3.1 | 7.6 | 2 | … | **????** |

# Linear regression

**Area**

**Crime Rate**

**Age**

**RAD**

**PTRATIO**

**Bedrooms**

**…**

**Price (K)**

600

1.05

12

2.4

10.1

1

…

500

1000

2.34

10

2.5

20.1

1

…

800

1200

1.45

3

3.1

9.7

3

…

1500

1500

1.56

30

1.7

7.2

2

…

1200

…

…

…

…

…

…

…

…

2700

1.01

20

0.9

4.3

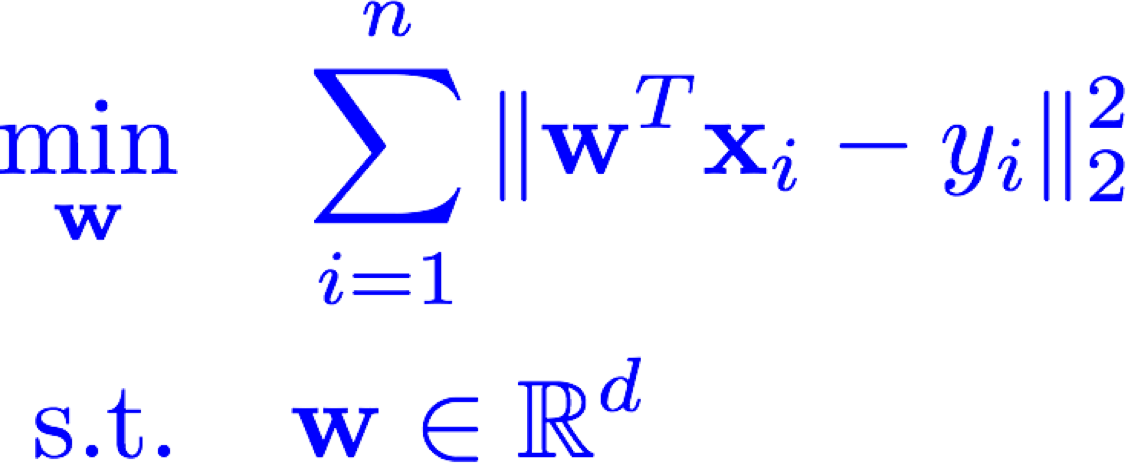
4

…

5000

Model: Linear predictor

Loss: L2 difference



Another Example: Classification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Radius** | **Texture** | **Area** | **Compactness** | **Symmetry** | **…** | **Rec/non-Rec** |
| 1.1 | 2.3 | 3.5 | 2.4 | 1.4 | … | 1 |
| 0.7 | 1.2 | 2.5 | 1.4 | 3.2 | … | 0 |
| 1.7 | 2.4 | 1.5 | 3.3 | 1.3 | … | 1 |
| … | … | … | … | … | … | … |
| 0.2 | 3.4 | 0.7 | 4.3 | 2.0 | … | 1 |
| 0.2 | 2.7 | 0.9 | 2.3 | 1.0 | … | **????** |

Logistic Regression

**Radius**

**Texture**

**Area**

**Compactness**

**Symmetry**

**…**

**Rec/non**

**-**

**Rec**

1.1

2.3

3.5

2.4

1.4

…

1

0.7

1.2

2.5

1.4

3.2

…

0

1.7

2.4

1.5

3.3

1.3

…

1

…

…

…

…

…

…

…

0.2

3.4

0.7

4.3

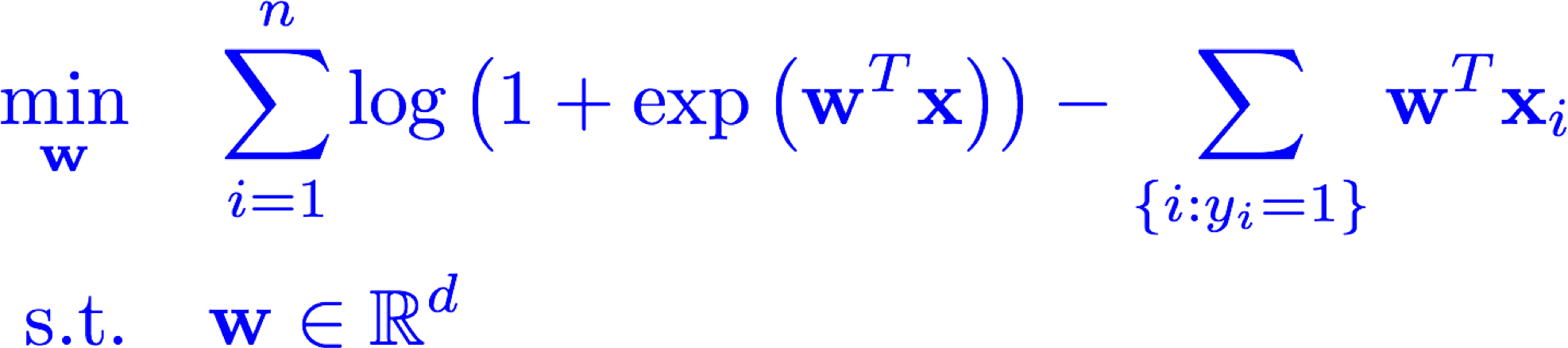
2.0

…

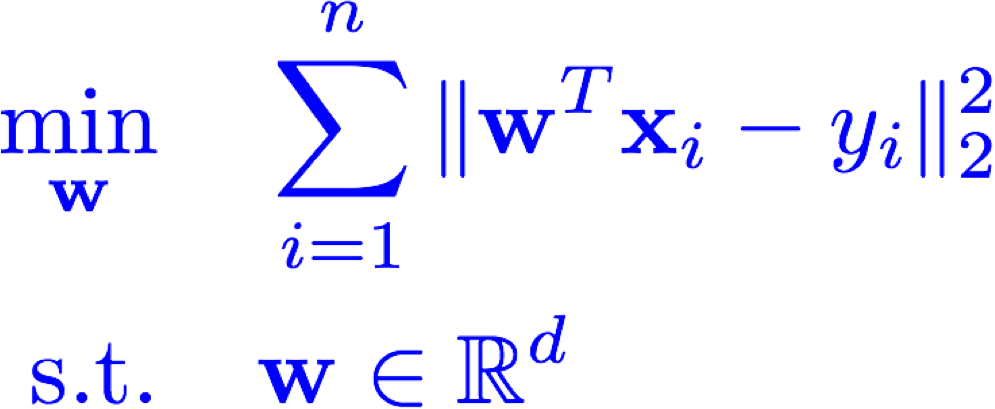
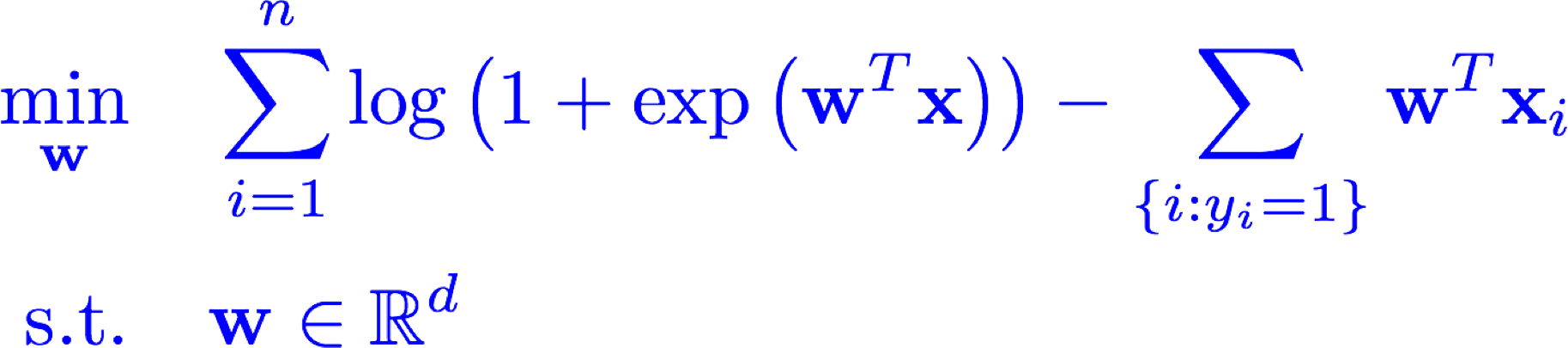
1

Model: logistic

Maximum likelihood estimator



Optimization in ML



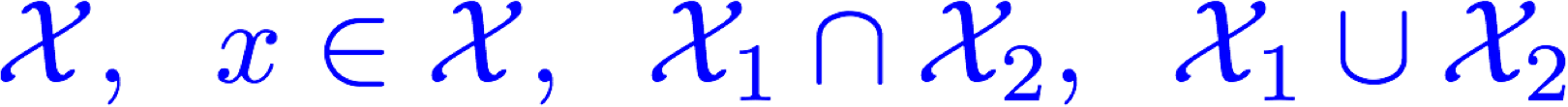
* Many more examples (K-means, SVM, Deep learning, …)
* Efficient algorithms: CPU, Memory requirements, Parallelizable, robustness, etc.
* Other issues: Non-convexity, Sparsity, Large values of *n/d*, Online implementation, Implicit bias, Privacy concerns, Overfitting, etc.
* But first, we need to review a little bit of optimization (targeted review!)
* Even before this, let’s review a bit of linear algebra and mathematical analysis

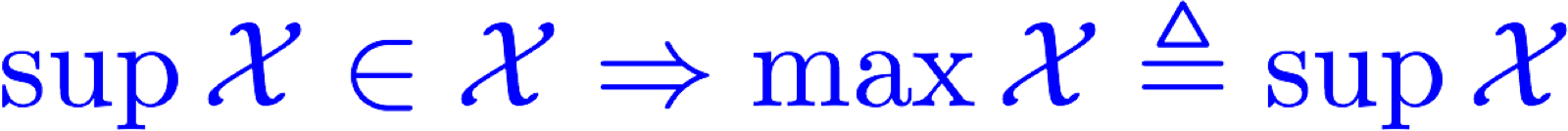
# Notations

* **Sets**

•

* Real numbers , Complex numbers

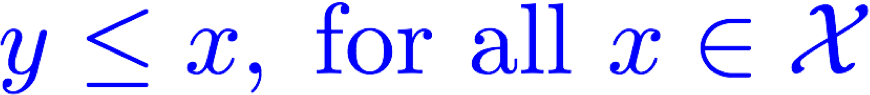
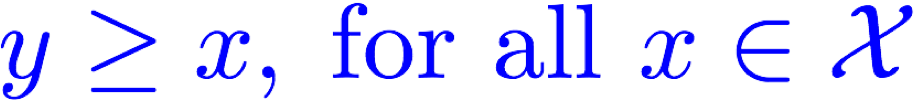


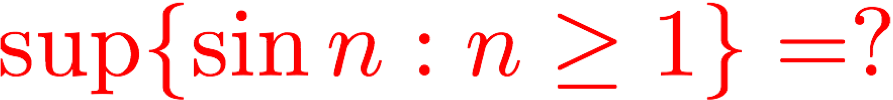
* **Inf and Sup**
* Supremum of the set is the smallest scalar • Infimum of the set is the largest scalar

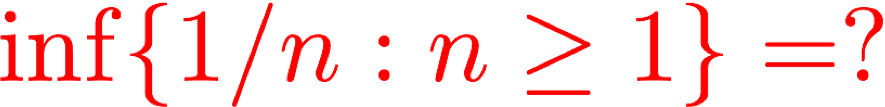
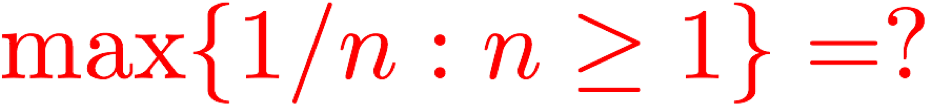
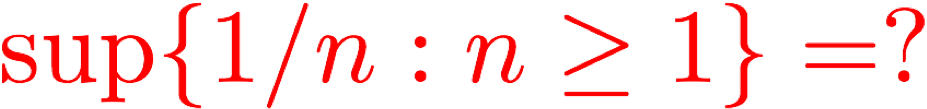


such that

such that



 inf{sin*n* : **Functions:**



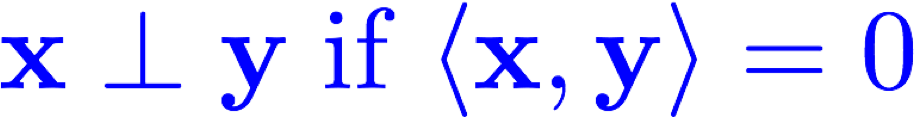
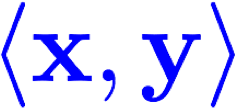


Vectors and Subspaces

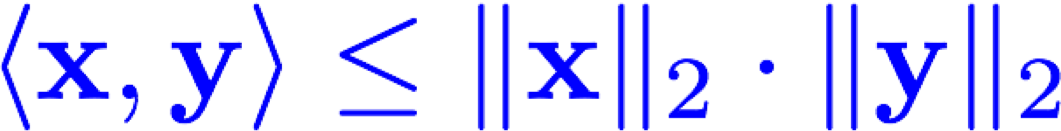
* **Linear combination:**



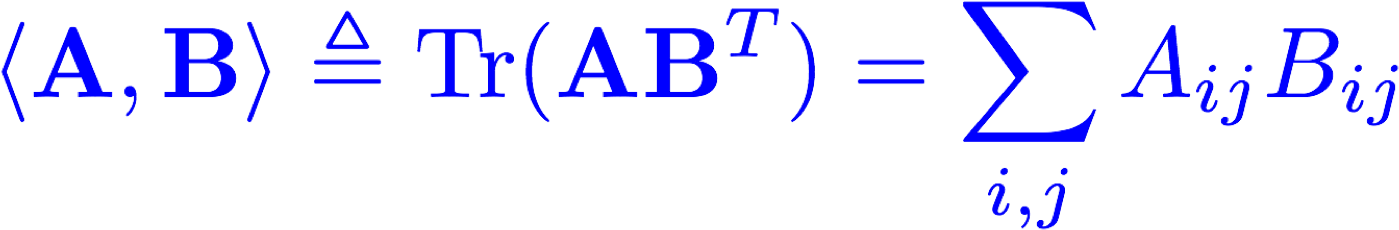
* **Subspace and linear independence**
* A set is called subspace if it is closed under linear combination
* A set of vectors is called linearly independent if no linear combination of them is equal to zero
* Inner product:



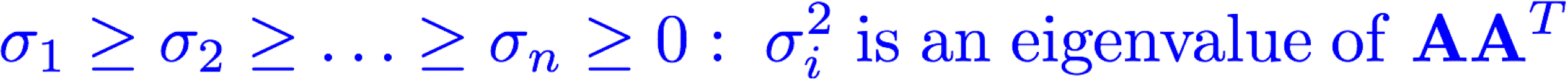
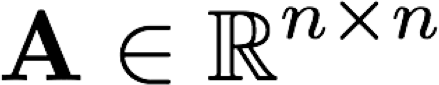
* Orthogonality:
* **Cauchy-Schwarz inequality**



# Matrices

* Matrix addition
* Matrix product
* Square matrix
* Inner product: 
* Spectral radius: 
* Eigenvalue decomposition of real symmetric matrices
* Positive (Semi-)definite matrices

# Matrices

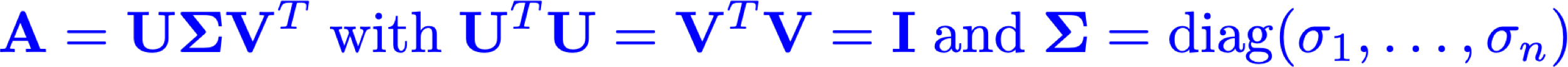
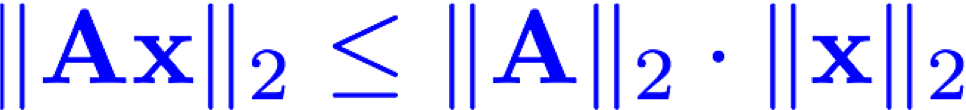
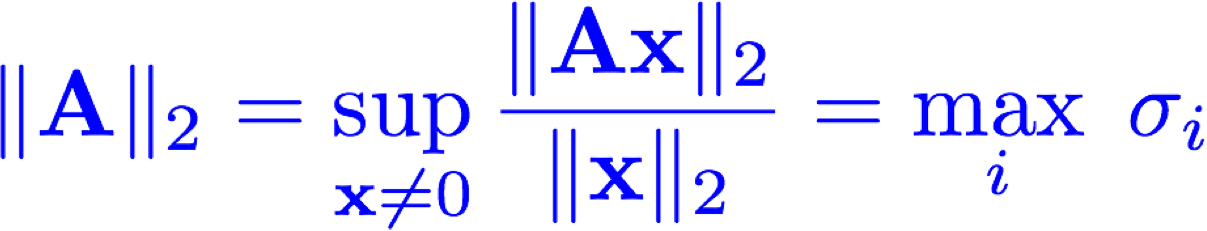
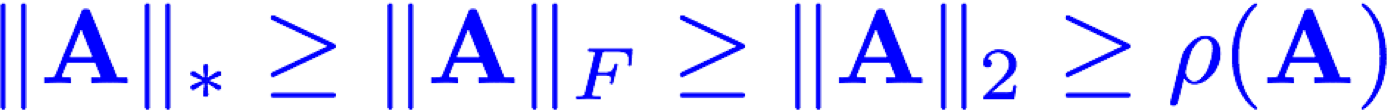
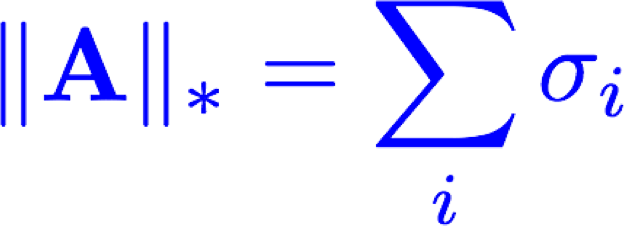
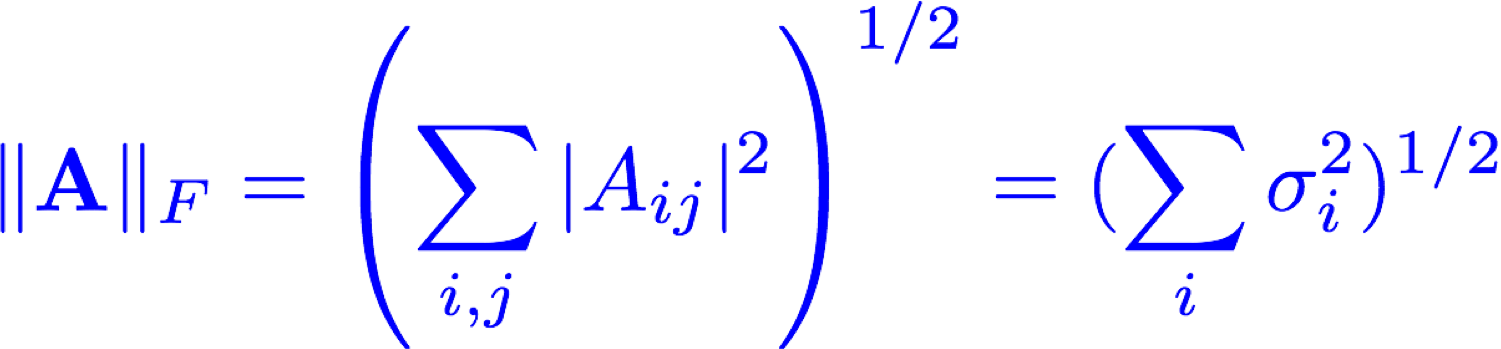
* Singular values: 
* Singular value decomposition of :
* Norms:

•

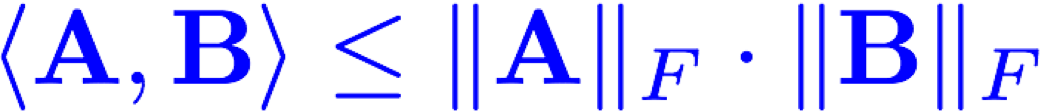
Nuclear norm:

•

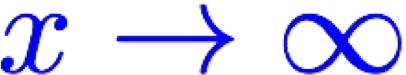
Useful inequalities:

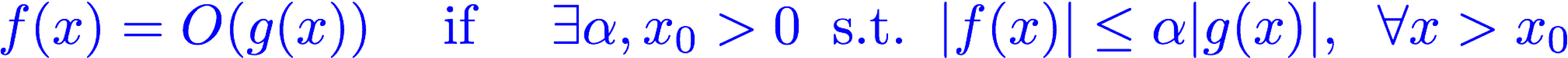


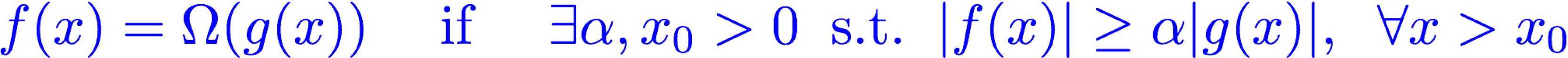
* Frobenius norm:
* Matrix 2-norm:

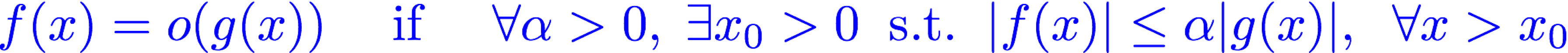


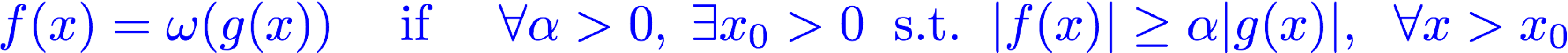
# Big Oh notations

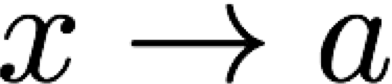
* Which one grows faster? Linear or quadratic?
* How to compare the limiting behavior of functions?
* When :



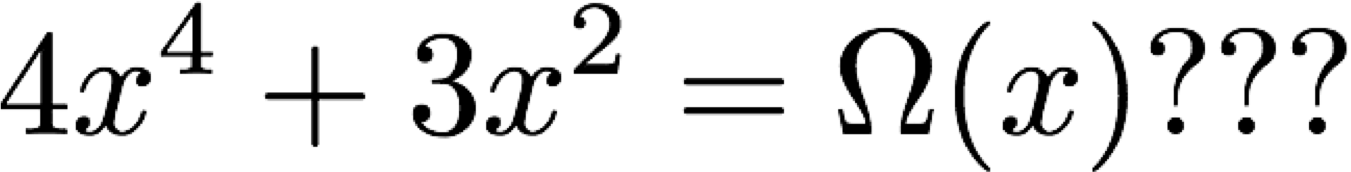
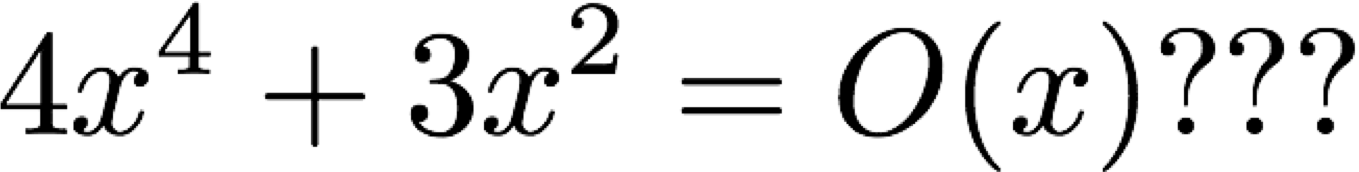
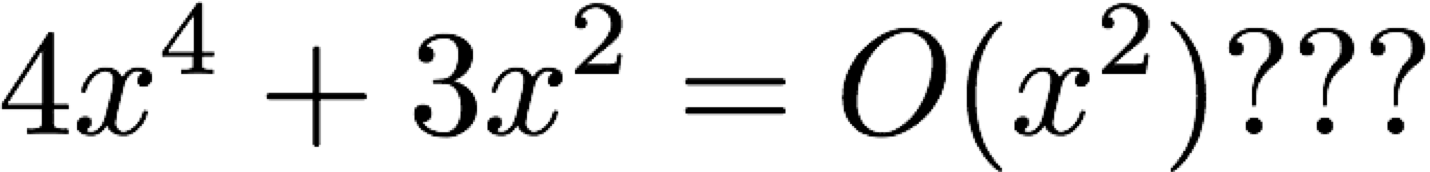
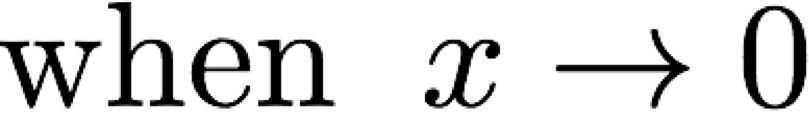
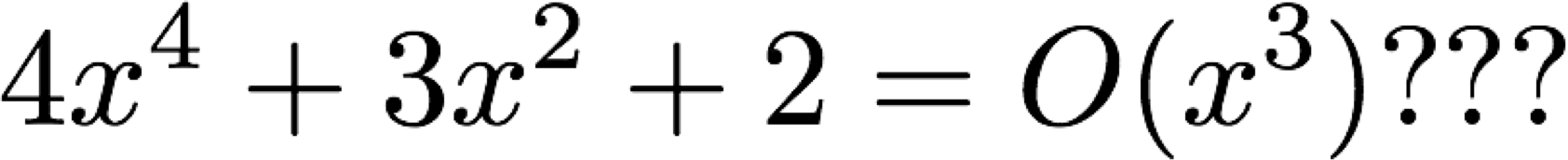
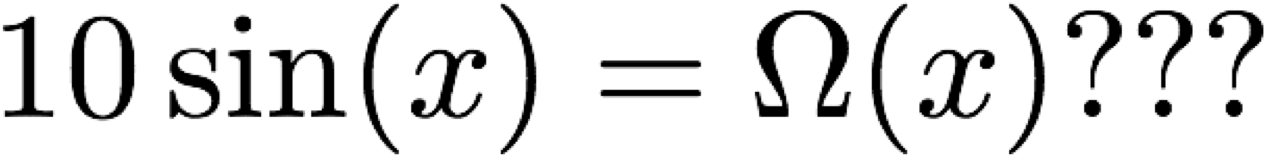
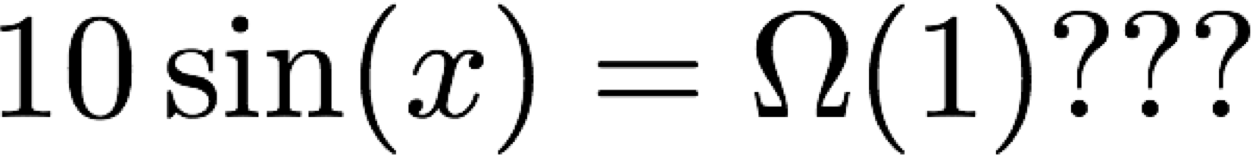
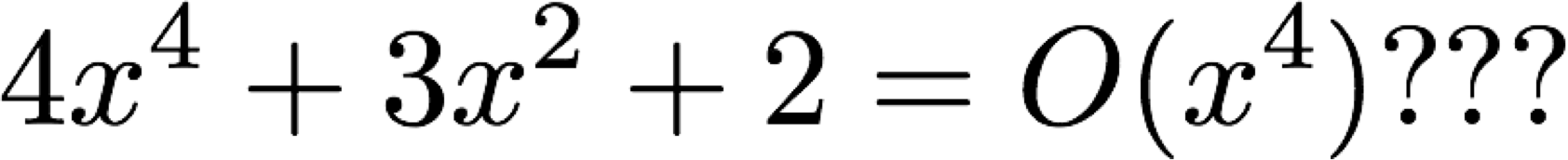
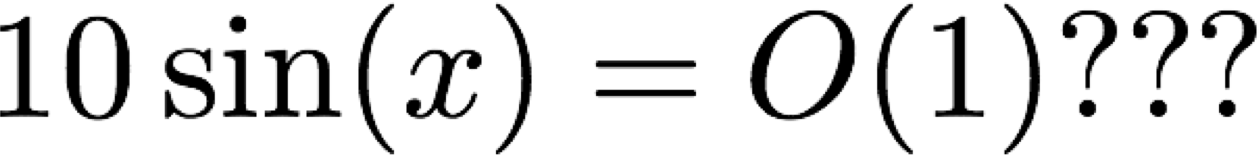
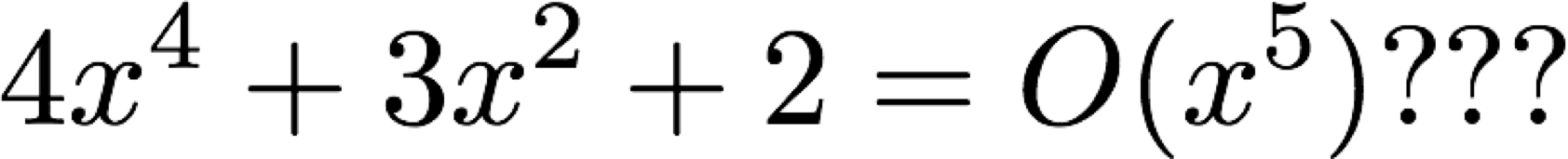




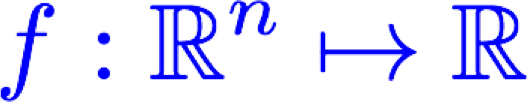


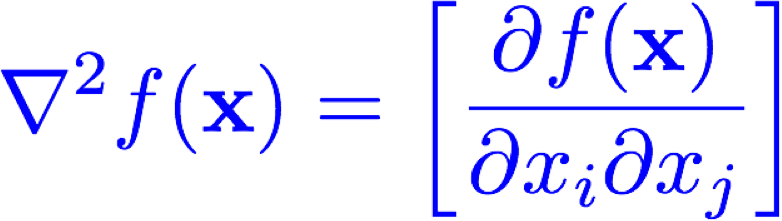
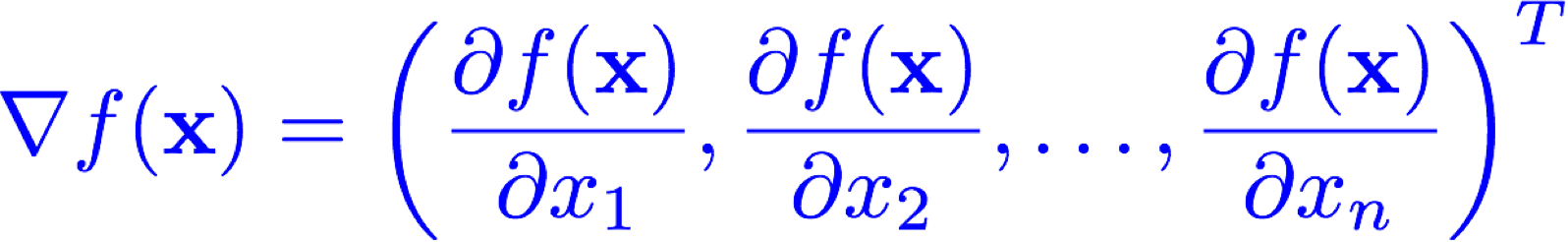
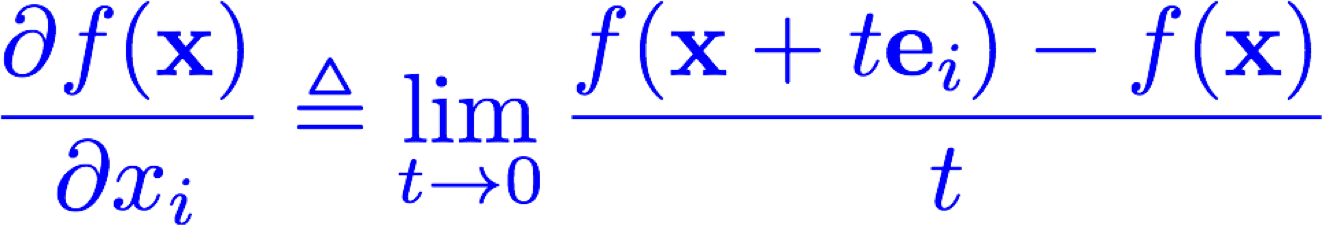
We can also define it for 

# Examples

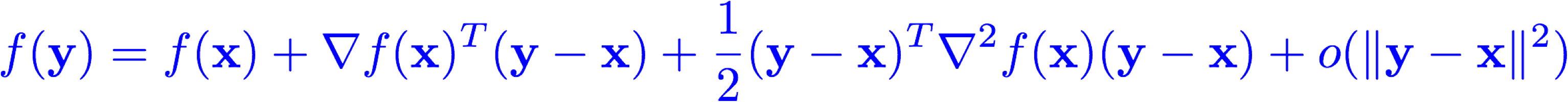


Derivatives

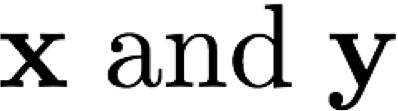
* Suppose is a twice continuously differentiable function
* Derivative:
* Gradient:
* Hessian Matrix:

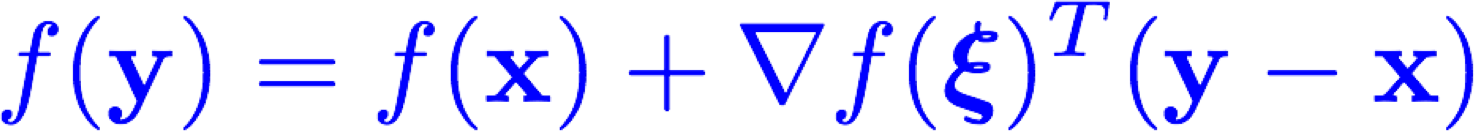


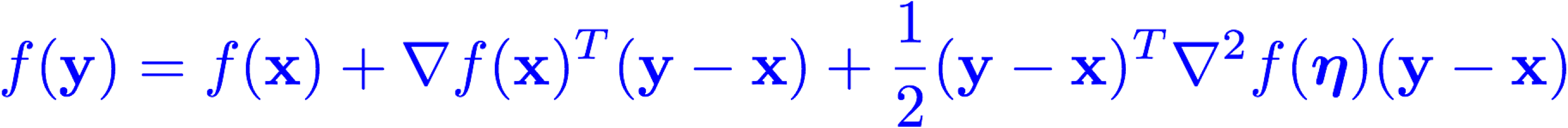
* Taylor Expansion:



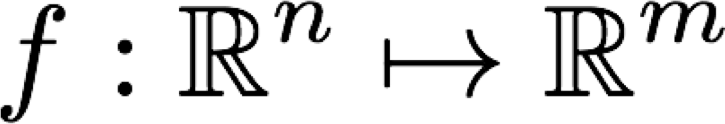
# Mean Value Theorem

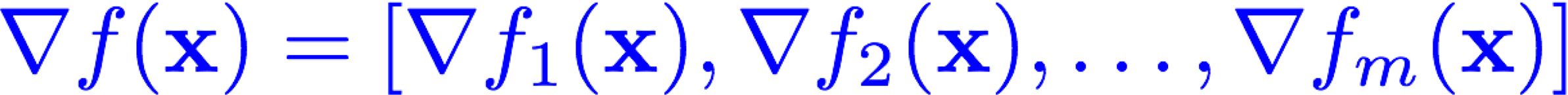
• There exists in the line segment connecting such that

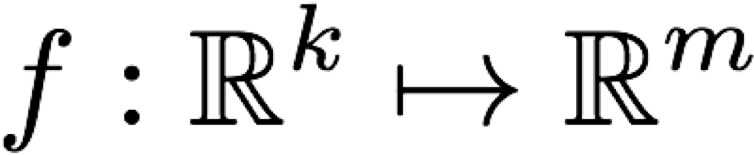
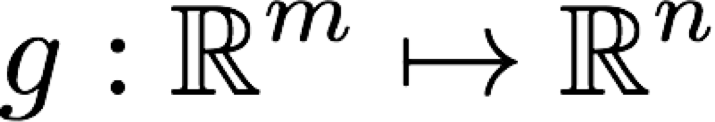
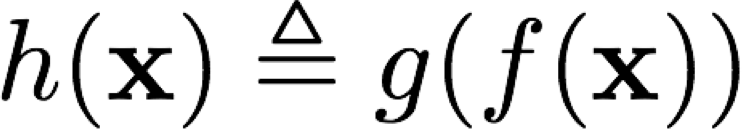


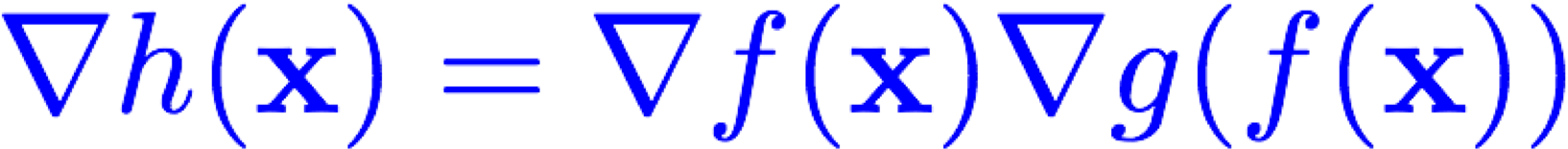


Chain Rule

**Jacobian Matrix for** 

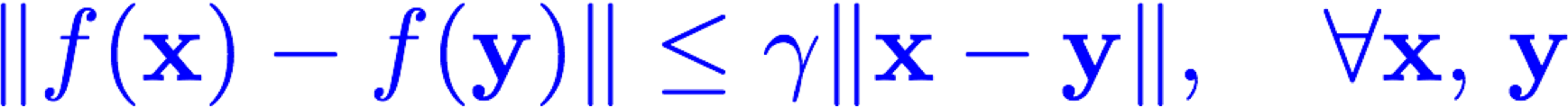


**Chain Rule:**   

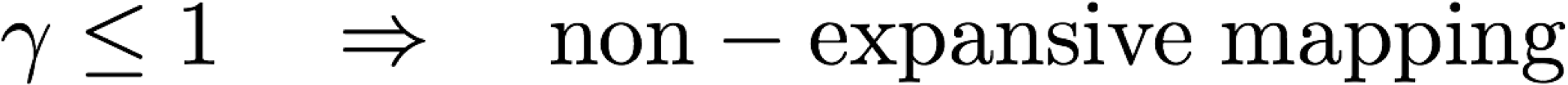


**Examples:**  

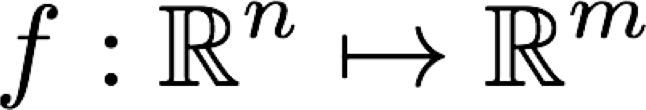
# Contraction Mappings

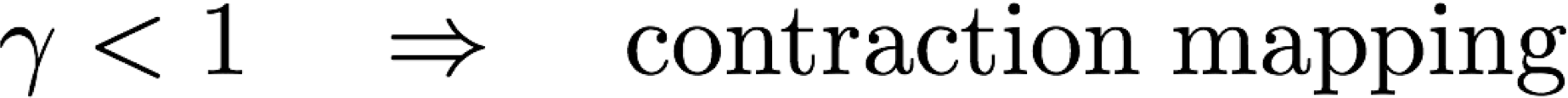


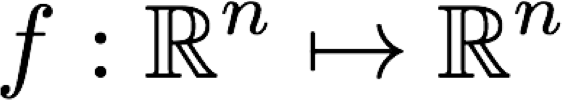
**Lipschitz constant**

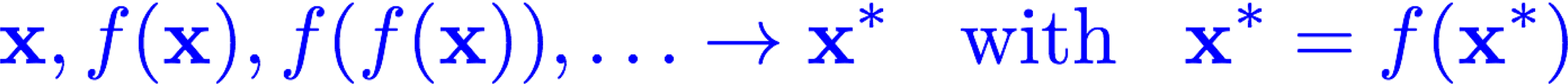


**Lipschitz Continuity:**





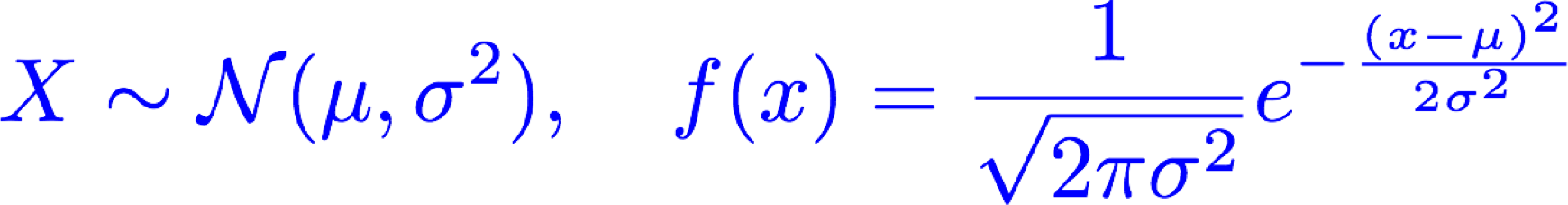
**Theorem:** For a contraction mapping , the iterated function sequence converges to a unique fixed point., i.e.,



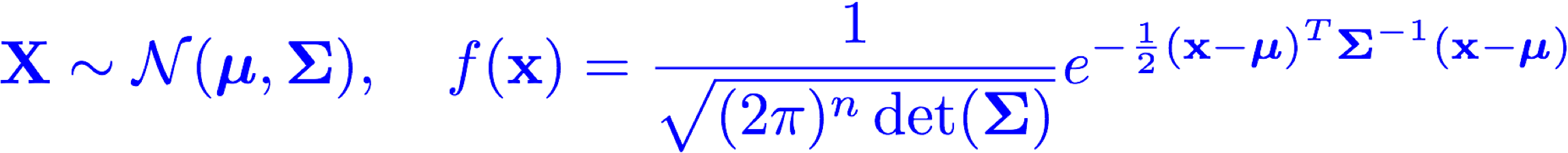
**True for non-expansive mappings???**

# Probability

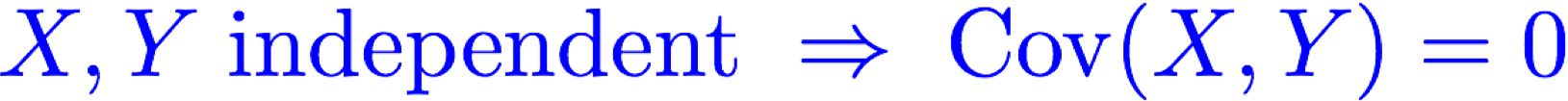
* Probability, Conditional probability, Random Variable, Independence
* Normal/Gaussian distribution



* Jointly multivariate Normal distribution

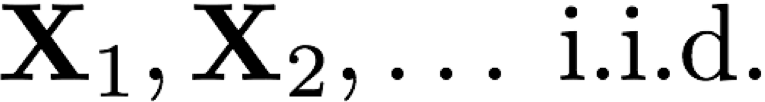
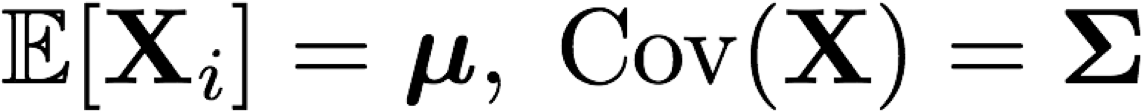
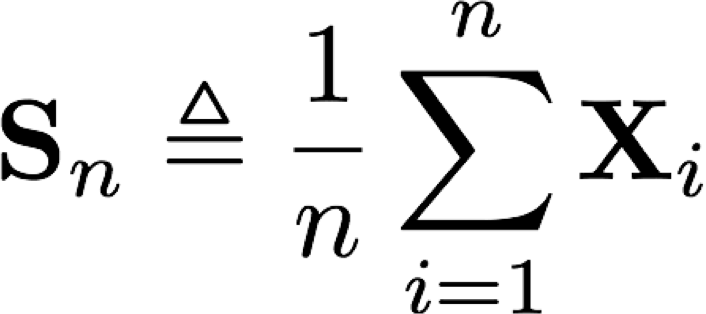


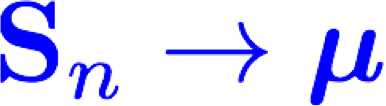
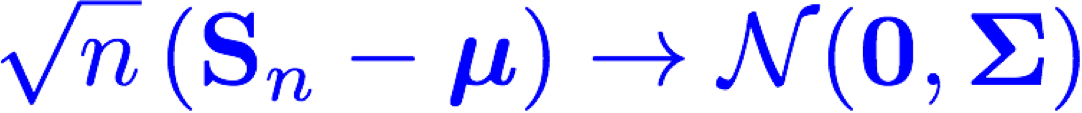
* Expected value, Variance, Covariance (Matrix)



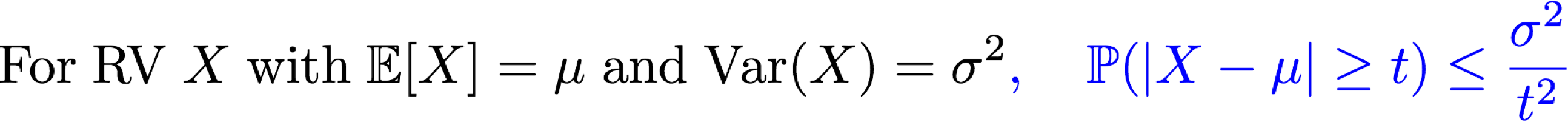
**Converse?**

# Probability

• For with and 

**Law of large numbers**  **Central Limit Theorem** 

**Markov’s inequality:**



**Chebyshev’s inequality:**

**Cauchy-Schwarz inequality:** 