LECTURE 4: APPLICATIONS

Unconstrained optimization models for machine learning

1. Data clustering
2. Least squares estimation
3. Linear regression

Machine learning and optimization

1. Many machine learning problems are formulated as minimization of some loss function that measures discrepancy between the predictions of the model being trained and the actual problem instances, or as maximization of some reward function that affirms an expected decision.

2. One major difference between machine learning and optimization lies in the goal of generalization - optimization intends to minimize the loss/maximize the reward on a set of seen examples while machine learning is concerned with minimizing the loss/maximizing the reward on unseen samples.

Gradient Functions Commonly Used

Gradient functions: ($\nabla f(x)$ is represented as a column vector)

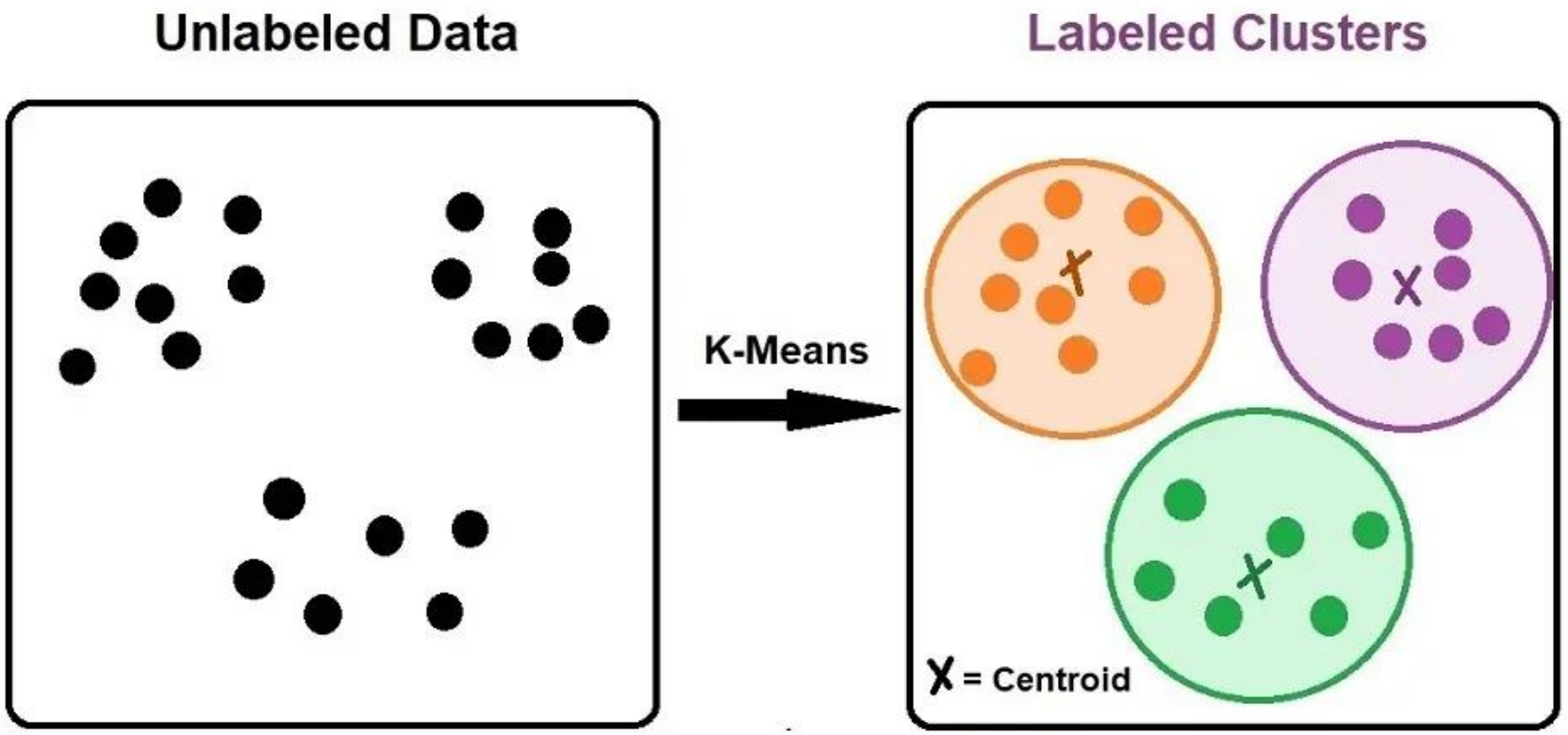
1. $\nabla(c^T x) = c, \qquad \forall c, x \in E^n$ $\qquad [(cx)' = c]$
2. $\nabla(x^T M x) = (M + M^T)x, \qquad \forall M \in E^{n\times n}, x \in E^n$ $\qquad [(mx^2)' = 2mx]$
3. $\nabla\|x\| = \frac{x}{\|x\|}, \qquad \forall x \in E^n$ $\qquad [\left(\sqrt{x^2}\right)' = \frac{x}{\sqrt{x^2}}]$
4. $\nabla\|x\|^2 = 2\|x\|(\nabla\|x\|) = 2x, \qquad \forall x \in E^n$ $\qquad [(x^2)' = 2x]$
5. $\nabla\|Ax - b\|^2 = \nabla[(Ax - b)^T(Ax - b)] = \nabla[\mathrm{x}^{\mathrm{T}} A^T A x - 2b^T A x + b^T b\,]$

   $= \nabla x^T (A^T A)x - \nabla(2b^T A x) = (A^T A + (A^T A)^T)x - 2A^T b$

   $= 2(A^T A)x - 2A^T b, \qquad \forall A \in E^{m\times n}, x \in R^n, b \in E\hat{}m$

Data clustering

- https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/k-means-clustering/



Centroid-based clustering

Data set: $S = \{x^i \in E^n : i = 1, \dots, N\}$

Centroid model

Find $x \in E^n$ such that the total distance to x^i ($i.e.$, $\sum_{i=1}^{N} \|x^i - x\|$) is minimized.

Quadratic optimization problem:

$$\text{Min} \sum_{i=1}^{N} \|x^i - x\|^2 \Leftrightarrow \nabla \sum_{i=1}^{N} \|x^i - x\|^2 = 0$$

$$\Leftrightarrow \sum_{i=1}^{N} (x^i - x) = 0$$

$$\Leftrightarrow x^* = \frac{1}{N} \sum_{i=1}^{N} x^i$$

Centroid of S is the mean of data points contained.

Centroid-based Clustering

Organize data points in S as $k > 1$ clusters such that

(i) Each data point falls in one and only one cluster

(ii) The total intra-cluster distance is minimized

Optimization problem:

Find k centroids $\{\mu^j \in E^n | \, j = 1, .., k\}$ *to*

Minimize $\sum_{j=1}^{k} \sum_{i=1}^{N} w_{ij} \left\| x^i - \mu^j \right\|^2$

where $w_{ij} = \begin{cases} 1, & if \ x^i \ is \ assigned \ \ to \ the \ j^{th} cluster \\ 0, & if \ x^i \ is \ not \ assigned \ to \ j^{th} \ cluster \end{cases}$

Heuristic Algorithm

k-means algorithm (when k is known)

Step 0: (Initial Step)

Randomly assign k points $\mu^j \in E^n, j = 1, .., k$, as the current centroids

Step 1: (Assign data points to clusters)

For each data point x^i, computer $\|x^i - \mu^j\|^2$ for each current centroid μ^j

assign x^i to the cluster with the shortest distance to its centroid

Step 2: (Update centroids of clusters)

For each cluster with assigned data points, compute its centroid μ^j_{new}

as the mean of its data points

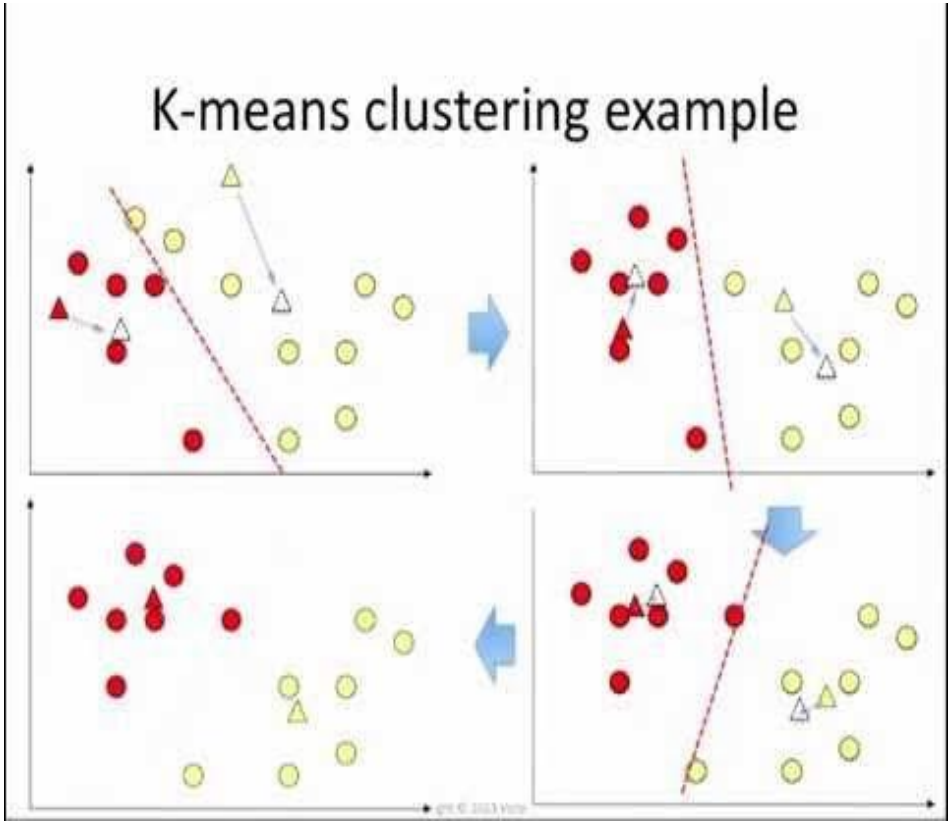
Step 3: (Stop)

Stop the algorithm if $\mu^j \sim \mu^j_{new}$ for $j = 1, .., k$

Otherwise, update $\mu^j \leftarrow \mu^j_{new}$ and go to Step 1

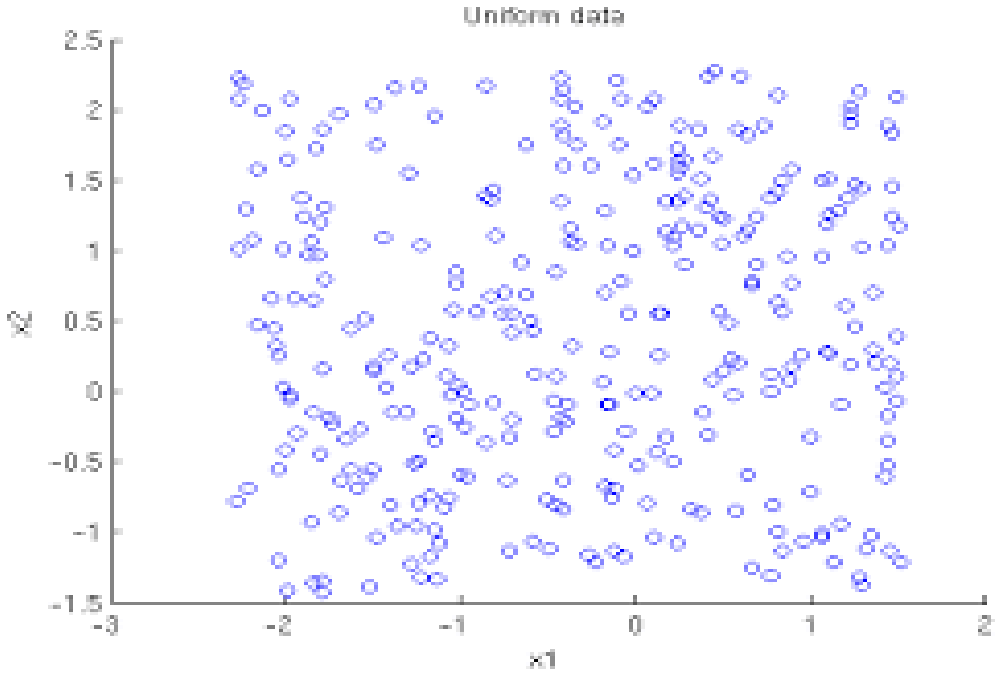
k-means Clustering Algorithm

- http://bit.ly/K-means



Best Number of Clusters

- $k = ?$



- https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSpDSnV0r9ilcndpK6lslr-8VvZvvDmeAki6_u1Qj0ctfbV2i_k84TZRPPsiO4rlUAOdiI&usqp=CAU

Best Number of Clusters

When the number of clusters k^* is to be determined

Observation: The total intra-cluster distance reduces as the number of clusters increases, for instance, the distance is zero when $k = N$.

Elbow rule method:

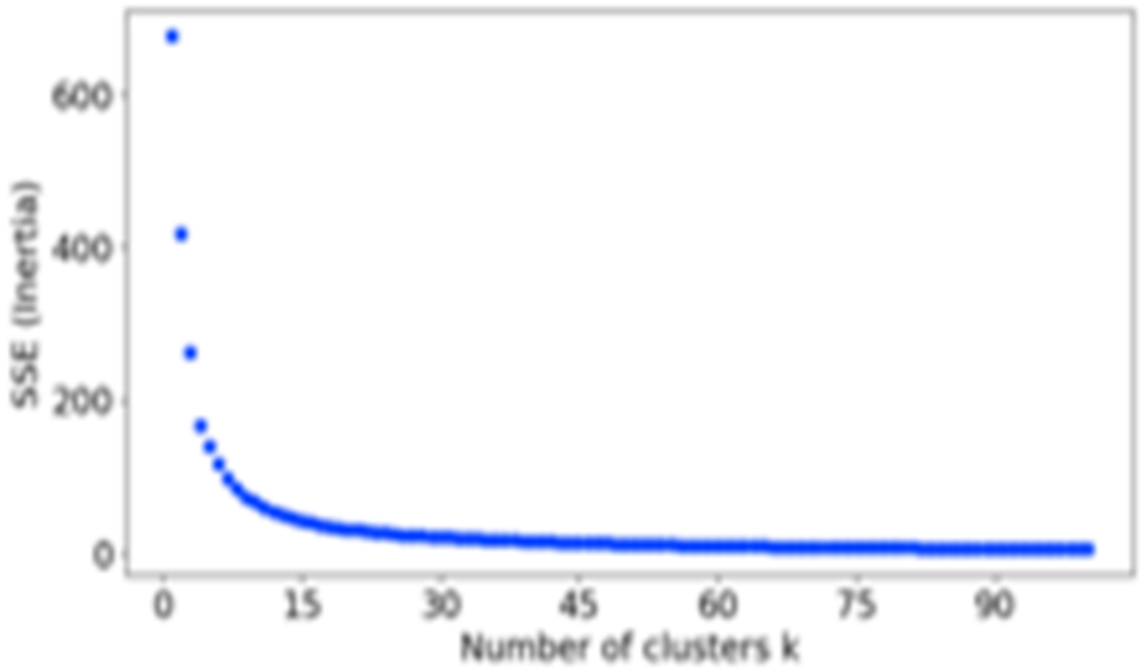
Step 1: Use the k-means algorithm to find the intra-cluster distance d_k for $k = 2, 3, 4, \ldots$

Step 2: Graph the intra-cluster distance d_k against the number of clusters k

Step 3: Find the elbow point k^* of the graph

Elbow Rule Method

- Example



Least Squares Estimation

Least squares estimation model

to find an approximate solution of $x \in E^n$ that has n variables

satisfying $m\ (> n)\ equations$

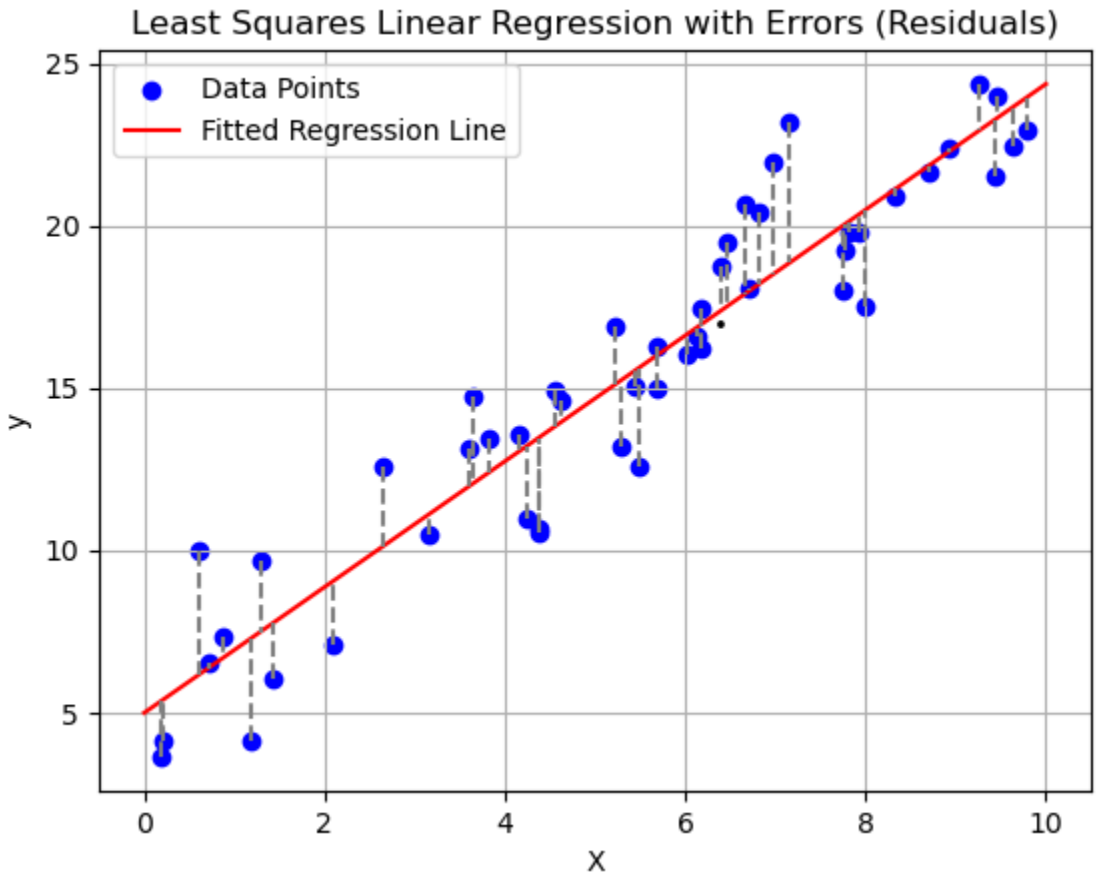
$$\mathrm{Min}_{x \in E^n} \|Ax - b\|^2 \Leftrightarrow \nabla \|Ax - b\|^2 = 0$$

$$\Leftrightarrow \quad (A^T A)x = A^T b \qquad \text{(Normal Equation)}$$

$$\Leftrightarrow \quad x = (A^T A)^{-1} A^T b, \qquad \forall\ A \in E^{m \times n} \text{ and } A \text{ is of full (column) rank}$$

1. The original linear system has no feasible solutions, so we try to find an approximate solution that is "almost feasible."
2. Inverting $A^T A$ matrix is of $O(n^3)$ complexity.
3. When A is of full column rank, $A^T A$ is positive semi-definite.
4. Normal equation may be solved by matrix decompositions such as Cholesky factorization ($L^T L$) method.

Least Squares Linear Regression



https://medium.com/physics-and-machine-learning/misconceptions-about-least-square-regression-1131841d240f

Linear Regression

Linear regression model

Data: $S = \{(x^i, y^i) : i = 1, \dots, N\}$, for $x^i \in E^n, y^i \in E^1$

Model: $y = m^T x + b + \epsilon$, $(i.e.,\ y \sim m^T x + b)$, for $m, x \in E^n, b \in E^1$

Error: $e_i = m^T x^i + b - y^i$

Least-squares problem:

$$\mathrm{Min}_{m,b} \sum_{i=1}^{N} e_i^2 = \mathrm{Min}_{m \in E^n, b \in E^1} \sum_{i=1}^{N} (m^T x^i + b - y^i)^2$$

Reformulation

Least-squares problem:

$$\mathrm{Min}_{m,b} \sum_{i=1}^{N} e_i^2 = \mathrm{Min}_{m \in E^n, b \in E^1} \sum_{i=1}^{N} (m^T x^i + b - y^i)^2$$

Arrange Data: $A = \begin{pmatrix} \frac{(x^1)^T}{\vdots} & \frac{1}{\vdots} \\ \frac{}{(x^N)^T} & \frac{}{1} \end{pmatrix} \in E^{N \times (n+1)}, z = \begin{pmatrix} m \\ b \end{pmatrix} \in E^{n+1}, y = \begin{pmatrix} y^1 \\ \vdots \\ y^N \end{pmatrix} \in E^N$

Reformulation problem: $\mathrm{Min}_{z \in E^{n+1}} \|Az - y\|^2$

Normal Equation: $(A^T A)z = A^T y$

Data Manipulation

Data manipulation: Divide by N on both sides for average effect:

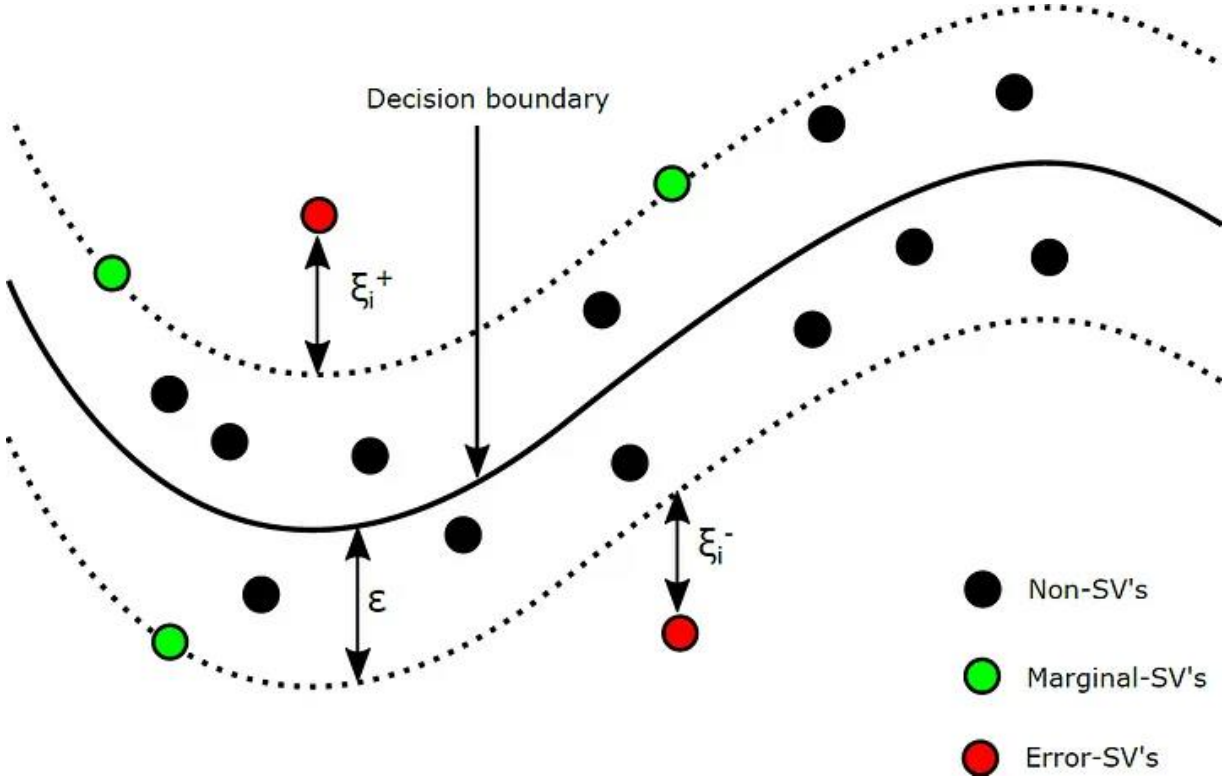
$$\frac{1}{N}A^TA = \frac{1}{N}\begin{pmatrix} x^1 & \cdots & x^N \\ 1 & \cdots & 1 \end{pmatrix}\begin{pmatrix} (x^1)^T & \frac{1}{\vdots} \\ \vdots & \vdots \\ (x^N)^T & 1 \end{pmatrix} = \frac{1}{N}\begin{pmatrix} \sum_{i=1}^N x^i(x^i)^T & \sum_{i=1}^N x^i \\ \sum_{i=1}^N (x^i)^T & N \end{pmatrix}$$

$$\frac{1}{N}A^Ty = \frac{1}{N}\begin{pmatrix} x^1 & \cdots & x^N \\ 1 & \cdots & 1 \end{pmatrix} y = \frac{1}{N}\begin{pmatrix} \sum_{i=1}^N y^i x_1^i \\ \vdots \\ \sum_{i=1}^N y^i x_n^i \\ \sum_{i=1}^N y^i \end{pmatrix}$$

1. The linear relationship $\begin{pmatrix} m \\ b \end{pmatrix}$ of the input vector x and output variable y is determined by the average behavior of x^i, $x^i(x^i)^T$, y^i and $y^i x^i$.
2. The dimensionality of the underlying problem depends solely on the number of features/attributes of the input and output variables $(n+1)$.
   It is independent of the sample size (N).

Data regression - Support vector regression

- https://medium.com/analytics-vidhya/support-vector-regression-svr-model-a-regression-based-machine-learning-approach-f4641670c5bb



Artificial neural network

- https://www.scalablepath.com/machine-learning/chatgpt-architecture-explained