EECS 545: Machine Learning
Linear Algebra Review

01/15/2020

Yijie Guo

Portions of this presentation adapted from
EECS 545 Winter 2010, Stanford CS229 Lecture Notes, CMU 10-701,
Delaware CISC 489/689 review slides.

Outline

- Basic Concepts and Notation
- Matrix Multiplication
- Operations and Properties
- Matrix Calculus

Outline

- **Basic Concepts and Notation**
- Matrix Multiplication
- Operations and Properties
- Matrix Calculus

Basic Concepts

- Why linear algebra?
  - Compact representation
  - Efficient representation
  - Tools like Matlab
  - Help appreciate Machine Learning

- Can represent:

$$4x_1 - 5x_2 = -13$$
$$-2x_1 - 3x_2 = 9$$

  As: $$Ax = b$$

  Where: $$A = \begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix}, b = \begin{bmatrix} -13 \\ 9 \end{bmatrix}$$

- Generalizes well to many machine learning contexts

Basic Notation

- $x \in \Re^n$ is a vector with n entries.
- x_i is the i^{th} entry of $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

Basic Notation

- $A \in \Re^{m \times n}$ is a matrix with m rows and n columns.
- a_{ij} or A_{ij} is the entry in the i^{th} row and j^{th}column of $A =$ $\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$
- a_j is the j^{th}column of A
- a_i^T is the i^{th}row of A

Transpose

- The transpose of a matrix $A \in \Re^{m \times n}$ is the matrix $A^T \in \Re^{n \times m}$, where:

$$(A^T)_{ij} = A_{ji}$$

$$\begin{pmatrix} a \\ b \end{pmatrix}^T = \begin{pmatrix} a & b \end{pmatrix}$$

- Properties of transposes
  - $(A^T)^T = A$
  - $(AB)^T = B^T A^T$
  - $(A + B)^T = A^T + B^T$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^T = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

- A matrix where $A = A^T$ is said to be *symmetric.*
- A matrix where $A = -A^T$ is said to be *anti-symmetric.*

Outline

- Basic Concepts and Notation
- **Matrix Multiplication**
- Operations and Properties
- Matrix Calculus

Vector-Vector Multiplication

- Inner product (dot product)

$$x^T y \in \Re = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i$$

- Outer product

$$xy^T \in \Re^{m \times n} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \cdots & x_m y_n \end{bmatrix}$$

  - Useful for compactly representing a matrix where all rows are a multiples of each other

Matrix-Vector Multiplication

- The product of a matrix $A \in \Re^{m \times n}$ and a vector in $x \in \Re^n$ is a vector $y \in \Re^m$, where

$$y = Ax = \begin{bmatrix} \leftarrow & a_1^T & \rightarrow \\ \leftarrow & a_2^T & \rightarrow \\ & \vdots & \\ \leftarrow & a_m^T & \rightarrow \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_1^T x \\ a_2^T x \\ \vdots \\ a_m^T x \end{bmatrix}$$

or

$$y_i = \sum_{k=1}^{n} a_{ik} x_k$$

Matrix-Matrix Multiplication

- The product of a matrix $A \in \Re^{m \times n}$ and a matrix in $B \in \Re^{n \times p}$ is a matrix $C \in \Re^{m \times p}$, where

$$C = AB = \begin{bmatrix} \leftarrow & a_1^T & \rightarrow \\ \leftarrow & a_2^T & \rightarrow \\ & \vdots & \\ \leftarrow & a_m^T & \rightarrow \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ b_1 & b_2 & \cdots & b_p \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_p \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_p \end{bmatrix}$$

or

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

- Notice, vector-vector and matrix-vector are special cases of matrix-matrix multiplication

Properties of Matrix Multiplication

- Matrix multiplication is associative:

$$(AB)C = A(BC)$$

- Matrix multiplication is distributive:

$$A(B + C) = AB + AC$$

- Matrix multiplication is ***not*** necessarily commutative:

$$AB \neq BA$$

Outline

- Basic Concepts and Notation
- Matrix Multiplication
- **Operations and Properties**
- Matrix Calculus

The Identity Matrix

- Identity is the matrix $I \in \Re^{n \times n}$, where:

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Or

$$I_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

- And has the property:

$$AI = A = IA$$

- Special case of the more general diagonal matrix D, where

$$D_{ij} = \begin{cases} d_i & i = j \\ 0 & i \neq j \end{cases}$$

Other Special matrices

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

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

$$\begin{pmatrix} a & b & 0 & 0 \\ c & d & e & 0 \\ 0 & f & g & h \\ 0 & 0 & i & j \end{pmatrix}$$ tri-diagonal

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

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ I (identity matrix)

The Inverse

- The inverse, A^{-1} , of a square matrix $A \in \Re^{n \times n}$ is the unique matrix such that
$$A^{-1}A = I = AA^{-1}$$
- A matrix A is ***invertible*** or ***non-singular*** if A^{-1} exists, and ***non-invertible*** or ***singular*** otherwise.
$$\forall A, B \in \Re^{n \times n}$$
  - $(A^{-1})^{-1} = A$
  - $(AB)^{-1} = B^{-1}A^{-1}$
  - $(A^{-1})^T = (A^T)^{-1}$
- If $Ax = b$, then $x = A^{-1}b$.

The Matrix Trace

- The trace of a square matrix $A \in \Re^{n\times n}$ is the sum of its diagonal elements:

$$\mathrm{trA} = tr(A) = \sum_{i=1}^{n} A_{ii}$$

- Properties:
  - $tr(A) = tr(A^T)$
  - $tr(A + B) = tr(A) + tr(B) \qquad (A, B \in \Re^{n\times n})$
  - $tr(tA) = t \cdot tr(A)$
  - $tr(AB) = tr(BA) \qquad (A \in \Re^{n\times m}, B \in \Re^{m\times n})$

Linear independence

- A set of vectors is linearly independent if none of them can be written as a linear combination of the others.
- Vectors $v_1,\ldots,v_k$ are linearly independent if $c_1v_1+\ldots+c_kv_k = 0$ implies $c_1=\ldots=c_k=0$

$$\begin{pmatrix} | & | & | \\ v_1 & v_2 & v_3 \\ | & | & | \end{pmatrix}\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

e.g.

$$\begin{pmatrix} 1 & 0 \\ 2 & 3 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

(u,v)=(0,0), i.e. the columns are linearly independent.

$$x_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad x_2 = \begin{bmatrix} 4 \\ 1 \\ 5 \end{bmatrix} \quad x_3 = \begin{bmatrix} 2 \\ -3 \\ -1 \end{bmatrix}$$

x3 = −2x1 + x2

Rank of a Matrix

- rank(A) (the rank of a m-by-n matrix A) is

  The maximal number of linearly independent columns

  =The maximal number of linearly independent rows

  =The dimension of col(A)

  =The dimension of row(A)

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 2 & 1 \\ 4 & 2 \end{pmatrix}$$

- If A is n by m, then
  - rank(A)<= min(m,n)
  - If n=rank(A), then A has full row rank
  - If m=rank(A), then A has full column rank

Linear Independence and Matrix Rank

- A set of vectors $\{x_1, x_2, \cdots, x_n\}$ is ***linearly independent*** if no vector can be written as a linear combination ($x_n = \sum_{i=1}^{n-1} \alpha_i x_i$) of the remaining vectors, and ***linearly dependent*** otherwise.
- The ***rank*** of a matrix is the largest number of linearly independent rows.
  - $rank(A) \le \min(m, n)$ (full rank if =) $\forall A \in \Re^{m \times n}$
  - $rank(A) = rank(A^T)$ $\forall A \in \Re^{m \times n}$
  - $rank(A + B) \le \mathrm{rank(A)} + \mathrm{rank(B)}$ $\forall A, B \in \Re^{m \times n}$
  - $rank(AB) \le \min\big(rank(A), rank(B)\big)$ $\forall A \in \Re^{m \times n}, B \in \Re^{n \times p}$

Orthogonal and Normal Matrices

- Two vectors are *orthogonal* if $x^T y = 0$
- A vector is said to be *normalized* if $\|x\|_2 = 1$
- A square matrix $U \in \Re^{n \times n}$ is orthogonal if all columns are normalized and orthogonal to each other

$$U^T U = I = U U^T$$

or

$$U^{-1} = U^T$$

Quadratic Forms

- Given a square matrix $A \in \Re^{n \times n}$ and a vector $x \in \Re^n$, the scalar value $x^T A x$ is called a quadratic form. Written explicitly, we see that

$$x^T A x = \sum_{i=1}^{n} x_i (Ax)_i = \sum_{i=1}^{n} x_i \left(\sum_{j=1}^{n} A_{ij} x_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j$$

$$x^T A x = (x^T A x)^T = x^T A^T x = x^T \left(\frac{1}{2} A + \frac{1}{2} A^T \right) x,$$

Positive Semidefinite Matrices

- A symmetric matrix $A \in S^n$ is **positive definite** (PD), usually denoted A $\succ$ 0 (or just A > 0), if for all non-zero vectors $x \in R^n, x^T Ax > 0$.
- A symmetric matrix $A \in S^n$ is **positive semidefinite** (PSD), denoted $A \geq 0$, if for all vectors $x \in R^n, x^T Ax \geq 0$.
- A symmetric matrix $A \in S^n$ is **negative definite** (ND), denoted $A < 0$, if for all non-zero vectors $x \in R^n, x^T Ax < 0$.
- A symmetric matrix $A \in S^n$ is **negative semidefinite** (NSD), denoted $A \leq 0$, if for all non-zero vectors $x \in R^n, x^T Ax \leq 0$.
- Finally, a symmetric matrix $A \in S^n$ is **indefinite**, if it is neither positive semidefinite nor negative semidefinite.

Outline

- Basic Concepts and Notation
- Matrix Multiplication
- Operations and Properties
- **Matrix Calculus**

Matrix Calculus

- First derivative: The Gradient
- Second derivative: The Hessian

The Gradient

- Suppose that $f : R^{m\times n} \rightarrow R$ is a function that takes as input a matrix A of size m × n and returns a real value (scalar). Then the gradient of f (with respect to $A \in R^{m\times n}$) is the matrix of partial derivatives, defined as:

$$\nabla_A f(A) \in \mathbb{R}^{m\times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

$$(\nabla_A f(A))_{ij} = \frac{\partial f(A)}{\partial A_{ij}}.$$

The Gradient

Note that the size of $\nabla_A f(A)$ is always the same as the size of A. So if, in particular, A is just a vector $x \in \mathbb{R}^n$,

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

- $\nabla_x(f(x) + g(x)) = \nabla_x f(x) + \nabla_x g(x)$.
- For $t \in \mathbb{R}$, $\nabla_x(t\, f(x)) = t\nabla_x f(x)$.

Gradient of Linear Functions

$$f(x) = \sum_{i=1}^{n} b_i x_i$$

$$\frac{\partial f(x)}{\partial x_k} = \frac{\partial}{\partial x_k} \sum_{i=1}^{n} b_i x_i = b_k.$$

$$\nabla_x b^T x = b$$

The Hessian

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a function that takes a vector in $\mathbb{R}^n$ and returns a real number. Then the ***Hessian*** matrix with respect to x, written $\nabla_x^2 f(x)$ or simply as H is the $n \times n$ matrix of partial derivatives,

$$\nabla_x^2 f(x) \in \mathbb{R}^{n\times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

The Hessian

In other words, $\nabla_x^2 f(x) \in \mathbb{R}^{n \times n}$, with

$$(\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

Note that the Hessian is always symmetric, since

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}.$$

Similar to the gradient, the Hessian is defined only when $f(x)$ is real-valued.

Hessians of Quadratic Functions

$$\frac{\partial^2 f(x)}{\partial x_k \partial x_\ell} = \frac{\partial}{\partial x_k}\left[\frac{\partial f(x)}{\partial x_\ell}\right] = \frac{\partial}{\partial x_k}\left[\sum_{i=1}^{n} A_{\ell i} x_i\right] = 2A_{\ell k} = 2A_{k\ell}$$

$$\nabla_x^2 x^T A x = 2A$$

Gradients and Hessians of Quadratic and Linear Functions (Recap)

- $\nabla_x b^T x = b$
- $\nabla_x x^T A x = 2Ax$ (if A symmetric)
- $\nabla_x^2 x^T A x = 2A$ (if A symmetric)

Linear Regression

- Main idea:
  - Compute gradient and set gradient to 0. (condition for optimal solution)
  - Solve the equation in a closed form
- Objective function:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(\sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)} \right)^2$$

- We will derive the gradient from matrix calculus

Linear Regression

- The design matrix is an NxM matrix, applying
  - the M basis functions (columns)
  - to N data points (rows)

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}^{(1)}) & \phi_1(\mathbf{x}^{(1)}) & \dots & \phi_{M-1}(\mathbf{x}^{(1)}) \\ \phi_0(\mathbf{x}^{(2)}) & \phi_1(\mathbf{x}^{(2)}) & \dots & \phi_{M-1}(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}^{(N)}) & \phi_1(\mathbf{x}^{(N)}) & \dots & \phi_{M-1}(\mathbf{x}^{(N)}) \end{pmatrix}$$

$$\Phi\mathbf{w} \approx \mathbf{y}$$

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}^{(1)}) & \phi_1(\mathbf{x}^{(1)}) & \dots & \phi_{M-1}(\mathbf{x}^{(1)}) \\ \phi_0(\mathbf{x}^{(2)}) & \phi_1(\mathbf{x}^{(2)}) & \dots & \phi_{M-1}(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}^{(N)}) & \phi_1(\mathbf{x}^{(N)}) & \dots & \phi_{M-1}(\mathbf{x}^{(N)}) \end{pmatrix}$$

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{n=1}^{N}(\sum_{j=0}^{M-1} w_j\phi_j(\mathbf{x}^{(n)}) - y^{(n)})^2 \\ &= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) - y^{(n)})^2 \\ &= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}))^2 - \sum_{n=1}^{N} y^{(n)}\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + \frac{1}{2}\sum_{n=1}^{N} y^{(n)2} \end{aligned}$$

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}^{(1)}) & \phi_1(\mathbf{x}^{(1)}) & \dots & \phi_{M-1}(\mathbf{x}^{(1)}) \\ \phi_0(\mathbf{x}^{(2)}) & \phi_1(\mathbf{x}^{(2)}) & \dots & \phi_{M-1}(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}^{(N)}) & \phi_1(\mathbf{x}^{(N)}) & \dots & \phi_{M-1}(\mathbf{x}^{(N)}) \end{pmatrix}$$

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{n=1}^{N}(\sum_{j=0}^{M-1} w_j\phi_j(\mathbf{x}^{(n)}) - y^{(n)})^2 \\ &= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) - y^{(n)})^2 \\ &= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}))^2 - \sum_{n=1}^{N} y^{(n)}\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + \frac{1}{2}\sum_{n=1}^{N} y^{(n)2} \\ &= \frac{1}{2}\mathbf{w}^T\Phi^T\Phi\mathbf{w} - \mathbf{w}^T\Phi^T\mathbf{y} + \frac{1}{2}\mathbf{y}^T\mathbf{y} \end{aligned}$$

Linear Regression

- Objective function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)})^2$$

Linear Regression

- Objective function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\sum_{j=0}^{M-1} w_j\phi_j(\mathbf{x}^{(n)}) - y^{(n)})^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) - y^{(n)})^2$$

Linear Regression

- Objective function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}^{(n)}) - y^{(n)})^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) - y^{(n)})^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}))^2 - \sum_{n=1}^{N} y^{(n)}\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + \frac{1}{2}\sum_{n=1}^{N} y^{(n)2}$$

Linear Regression

- Objective function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\sum_{j=0}^{M-1} w_j\phi_j(\mathbf{x}^{(n)}) - y^{(n)})^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) - y^{(n)})^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}))^2 - \sum_{n=1}^{N} y^{(n)}\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + \frac{1}{2}\sum_{n=1}^{N} y^{(n)2}$$

$$= \frac{1}{2}\mathbf{w}^T\Phi^T\Phi\mathbf{w} - \mathbf{w}^T\Phi^T\mathbf{y} + \frac{1}{2}\mathbf{y}^T\mathbf{y}$$

- Trick: vectorization (by defining data matrix)

Linear Regression

- Compute gradient and set to zero

$$
\begin{aligned}
\nabla_{\mathbf{w}} E(\mathbf{w}) &= \nabla_{\mathbf{w}} \left(\frac{1}{2}\mathbf{w}^T \Phi^T \Phi \mathbf{w} - \mathbf{w}^T \Phi^T \mathbf{y} + \frac{1}{2}\mathbf{y}^T \mathbf{y} \right) \\
&= \Phi^T \Phi \mathbf{w} - \Phi^T \mathbf{y} \\
&= 0
\end{aligned}
$$

- Solve the resulting equation (normal equation)

$$\Phi^T \Phi \mathbf{w} = \Phi^T \mathbf{y}$$

$$\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

This is the *Moore-Penrose pseudo-inverse*: $\mathbf{\Phi}^\dagger = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T$

applied to: $\Phi \mathbf{w} \approx \mathbf{y}$