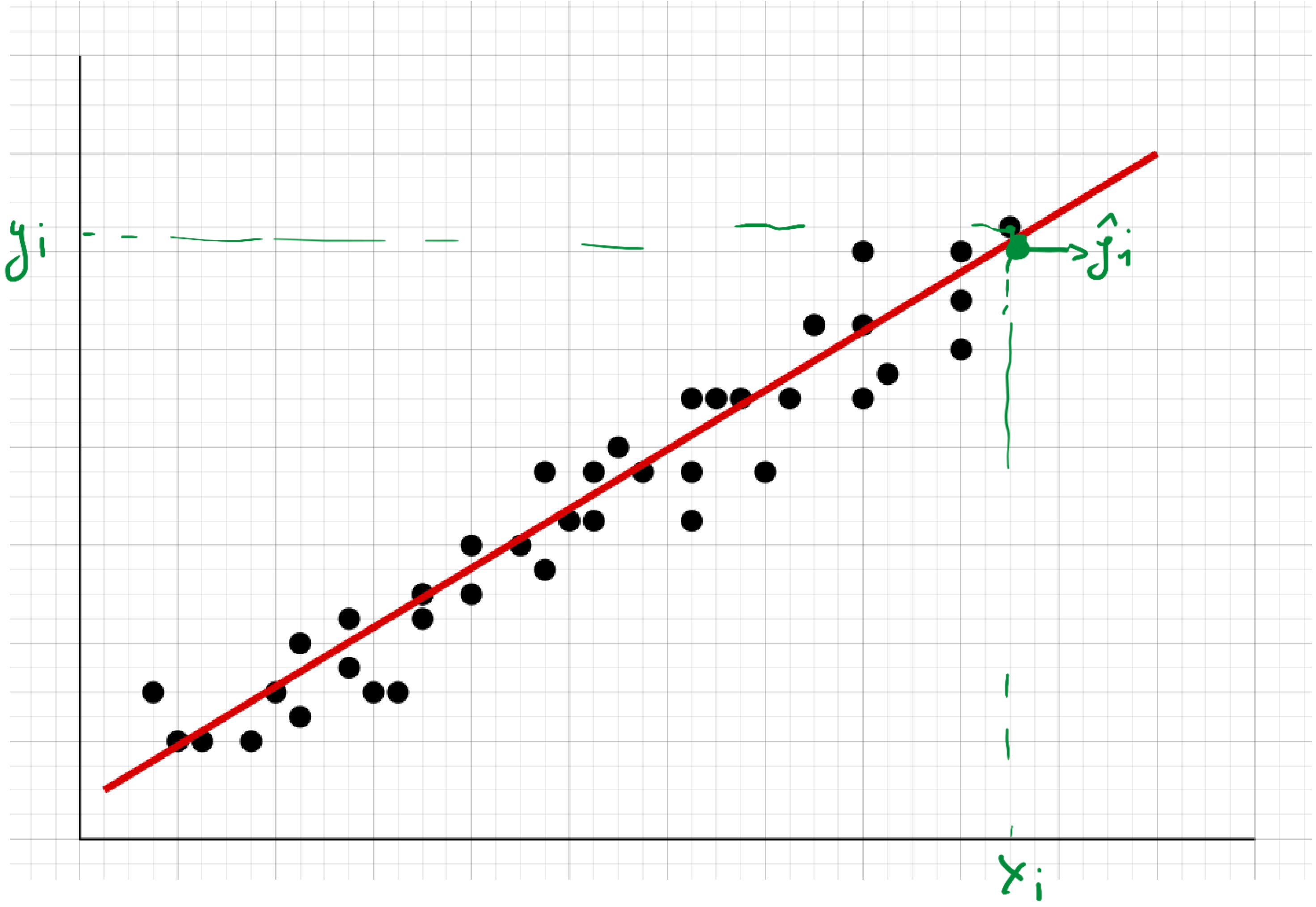
BİL 475 Örüntü Tanıma

Hafta-7:

Gradyan Azalma Algoritması

Matematiksel Tanım (**Matris Formu**)



$$\hat{y}_i = a \cdot x_i + b$$

$$\left.\begin{aligned}
\hat{y}_1 &= a \cdot x_1 + b \\
\hat{y}_2 &= a \cdot x_2 + b \\
\hat{y}_3 &= a \cdot x_3 + b \\
&\vdots \\
\hat{y}_N &= a \cdot x_N + b
\end{aligned}\right\}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$w = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\bar{y} = \bar{\bar{X}} \cdot \bar{w}$$

$$\bar{\bar{X}} = [\bar{x}, \bar{1}]$$

$$\boxed{w = (X^T . X)^{-1} . X^T . y} \qquad X \in R^{N \times 2}$$

Doğrusal Regresyon: Matematiksel Tanım

Least Mean Square (LMS) Error

$$\hat{\bar{y}}_p = \bar{\bar{X}} . \bar{w} \;, \quad \bar{y}_t \Rightarrow L_{avg} = \frac{1}{2N} \| \underbrace{\bar{y}_t - \bar{y}_p}_{\bar{e}} \|_2^2 \Rightarrow (y_1^t - y_1^p)^2 + (y_2^t - y_2^p)^2 + \dots + (y_N^t - y_N^p)^2$$

$$\frac{\partial L_{avg}}{\partial w} = \left(\frac{1}{2N} \| \bar{y}_t - \bar{\bar{X}} . \bar{w} \|_2^2 \right)$$

$$\frac{\partial L_{avg}}{\partial w} = -X^T . (y - X . w) = 0$$

$$-X^T . y + X^T . X . w = 0$$

$$\underbrace{X^T . X}_{C} . w = X^T . y$$

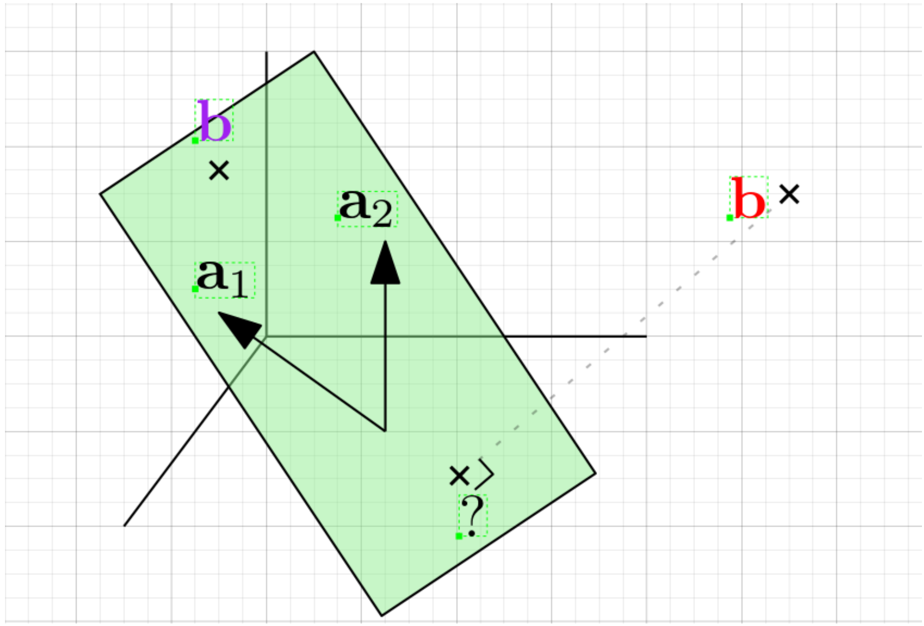
$$\underbrace{inv(C) . C}_{I} . w = inv(C) . X^T . y$$

$$w = (X^T . X)^{-1} . X^T . y$$

Doğrusal Regresyon: Matematiksel Tanım

Least Mean Square (LMS) Error

$$\begin{bmatrix} x_{11} & 1 \\ x_{21} & 1 \\ \cdot & \cdot \\ \cdot & \cdot \\ x_{N1} & 1 \end{bmatrix} \boldsymbol{w} = \boldsymbol{y}$$



Doğrusal Regresyon: Çözüm Algoritması (N Boyut)

$N \to \quad \bar{I} = \text{ones}(N, 1)$

$\bar{\bar{X}} = [\bar{x}, \bar{I}]$

$C = \bar{\bar{X}}^T \cdot \bar{\bar{X}}$

$w = \bar{\bar{C}}^{-1} \cdot \bar{\bar{X}}^T \cdot \bar{y}$

$\underline{C} + \alpha \cdot \underline{I}$

eye

$\boxed{\alpha = 10^{-4}}$

$$\begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

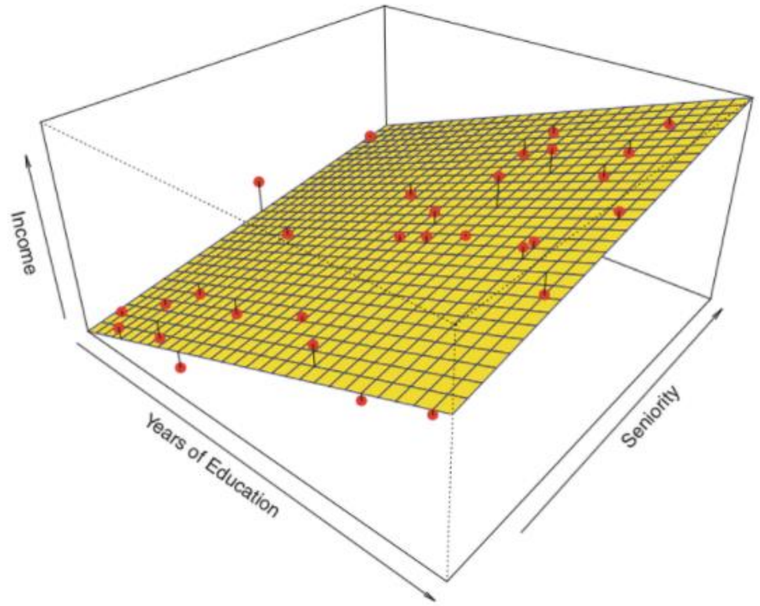
100

5000

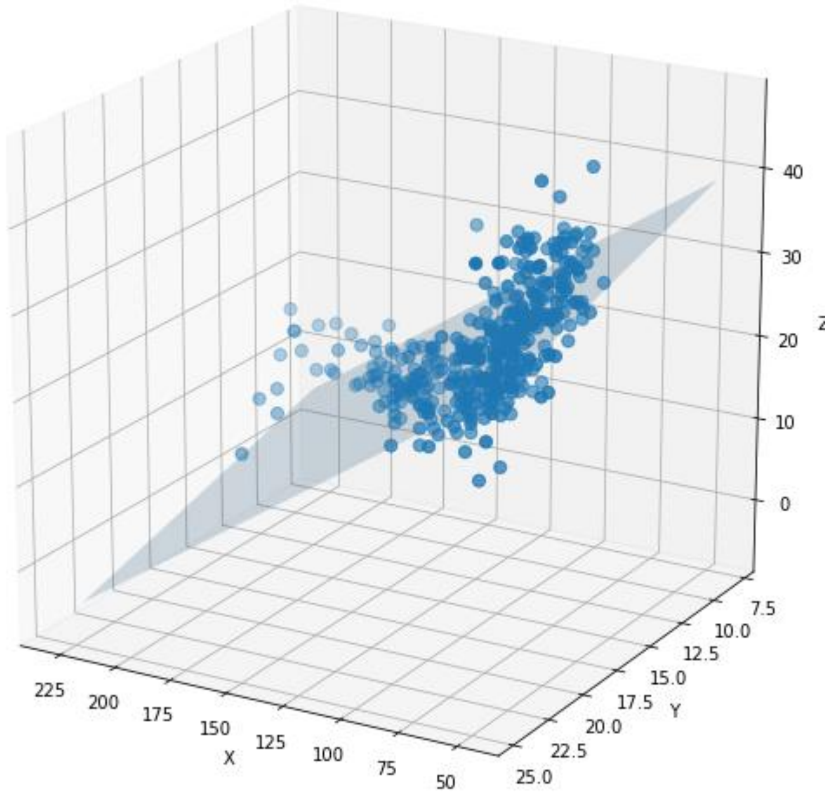
$X^T \cdot X + \alpha \cdot I \quad \alpha = 10^{-}$

$\boxed{100 \times 100}$

Doğrusal Regresyon: Çözüm Algoritması



Source: James et al. *Introduction to Statistical Learning* (Springer 2013)



https://i.stack.imgur.com/O5036.png

Doğrusal Regresyon: Eğitim, Test ve Skor

Skor Metrikleri:

$$MSE = \mathrm{mean}\left(|\bar{y}_t - \bar{y}_p|^2\right)$$

$$MAE = \mathrm{mean}\left(|\bar{y}_t - \bar{y}_p|\right)$$

$$RME =$$

$$MAPE = \mathrm{mean}\left(\left|\frac{\bar{y}_t - \bar{y}_p}{\bar{y}_t}\right|\right) \cdot 100\ \%$$

$$SMAPE = \mathrm{mean}\left\{\left|\frac{\bar{y}_t - \bar{y}_p}{\left(\frac{\bar{y}_t + \bar{y}_p}{2}\right)}\right|\right\}$$

760 820 640

740 850 630

e = 20 30 10

MAE

$\frac{60}{3} = 20$

$\frac{20}{760}$ $\frac{30}{820}$ $\frac{10}{640}$ $\sim \%6$

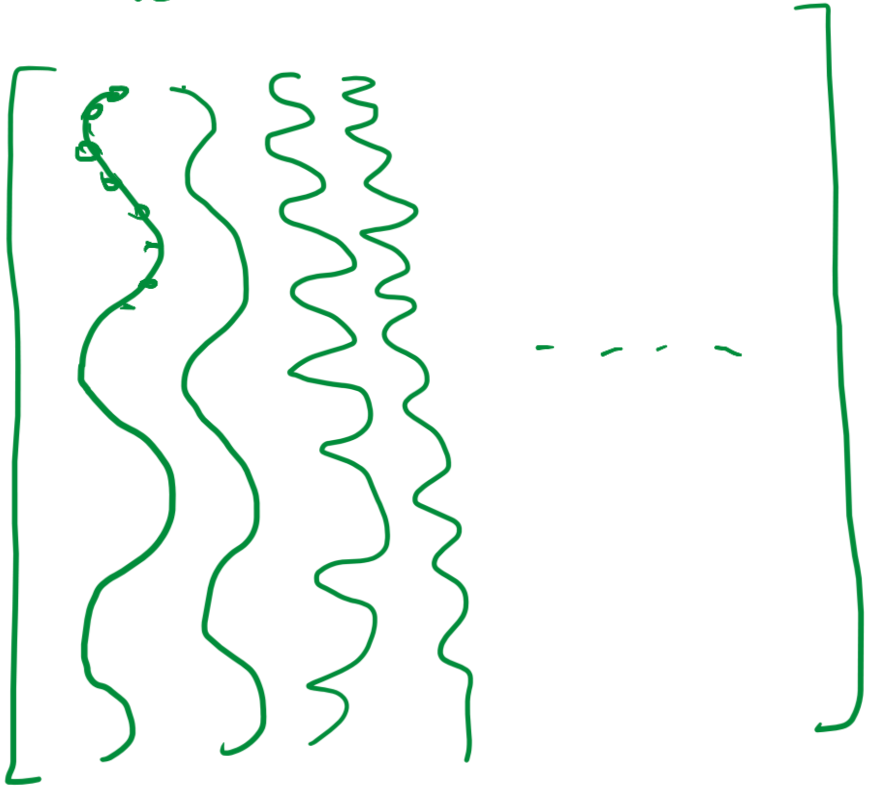
0 2 5 1

0,1 2,01 5,02 1,03

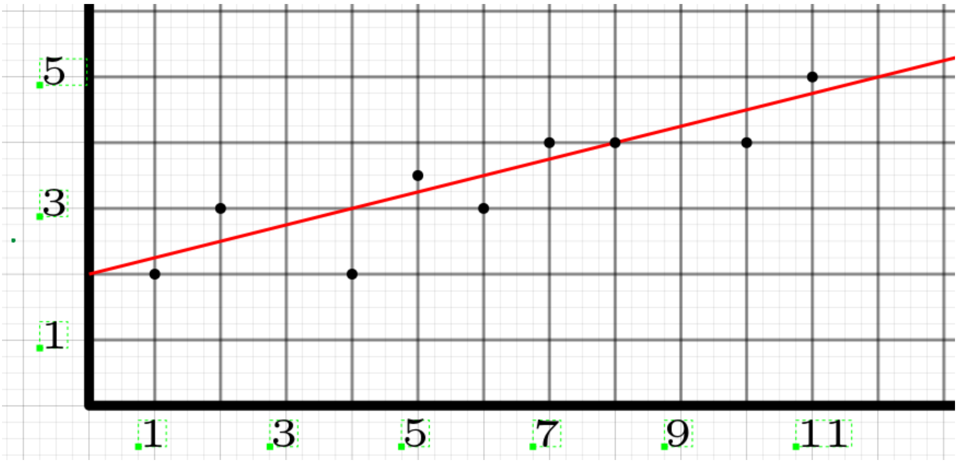
Doğrusal Regresyon: ~~Taylor~~ ve Fourier Serileri

$$\tilde{X}_p = \sum_{100} a_k \cos(k\Omega_0 t) + b_k \sin(k\Omega_0 t) \Rightarrow \text{Doğrusal Regresyon}$$

5000



$$\begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \\ a_3 \\ b_3 \\ \vdots \\ a_{500} \\ b_{500} \end{bmatrix} = \tilde{X}_p$$



x	y
1	2
2	3
4	2
5	3.5
6	3
7	4
8	6
10	4
11	5

$$\bar{x} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 11 \end{bmatrix} \qquad \bar{\bar{X}} = \begin{bmatrix} \bar{x}, \bar{1} \end{bmatrix}$$

$$C = \bar{\bar{X}}^T \cdot \bar{\bar{X}} \qquad \boxed{w_p = C^{-1} \cdot \bar{\bar{X}}^T \cdot y}$$

$$\bar{\bar{X}} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 30 & 12 \\ 12 & 4 \end{bmatrix}$$

Doğrusal Regresyon: Aykırı Örnekler ve RANSAC Algoritması

RANSAC: Random Sample Consensus

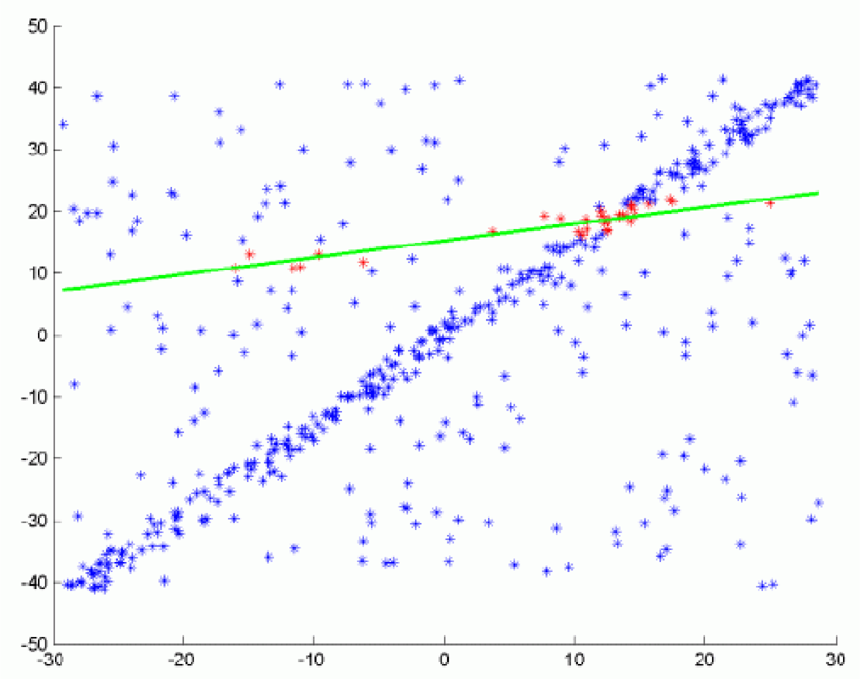
GİRDİLER
    $D = \{x_i, y_i : i = 1, \ldots, N\}$: veri kümesi
    K: Rastgele seçilecek örnek sayısı
    τ: Model ile veri arasındaki en büyük uzaklık
    T: İterasyon sayısı
ÇIKTILAR
    M: Uyumlanan model

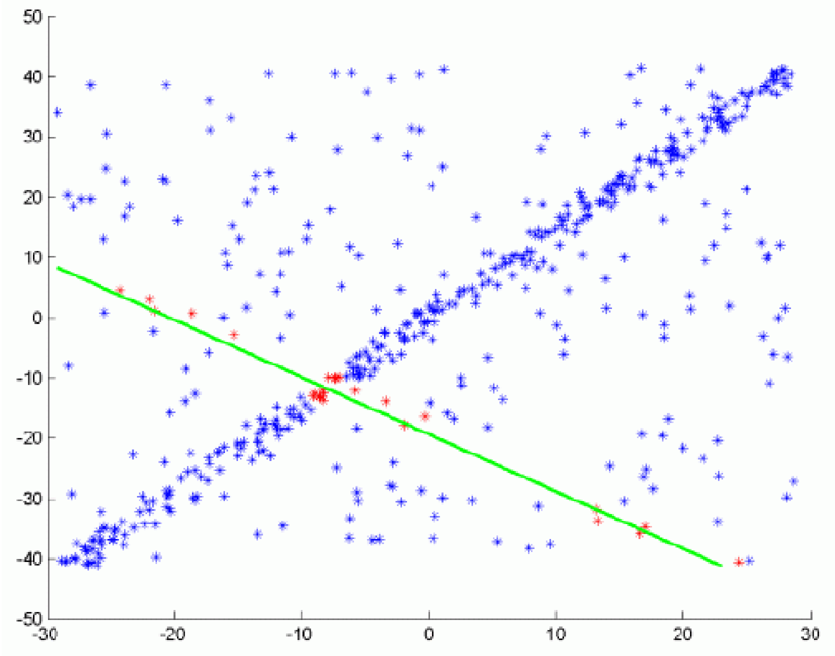
EnYuksekUyum = 0
t = 0
while t++ < T
    N veri-etiket çiftinden K tanesini rastgele örnekle, $S = \{x_k, y_k : k = 1, \ldots, K\}$
    K tane veri üzerinden model uyumlama gerçekleştir, fit(S, Model)
    D kümesi ile model uyumunu hesapla, $u = \#\{x_i, y_i : \|\mathrm{Model}(x_i) - y_i\| \leq \tau, i = 1, \ldots, N\}$
    if u > EnYuksekUyum
        M = Model
        EnYuksekUyum = u
return M

https://imlab.io/2020/08/23/ransac-algorithm/

Doğrusal Regresyon: Aykırı Örnekler ve RANSAC Algoritması

RANSAC: Random Sample Consensus

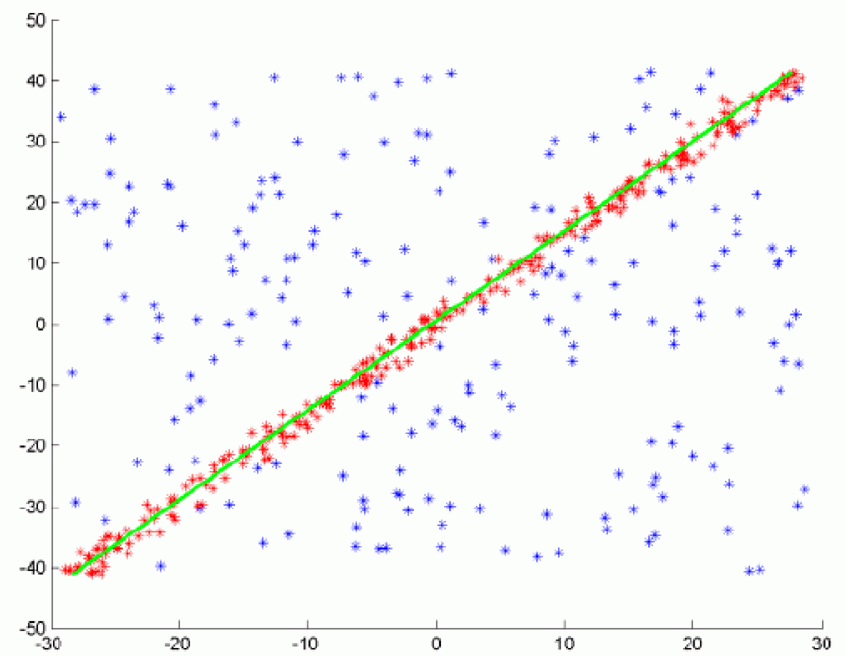




https://www.analyticsvidhya.com/blog/2021/02/new-approach-for-regression-analysis-ransac-and-mlesac/

Doğrusal Regresyon: Aykırı Örnekler ve RANSAC Algoritması

RANSAC: Random Sample Consensus



https://www.analyticsvidhya.com/blog/2021/02/new-approach-for-regression-analysis-ransac-and-mlesac/

sklearn.linear_model: Linear Models

The sklearn.linear_model module implements a variety of linear models.

User guide: See the Linear Models section for further details.

The following subsections are only rough guidelines: the same estimator can fall into multiple categories, depending on its parameters.

Linear classifiers

linear_model.LogisticRegression([penalty, ...])	Logistic Regression (aka logit, MaxEnt) classifier.
linear_model.LogisticRegressionCV(*[, Cs, ...])	Logistic Regression CV (aka logit, MaxEnt) classifier.
linear_model.PassiveAggressiveClassifier(*)	Passive Aggressive Classifier.
linear_model.Perceptron(*[, penalty, alpha, ...])	Linear perceptron classifier.
linear_model.RidgeClassifier([alpha, ...])	Classifier using Ridge regression.
linear_model.RidgeClassifierCV([alphas, ...])	Ridge classifier with built-in cross-validation.
linear_model.SGDClassifier([loss, penalty, ...])	Linear classifiers (SVM, logistic regression, etc.) with SGD training.
linear_model.SGDOneClassSVM([nu, ...])	Solves linear One-Class SVM using Stochastic Gradient Descent.

Support vector Regression

Classical linear regressors

linear_model.LinearRegression(*[, ...])	Ordinary least squares Linear Regression.
linear_model.Ridge([alpha, fit_intercept, ...])	Linear least squares with l2 regularization.
linear_model.RidgeCV([alphas, ...])	Ridge regression with built-in cross-validation.
linear_model.SGDRegressor([loss, penalty, ...])	Linear model fitted by minimizing a regularized empirical loss with SGD.

MLP

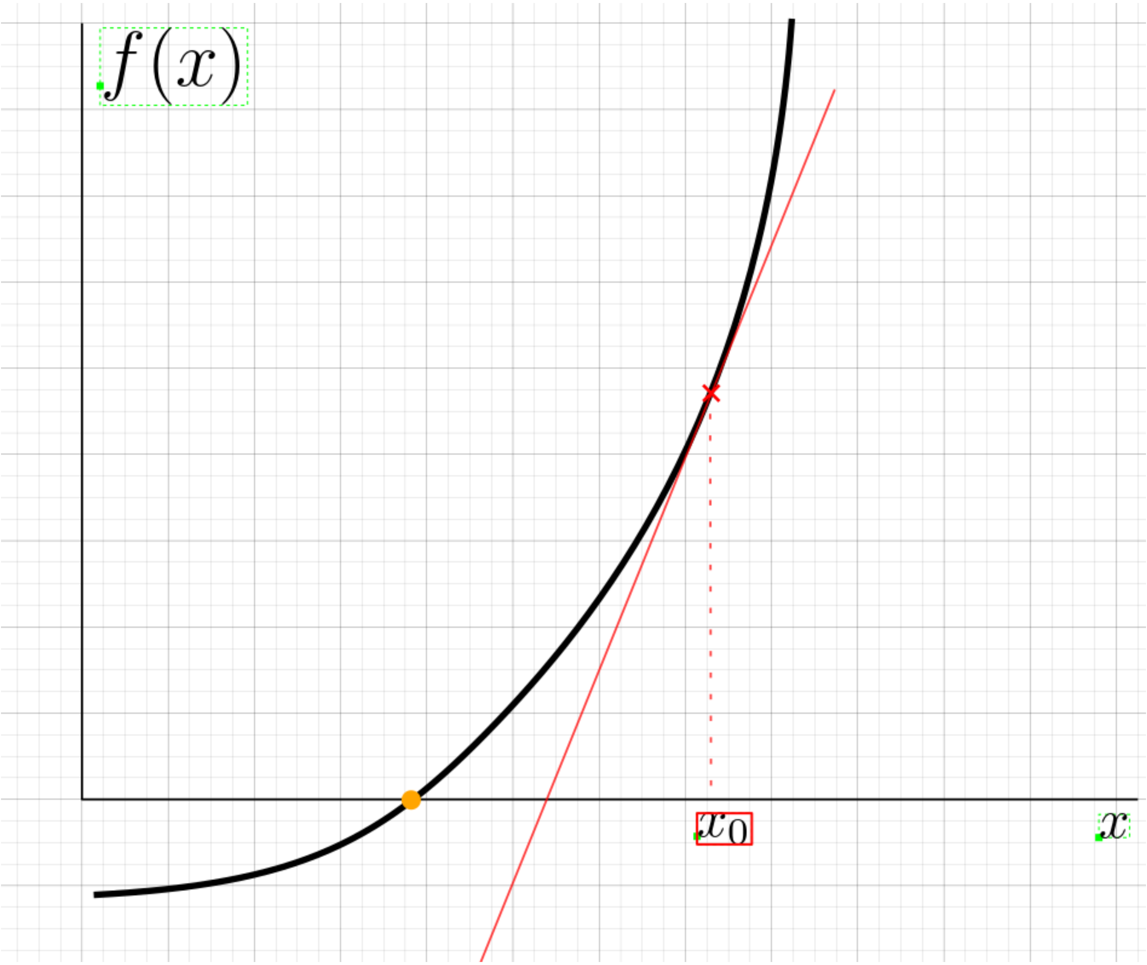
Regressors with variable selection

The following estimators have built-in variable selection fitting procedures, but any estimator using a L1 or elastic-net penalty also performs variable selection: typically SGDRegressor or SGDClassifier with an appropriate penalty.

linear_model.ElasticNet([alpha, l1_ratio, ...])	Linear regression with combined L1 and L2 priors as regularizer.
linear_model.ElasticNetCV(*[, l1_ratio, ...])	Elastic Net model with iterative fitting along a regularization path.
linear_model.Lars(*[, fit_intercept, ...])	Least Angle Regression model a.k.a.
linear_model.LarsCV(*[, fit_intercept, ...])	Cross-validated Least Angle Regression model.
linear_model.Lasso([alpha, fit_intercept, ...])	Linear Model trained with L1 prior as regularizer (aka the Lasso).

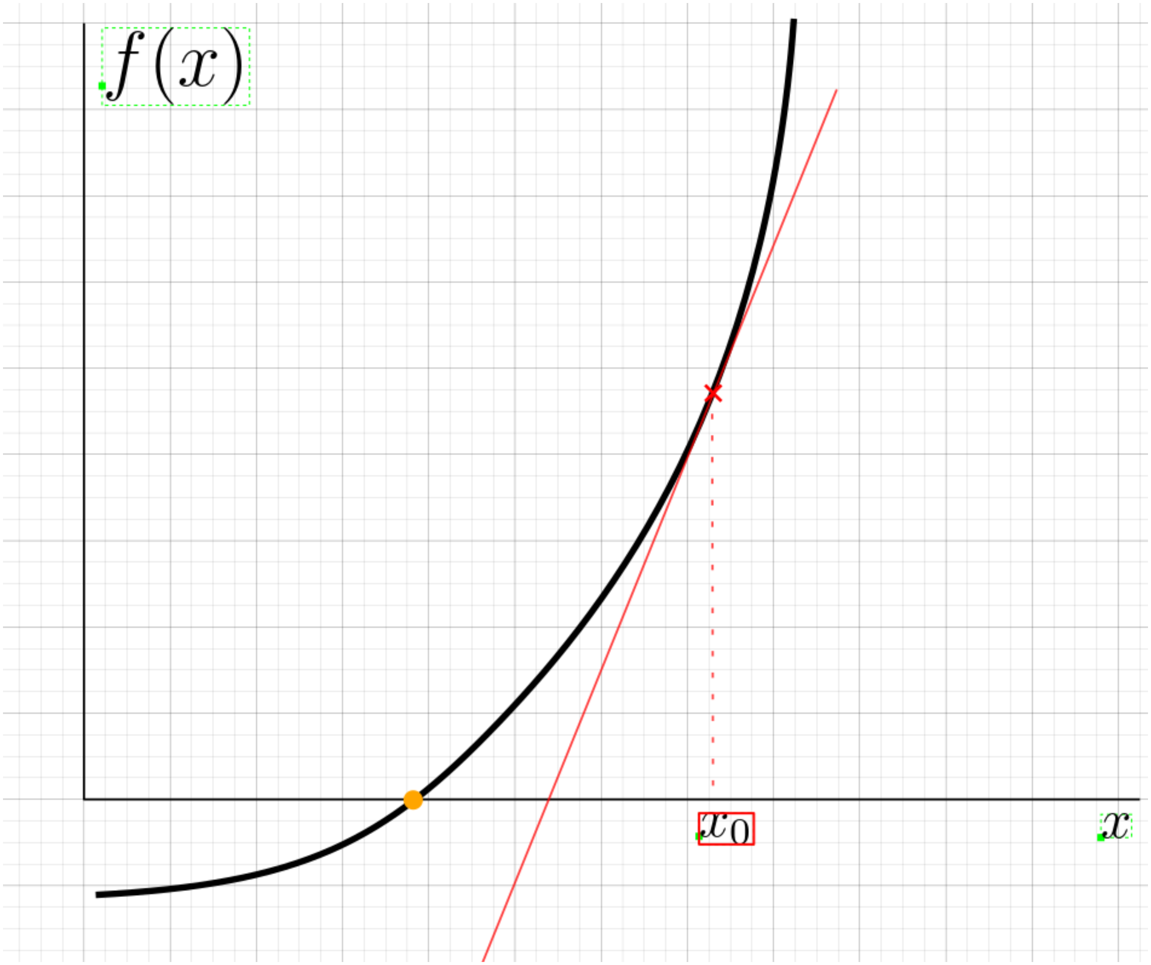
Gradyan Azalma Algoritması

Gradyan İnişi : Vekil Fonksiyon Kavramı



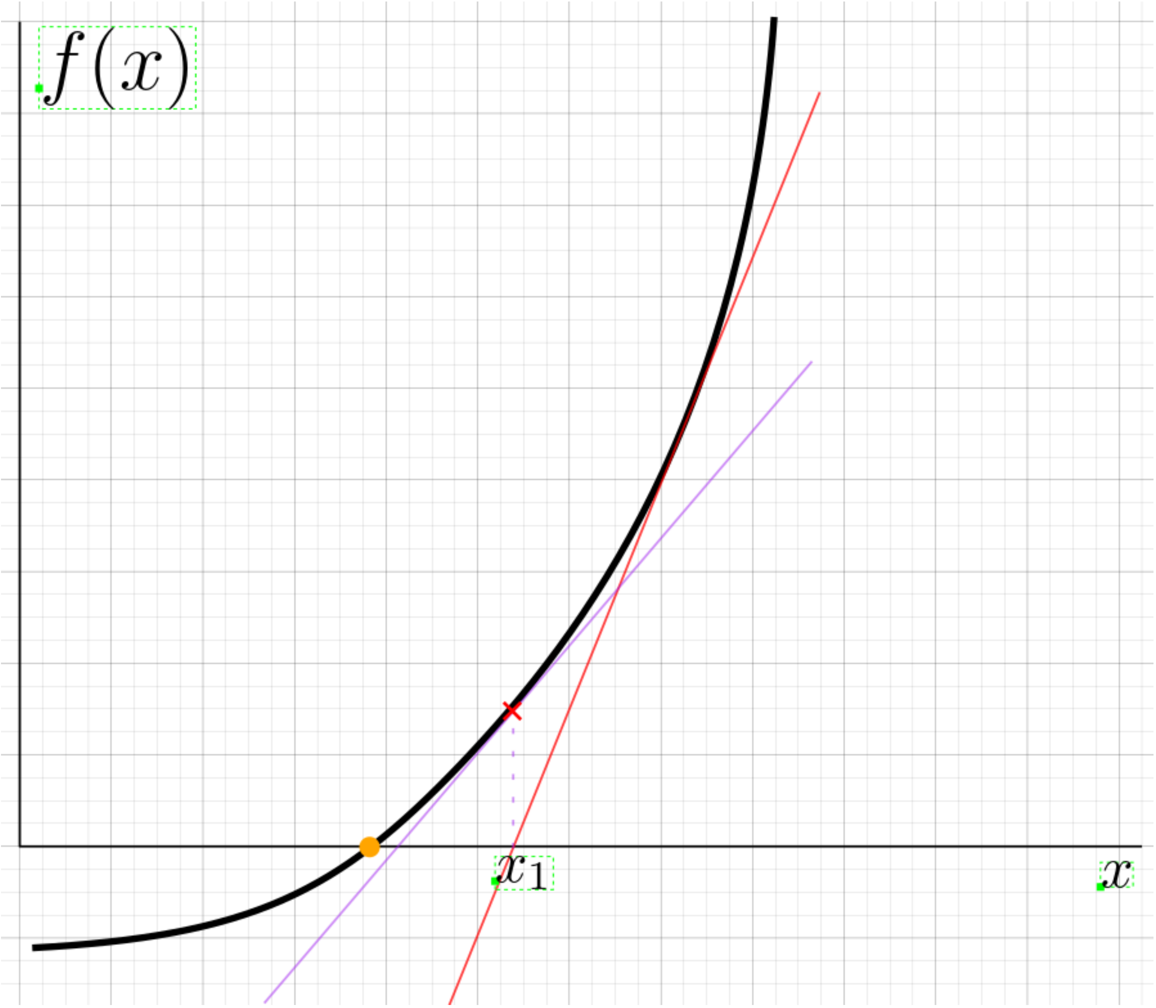
Kök Bulma Problemi !

Gradyan İnişi : Vekil Fonksiyon Kavramı



- Rasgele bir x0 noktası belirle
- x0'a göre türev al (m0)
- Eğimi ve bir noktası bilinen doğru denklemini bul
- Doğrunun y=0 noktasını tespit et
- 2. satıra geri dön

Gradyan İnişi : Vekil Fonksiyon Kavramı



- Rasgele bir x0 noktası belirle
- x0’a göre türev al (m0)
- Eğimi ve bir noktası bilinen doğru denklemini bul
- Doğrunun y=0 noktasını tespit et
- 2. satıra geri dön

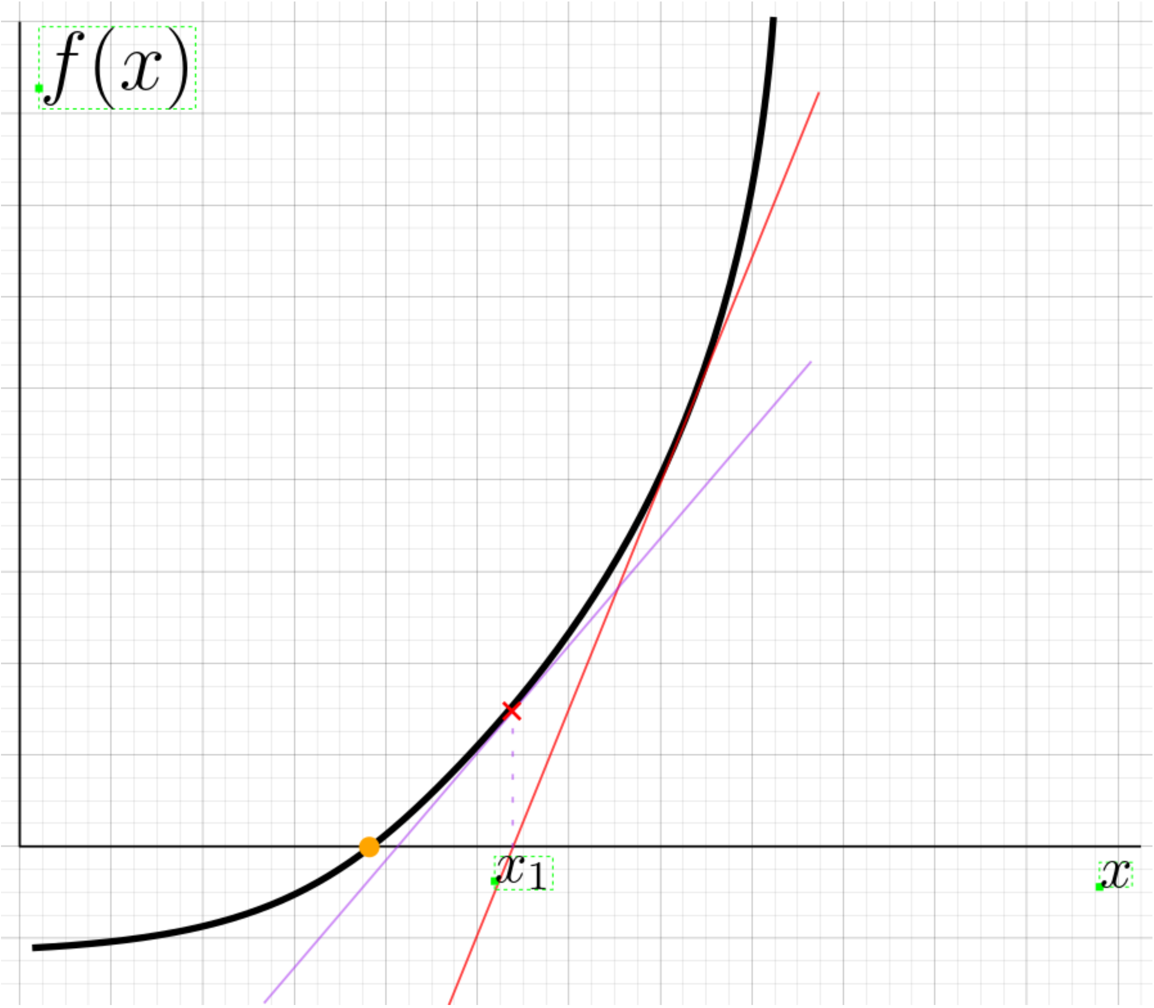
$$m_0 = f'(x_0)$$

$$y = m_0(x - x_0) + y_0 \qquad y = m_0(x - x_0) + f(x_0)$$

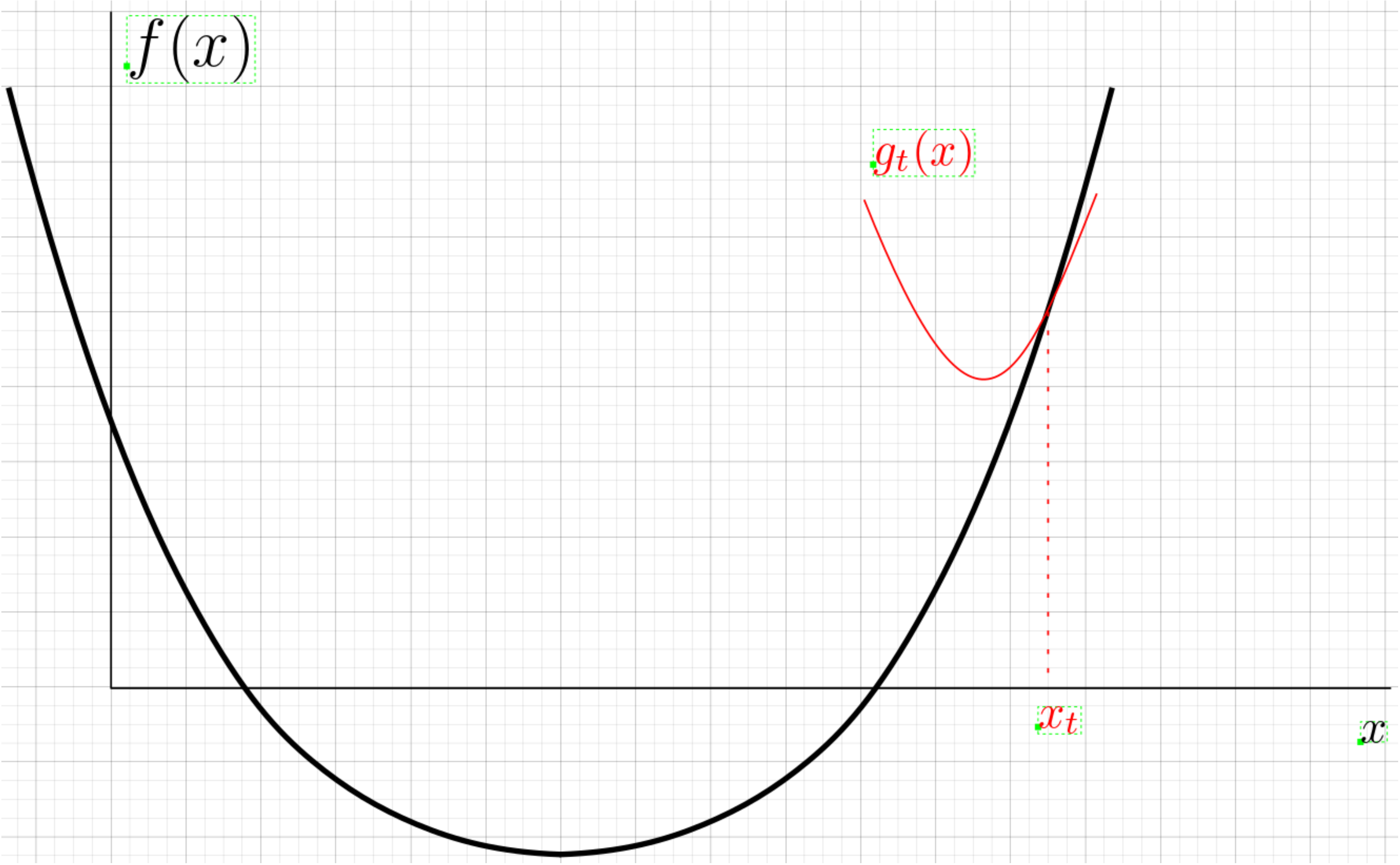
$$0 = m_0(x - x_0) + f(x_0)$$

$$x = x_0 - \frac{f(x_0)}{m_0} \qquad x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

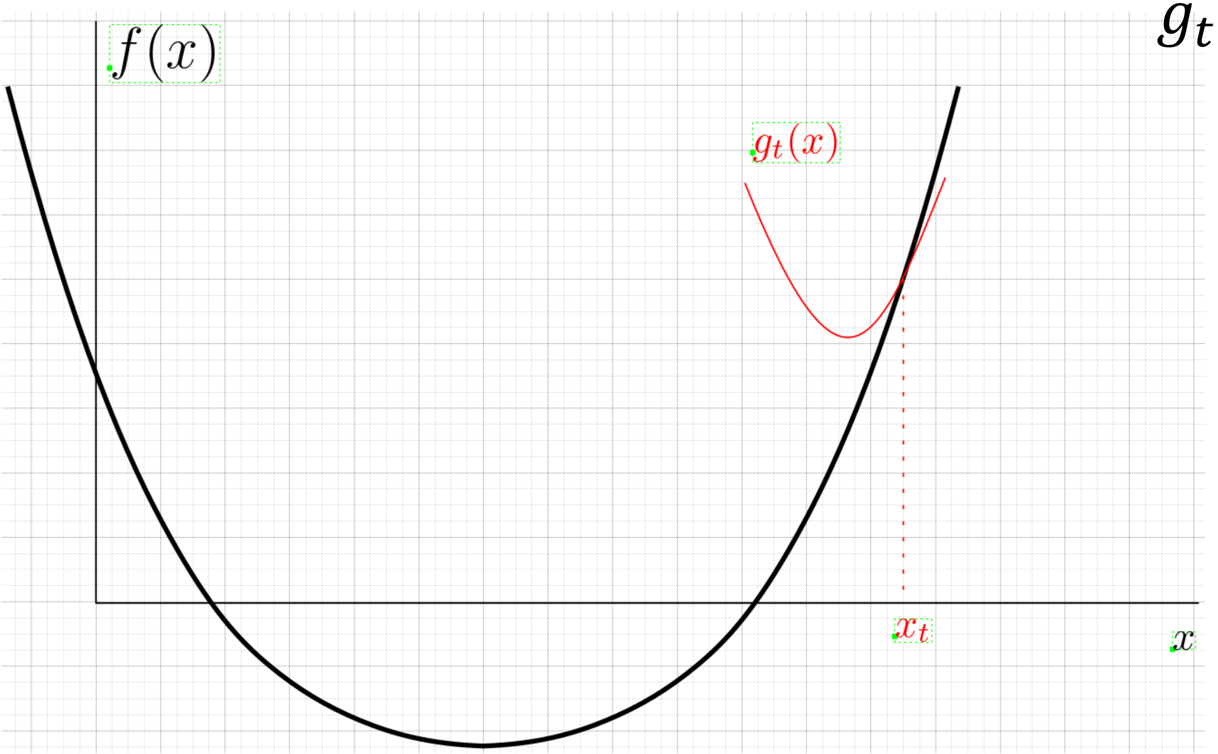
Gradyan İnişi : Vekil Fonksiyon Kavramı



$$g_t(x) = f(x_t) + f'(x_t)(x - x_t)$$

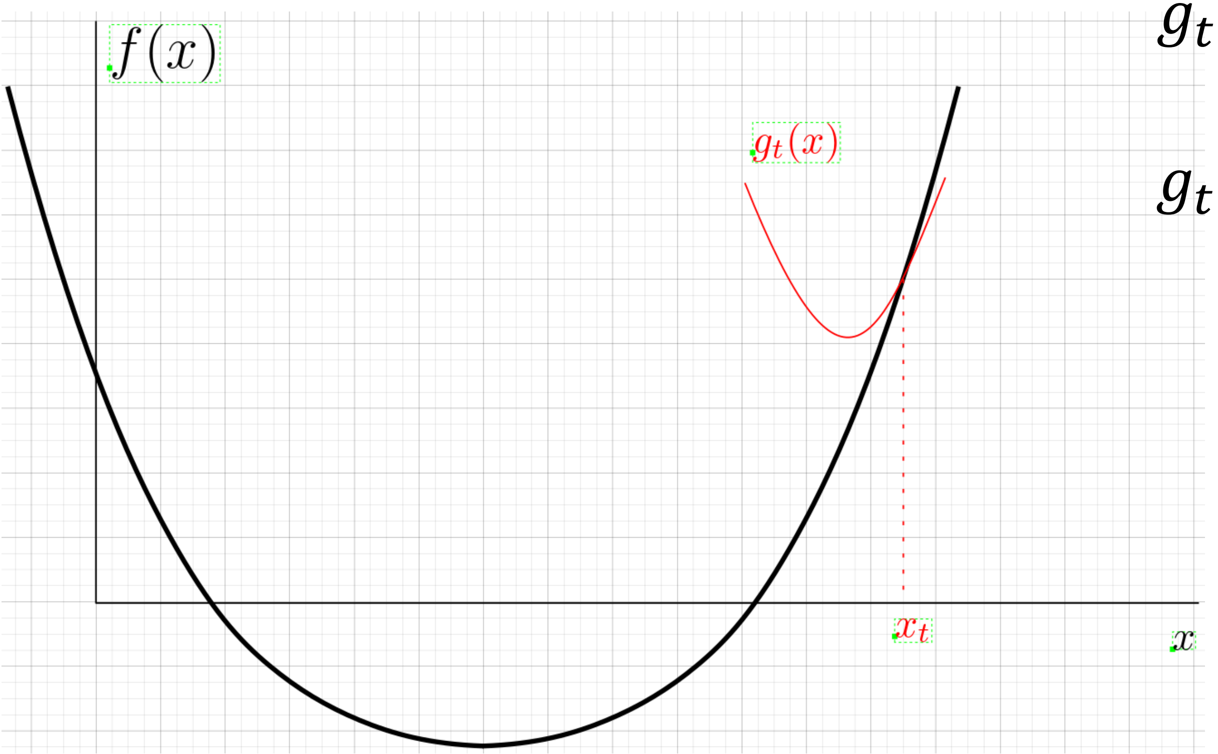

$f(x)$
$g_t(x)$
x_t
x

Gradyan İnişi : Vekil Fonksiyon Kavramı



$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + f''(x_t)(x - x_t)^2$$

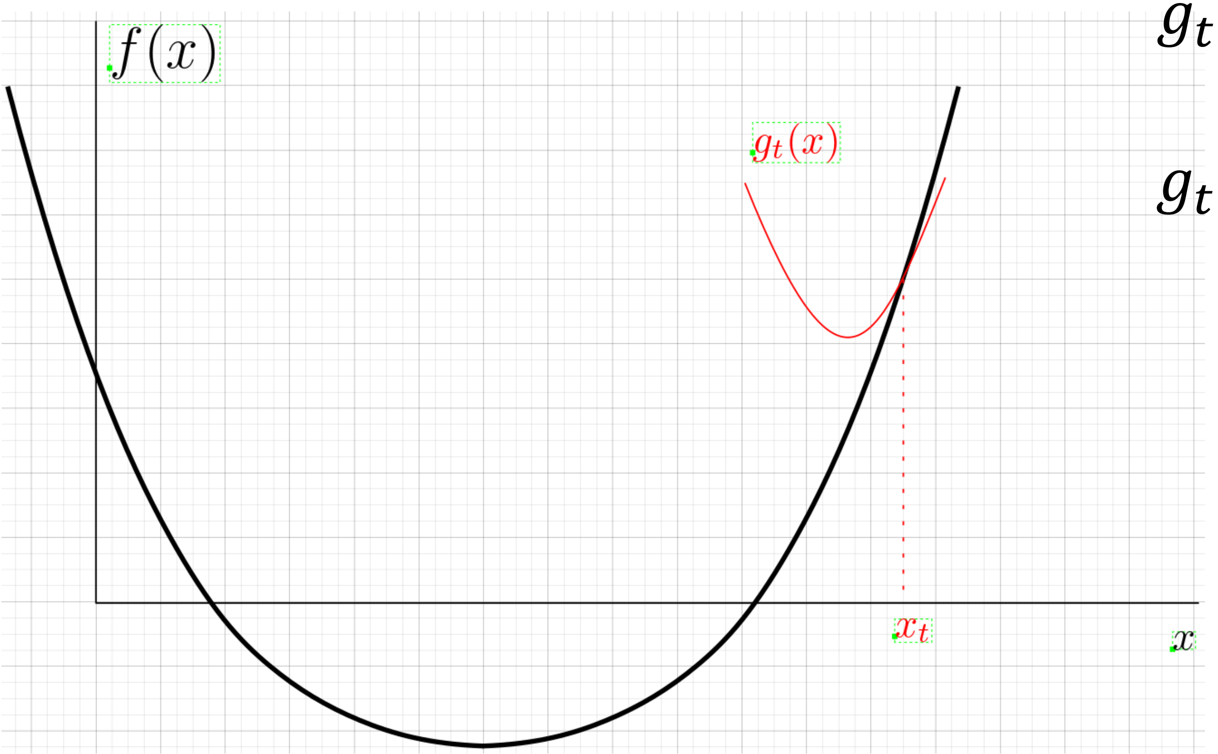
Gradyan İnişi : Vekil Fonksiyon Kavramı



$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + f''(x_t)(x - x_t)^2$$

$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + \alpha(x - x_t)^2$$

Gradyan İnişi : Vekil Fonksiyon Kavramı

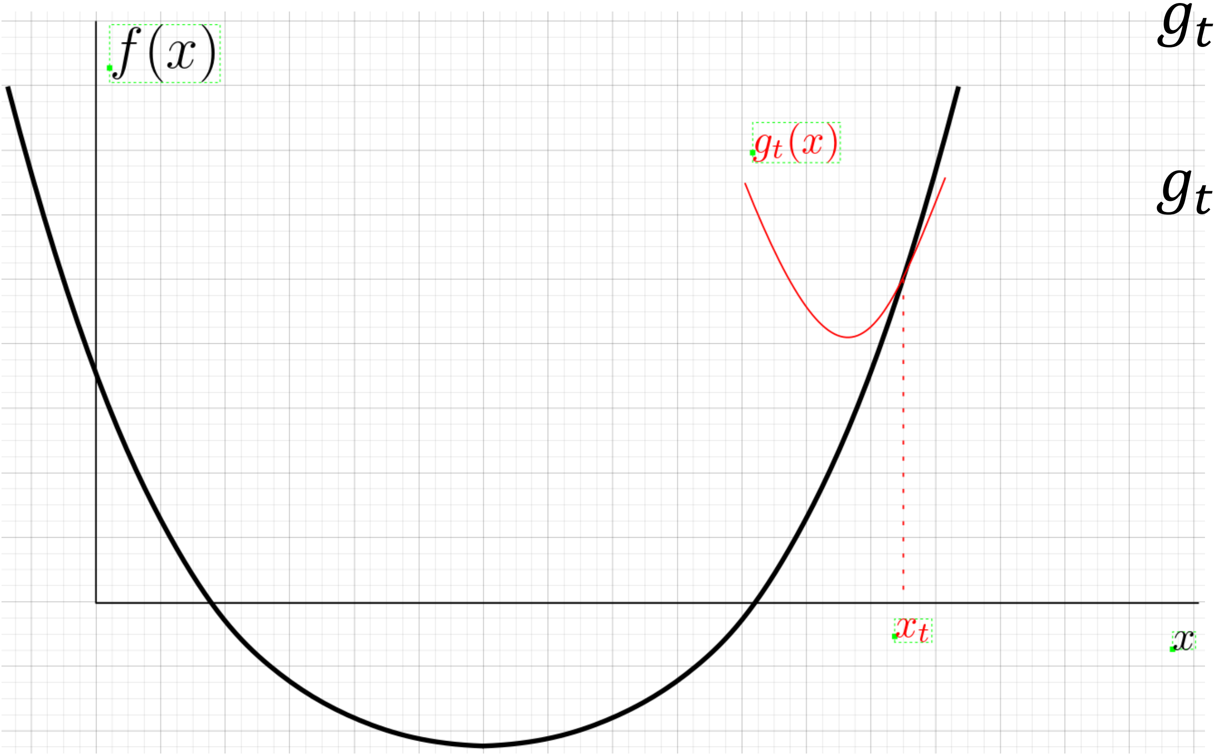


$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + f''(x_t)(x - x_t)^2$$

$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + \alpha(x - x_t)^2$$

$$g_t'(x) = 0$$

Gradyan İnişi : Vekil Fonksiyon Kavramı



$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + f''(x_t)(x - x_t)^2$$

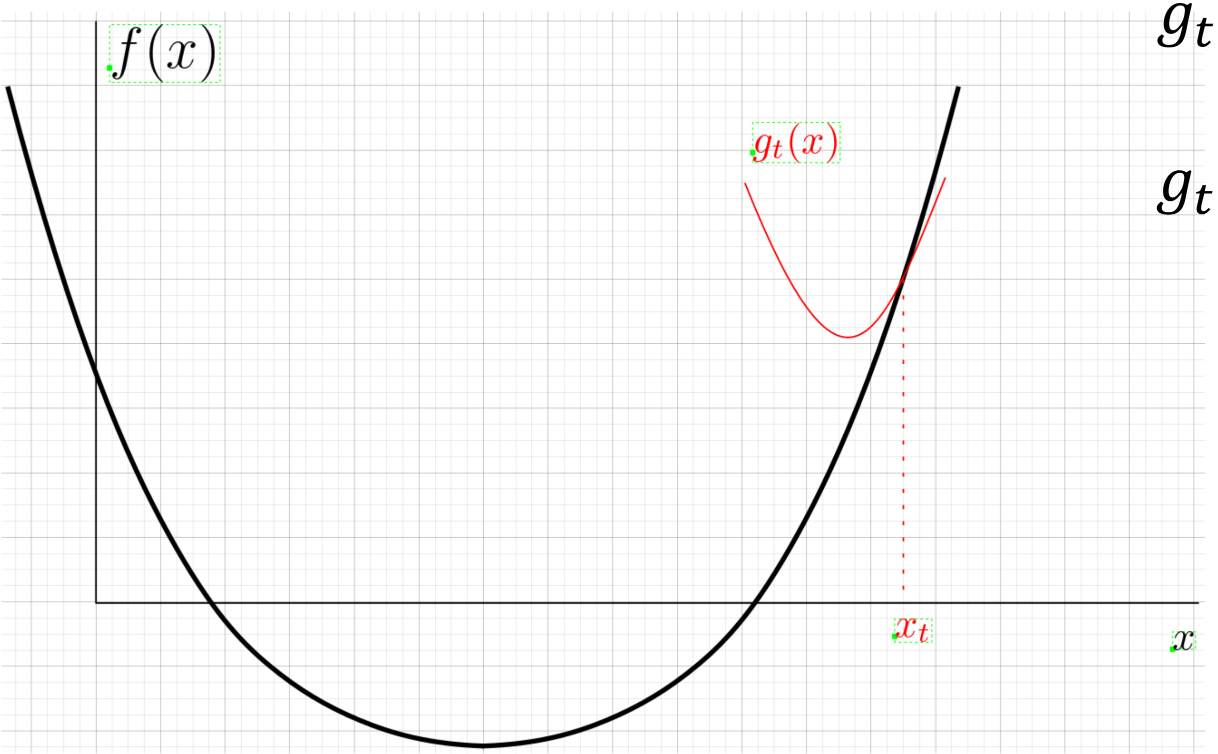
$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + \alpha(x - x_t)^2$$

$$g_t'(x) = 0$$

$$0 = f'(x_t) + 2\alpha(x - x_t)$$

$$x = x_t - \frac{1}{2\alpha} f'(x_t)$$

Gradyan İnişi : Vekil Fonksiyon Kavramı



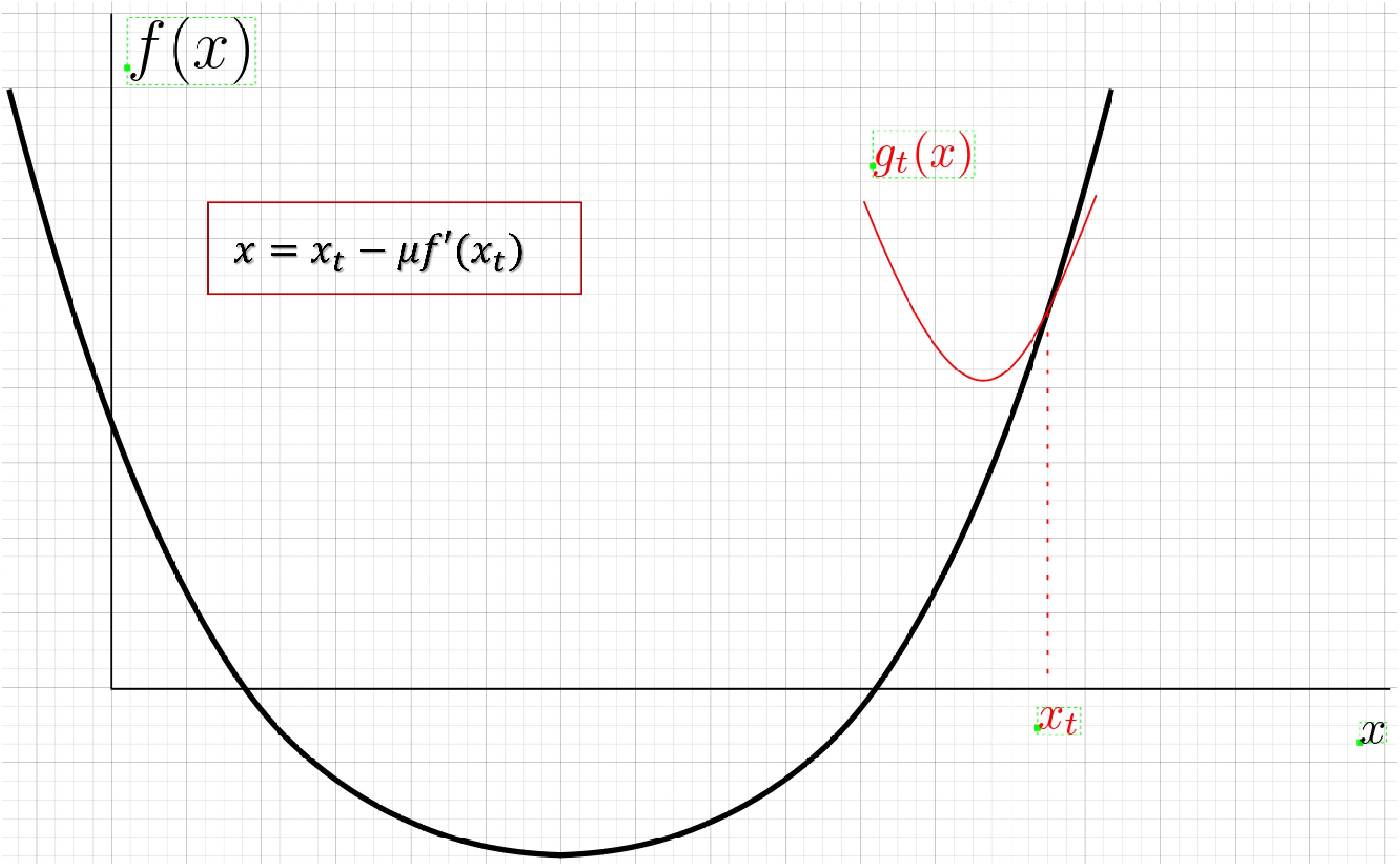
$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + f''(x_t)(x - x_t)^2$$

$$g_t(x) = f(x_t) + f'(x_t)(x - x_t) + \alpha(x - x_t)^2$$

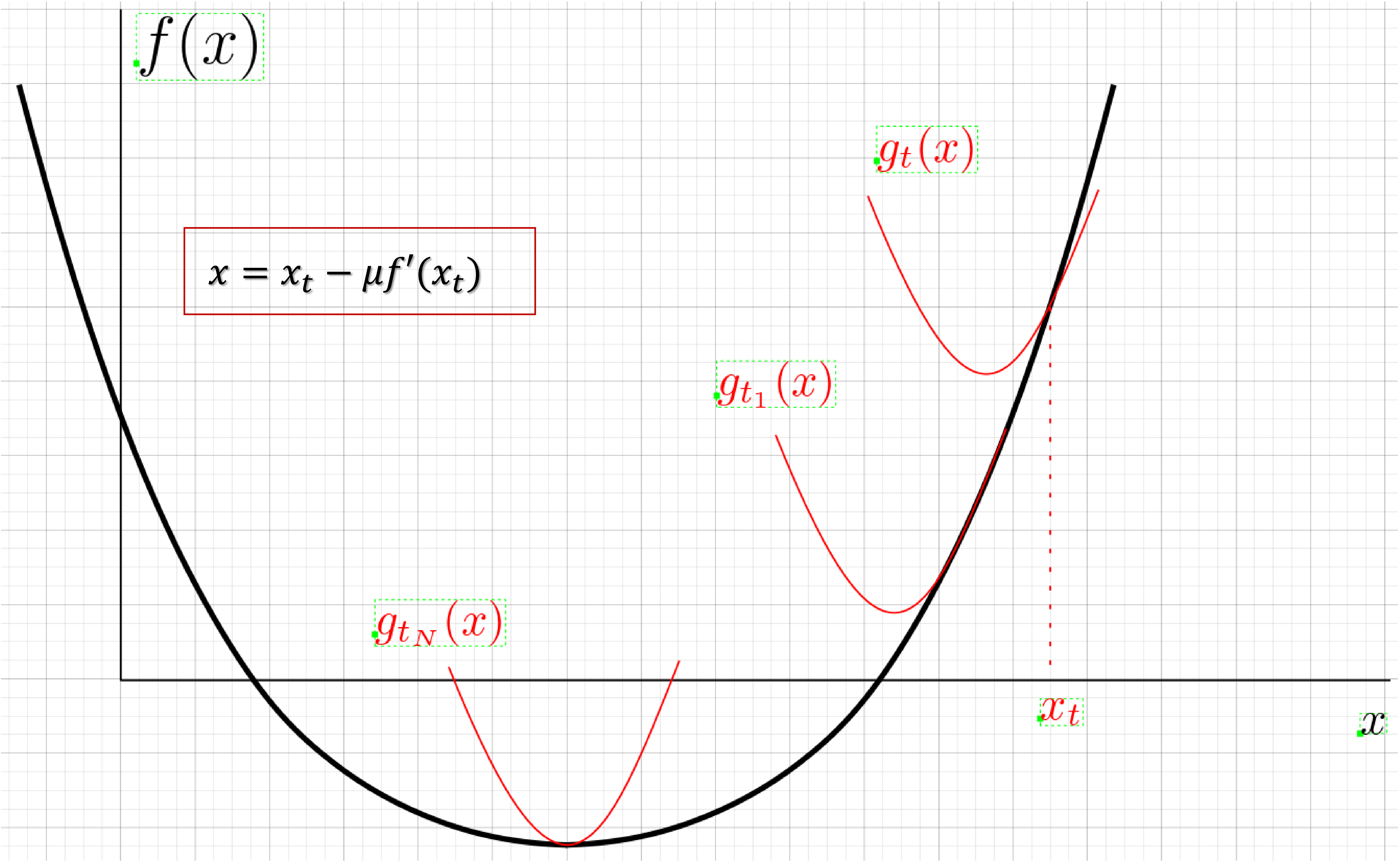
$$g_t'(x) = 0$$

$$0 = f'(x_t) + 2\alpha(x - x_t)$$

$$\boxed{x = x_t - \mu f'(x_t)}$$



$f(x)$
$x = x_t - \mu f'(x_t)$
$g_t(x)$
x_t
x



$f(x)$
$x = x_t - \mu f'(x_t)$
$g_t(x)$
$g_{t_1}(x)$
$g_{t_N}(x)$
x_t
x

Gradyan İnişi – Nümerik Türev

Gradyan İnişi-Öğrenme Oranının Durumları

Gradyan İnişi

S3C (P.10) $f(x) = (x-2)^2$ kayıp fonksiyonunun $x_0 = 10$ noktasından başlayarak 3 adım boyunca gradyan inişini hesaplayınız. $(lr = 0.5)$

x	f(x)	f'(x)	x_{t+1}

Gradyan İnişi-Problemin İç Bükey Olması/Olmaması