CS57300
PURDUE UNIVERSITY

FEBRUARY 19, 2019

DATA MINING

ANNOUNCEMENT

- Assignment 3 is out
  - Implement Logistic Regression and Linear SVM for speed dating event outcome prediction
  - Due: March 8 (Friday), 11:59pm
- In-class midterm exam in two weeks
  - March 5, 4:30-5:45pm, WANG 2599

SMOOTH OPTIMIZATION

SOLVE CONVEX OPTIMIZATION PROBLEM

- Minimize a convex function without any constraints on the variables
    - If $f'(x)=0$ then x is a stationary point of f
    - If $f'(x)=0$ and $f''(x)$ is not negative then x is a local minimum of f (for convex function, this is also a global minimum)
    - If f is a strictly convex function, any stationary point of ***f*** is the unique global minimum of f
- What about minimizing a convex function with constraints?

USE LAGRANGE MULTIPLIERS TO SOLVE CONVEX OPTIMIZATION

- For a standard form of convex optimization problem (f_0 are f_i are convex, h_i is linear):

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad \text{for } i = 1, \ldots, m. \\
& h_i(x) = 0, \quad \text{for } i = 1, \ldots, k.
\end{array}
$$

- The Lagrangian function of it is

$$L(x, \lambda, v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{k} v_i h_i(x)$$

  - $\lambda_i \geq 0$ is the **Lagrange multiplier** for the *i*-th inequality constraint, v_i is the **Lagrange multiplier** for the *i*-th equality constraint
  - Solve the constrained optimization problem by finding the stationary point of the Lagrangian function

LOGISTIC REGRESSION LEARNING

- Logistic regression: $P(y=1|\mathbf{x}) = \dfrac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+w_0)}}$
  - Maximize (log) likelihood: $\mathsf{w} = (\mathsf{w}, w_0), \mathsf{x_i} = (\mathsf{x_i}, 1)$

$$
\begin{aligned}
logL(\mathsf{w}\,|\,D) &= \sum_{i=1}^{N} logp(y_i\,|\,\mathsf{w}) \\
&= \sum_{i=1}^{N} log[(\frac{1}{1+e^{-\mathsf{w}^\mathsf{T}\mathsf{x_i}}})^{y_i}(\frac{e^{-\mathsf{w}^\mathsf{T}\mathsf{x_i}}}{1+e^{-\mathsf{w}^\mathsf{T}\mathsf{x_i}}})^{1-y_i}] \\
&= \sum_{i=1}^{N} (y_i\mathsf{w}^\mathsf{T}\mathsf{x_i} - log(1+e^{\mathsf{w}^\mathsf{T}\mathsf{x_i}}))
\end{aligned}
$$

  - Minimize: $\sum_{i=1}^{N} (-y_i\mathsf{w}^\mathsf{T}\mathsf{x_i} + log(1+e^{\mathsf{w}^\mathsf{T}\mathsf{x_i}}))$

LOGISTIC REGRESSION LEARNING

$$minimize \sum_{i=1}^{N} (-y_i \mathbf{w}^T \mathbf{x_i} + log(1 + e^{\mathbf{w}^T \mathbf{x_i}}))$$

$$\frac{dlogL(\mathbf{w} \,|\, D)}{dw_j} = \sum_{i=1}^{N} (-y_i x_{ij} + \frac{1}{1 + e^{\mathbf{w}^T \mathbf{x_i}}} e^{\mathbf{w}^T \mathbf{x_i}} x_{ij})$$

$$= \sum_{i=1}^{N} (-y_i + \frac{1}{1 + e^{\mathbf{w}^T \mathbf{x_i}}} e^{\mathbf{w}^T \mathbf{x_i}}) x_{ij}$$

$$= \sum_{i=1}^{N} (-y_i + P(y_i = 1 \,|\, \mathbf{w})) x_{ij}$$

Convex!

But no closed form solution!

GRADIENT DESCENT

- For some convex functions, we may be able to take the derivative, but it may be difficult to directly solve for parameter values
- Solution:
  - Start at some value of the parameters
  - Take derivative and use it to move the parameters in the direction of the negative gradient
  - Repeat until stopping criteria is met (e.g., gradient close to 0)

<u>Gradient Descent Rule:</u>

$$\underline{\mathbf{w}}_{\text{new}} = \underline{\mathbf{w}}_{\text{old}} - \eta \; \Delta(\underline{\mathbf{w}})$$

where

$\Delta(\underline{w})$ is the gradient and

η is the learning rate (small, positive)

Notes:

1. This moves us downhill in direction $\mathbf{\Delta}(\underline{\mathbf{w}})$ (steepest downhill direction)
2. How far we go is determined by the value of $\boldsymbol{\eta}$

ILLUSTRATION OF GRADIENT DESCENT

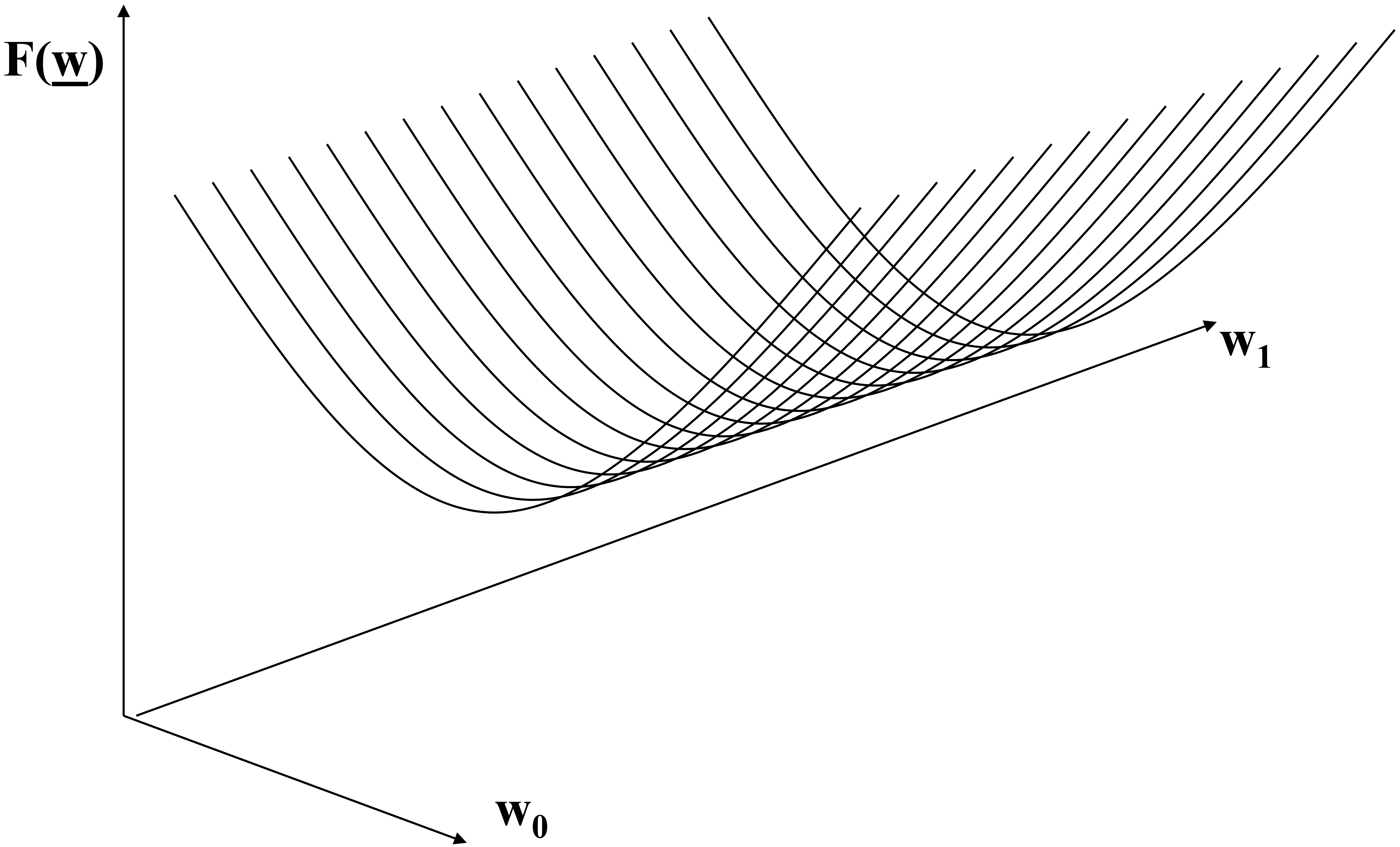


ILLUSTRATION OF GRADIENT DESCENT

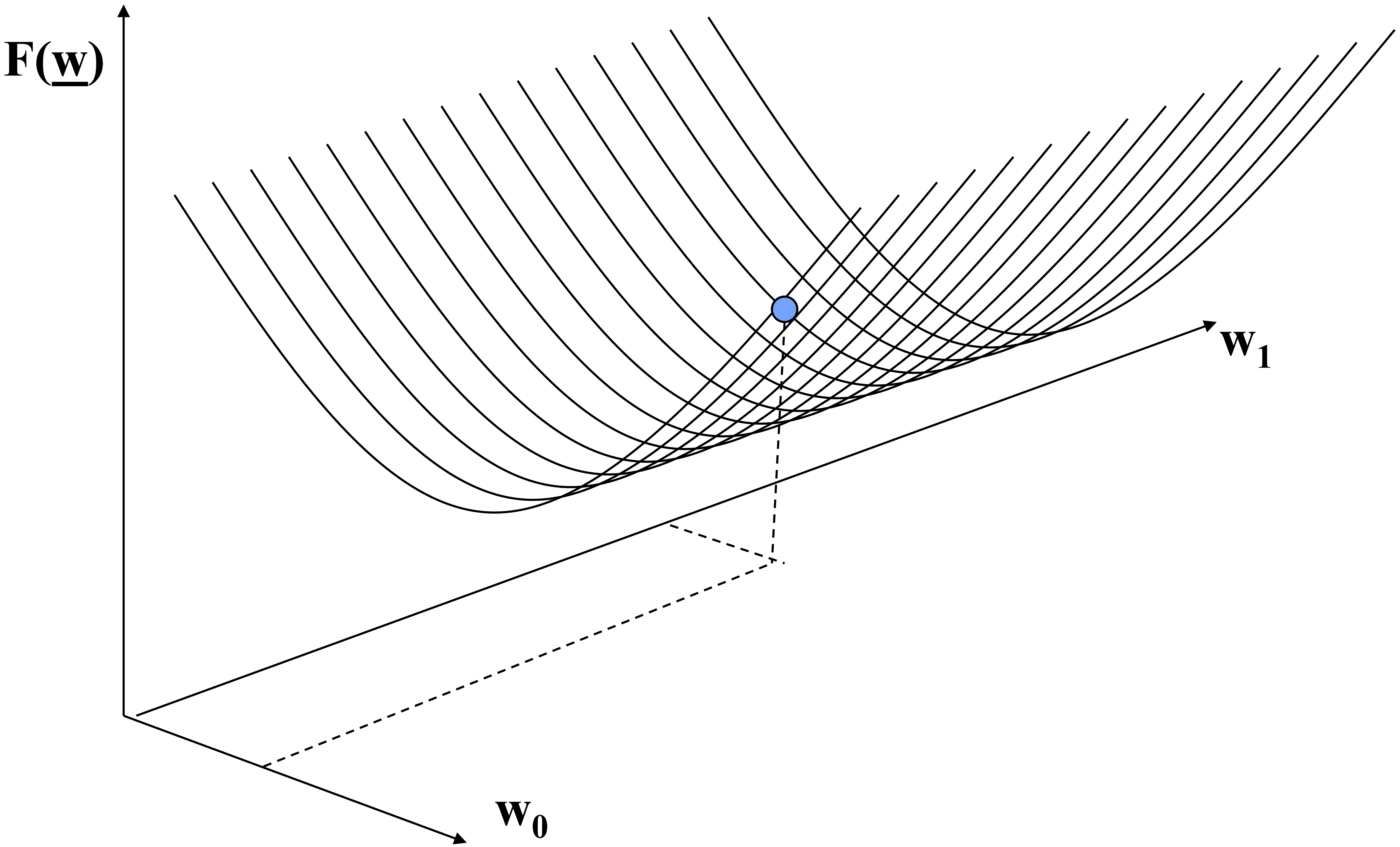


ILLUSTRATION OF GRADIENT DESCENT

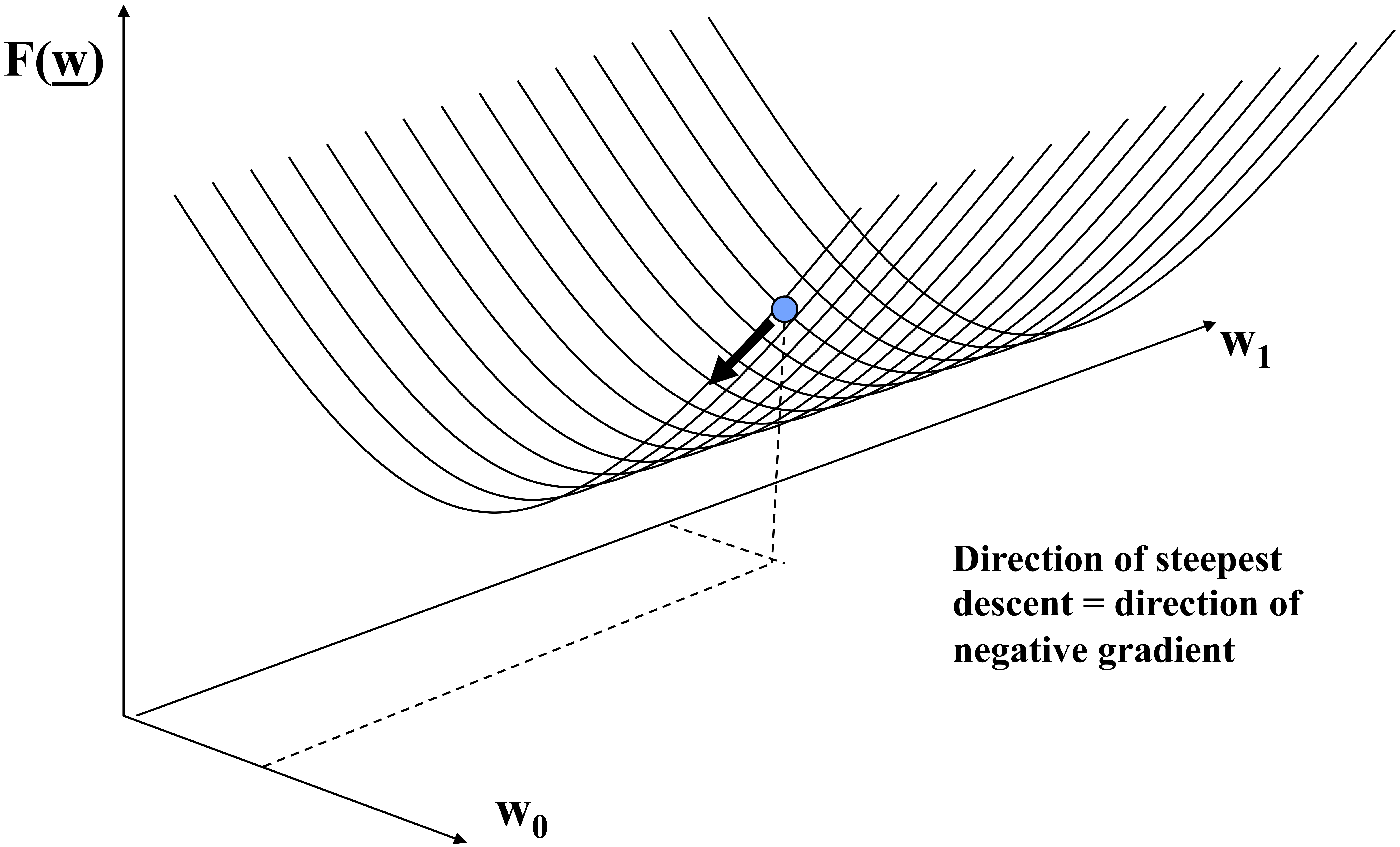
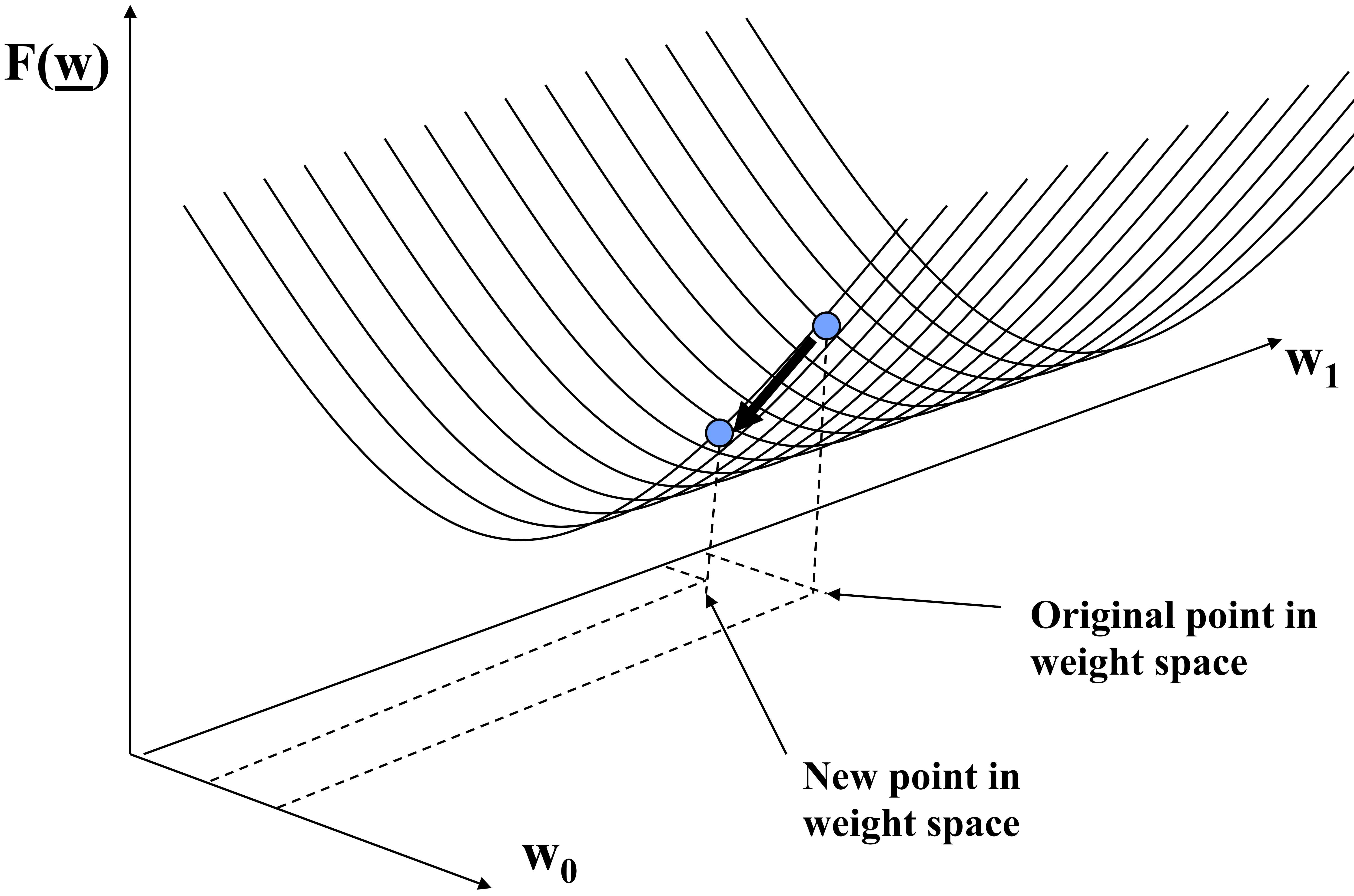


ILLUSTRATION OF GRADIENT DESCENT



For convex functions, when gradient descent converges, the solution is global minimum.

STOPPING CRITERIA FOR GRADIENT DESCENT

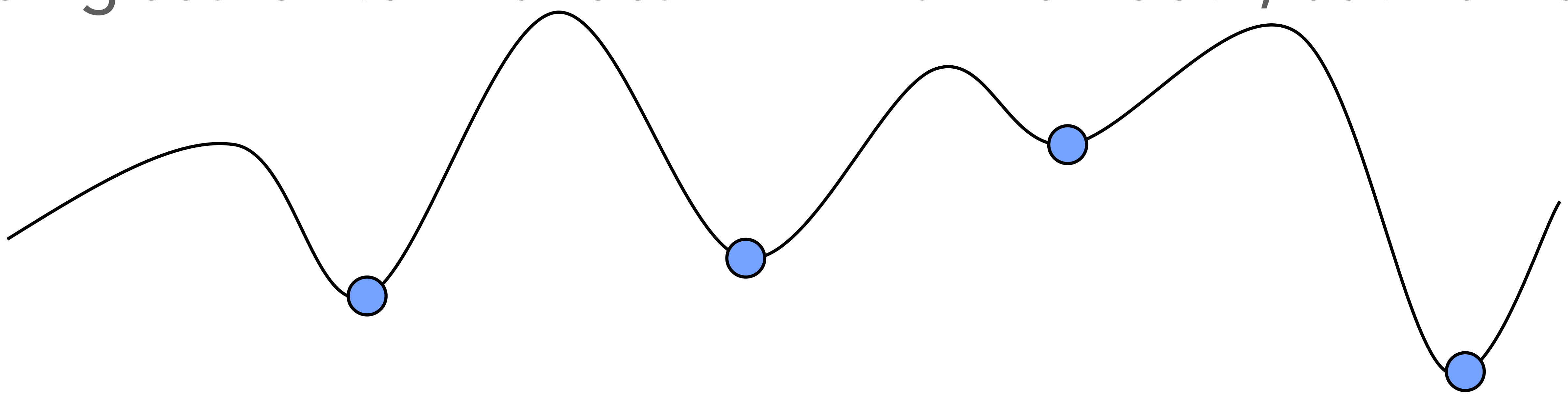
- Ideally, f′(x)=0…
- In practice…
  - $\|\nabla f(x)\| < \varepsilon$
  - $|f(x_{k+1}) - f(x_k)| < \varepsilon$
  - $\|x_{k+1} - x_k\| < \varepsilon$
  - Maximum number of iterations has been reached

GRADIENT ASCENT

- For concave functions that you want to *maximize*, take a step in direction of gradient (i.e., $w_{new} \leftarrow w_{old} + \eta\nabla(w)$)
- Otherwise same as gradient descent:
  - Start at some parameter values
  - Take derivative, move the parameters in the direction of gradient
  - Repeat until stopping criteria is met (e.g., gradient close to 0)

GRADIENT DESCENT FOR NON-CONVEX OPTIMIZATION

- Works on any objective function F(θ)
  - as long as we can evaluate the gradient Δ(θ)
  - this can be very useful for minimizing complex functions F
- Can be used in hill-climbing search to find local minima in smooth, but non-convex functions



- If function has multiple local minima, gradient descent goes to the closest local minimum:
  - solution: random restarts from multiple places in model space

LOGISTIC REGRESSION: RECAP

LOGISTIC REGRESSION

- Same parametric form as standard regression, but uses logistic function for binary classification

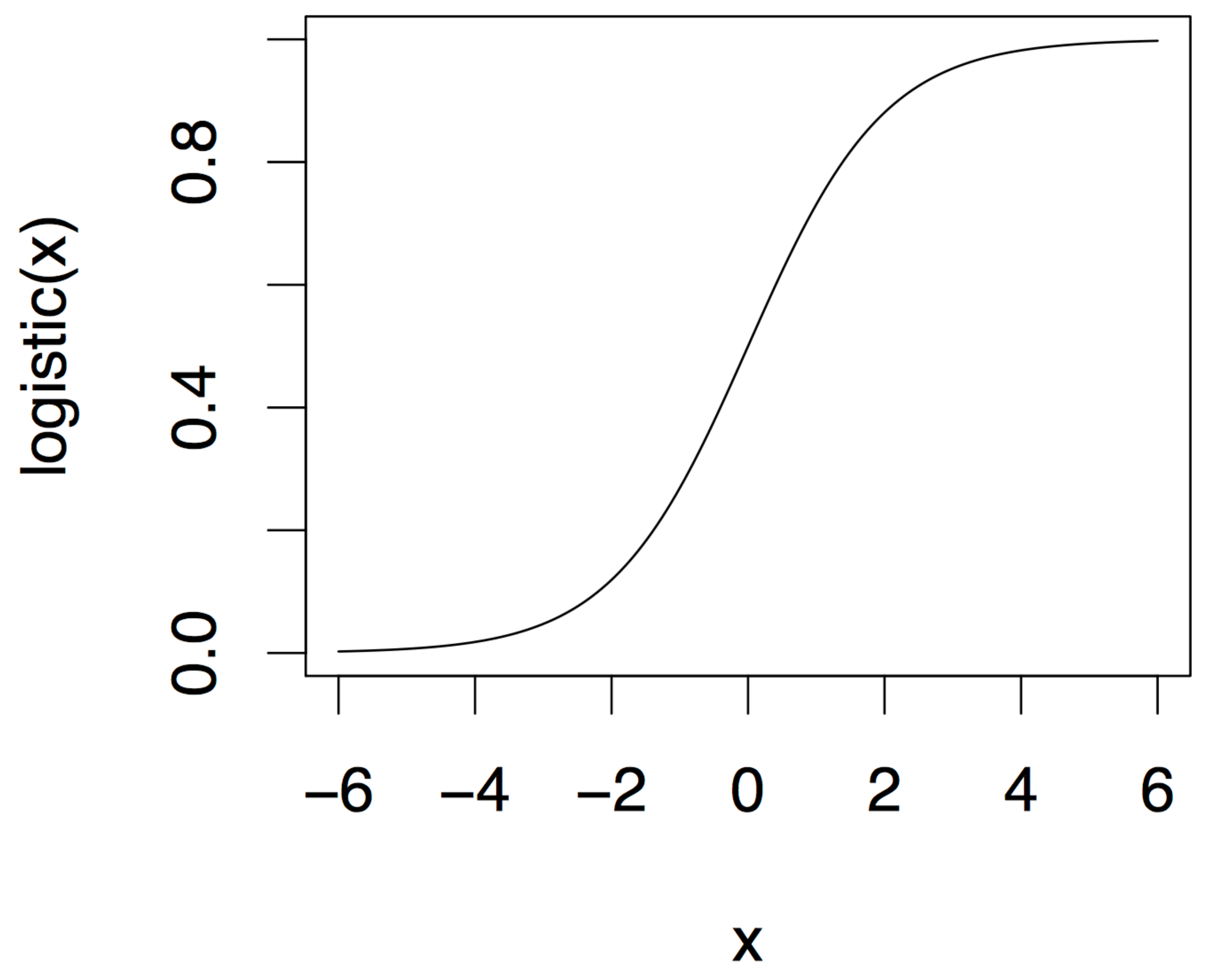
Logistic regression model:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x} + w_0)}}$$

- Output is the (positive) class probability rather than the binary prediction
- Logistic function transform ensures output is [0,1]

Logistic function:

$$\text{logistic}(x) := \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$



LR EXAMPLE

$$P(BC = 1|A, I, S, CR) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x})}}$$

$$\mathbf{x} = [Int, A, I, S, CR]$$
$$\mathbf{w} = [w_0, w_A, w_I, w_S, w_{CR}]$$

LR parameters = w

Intercept	Age>40	Income=high	Student=yes	Credit=fair	BuysComp?
1	0	1	0	1	0
1	0	1	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	0	1	0	0
1	0	0	1	0	1
1	0	0	0	1	0
1	0	0	1	1	1
1	1	0	1	1	1
1	0	0	1	0	1
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	0	0	0

- Score function: likelihood
- Estimate **w** with maximum likelihood estimation

LR LEARNING

- Score function: likelihood function

$$minimize \sum_{i=1}^{N} (-y_i \mathbf{w}^T \mathbf{x_i} + log(1 + e^{\mathbf{w}^T \mathbf{x_i}}))$$

- Estimate optimal **w** using gradient descent $\dfrac{dlogL}{dw_j} = \sum_{i=1}^{N} (-y_i + P(y_i = 1 \,|\, \mathbf{w}, \mathbf{x}_i))x_{ij}$

Gradient descent:

Start at some **w**, e.g., **w**=[0,0,0,0,0]

Make predictions given current **w**: $\forall i \;\; \widehat{y_i} = P(y_i = 1|\mathbf{x}_i) = \dfrac{1}{1 + e^{-\mathbf{w}^T \mathbf{x_i}}}$

Calculate gradient for each parameter: $\forall j \;\; \dfrac{d\,logL}{d\,w_j} = \left[\sum_{i=1}^{n}(-y_i + \hat{y}_i)x_{ij}\right] = \nabla_j$

Move parameters in direction of negative gradient: $\forall j \;\; w_j^{new} = w_j - \eta \nabla_j$

Repeat until stopping criteria is met

LR PREDICTION

Intercept	Age>40	Income=high	Student=yes	Credit=fair	BuysComp?
1	0	1	0	1	0
1	0	1	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	0	1	0	0
1	0	0	1	0	1
1	0	0	0	1	0
1	0	0	1	1	1
1	1	0	1	1	1
1	0	0	1	0	1
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	**0**	**1**	**0**	**0**	**?**

- What is the probability that new person will buy a computer?

$$\mathbf{x} = [1, 0, 1, 0, 0]$$

$$\mathbf{w} = [-1.3, 1, 2, -2, 0.7]$$

$$\mathbf{x}^T \mathbf{w} = 0.7$$

$$P(BC = 1 | \mathbf{x}\ ) = \frac{1}{1 + e^{-0.7}}$$
$$= 0.668$$

DEAL WITH OVERFITTING

- Simply finding the parameter values that lead to maximum likelihood function value in the training dataset may imply overfitting!
- Solution: add a **regularization term** in the scoring function to penalize complex models
  - e.g., L2 regularization term: $\frac{\lambda}{2}\|w\|^2$
  - λ is the regularization parameter; the larger the value, the more we are in favor of simple models

LR LEARNING WITH REGULARIZATION TERM

- Score function: likelihood with L2 regularization

$$minimize \sum_{i=1}^{N} (-y_i \mathbf{w}^T \mathbf{x}_i + log(1 + e^{\mathbf{w}^T \mathbf{x}_i})) + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- Estimate optimal **w** using gradient descent

Gradient descent:

Start at some **w**, e.g., **w**=[0,0,0,0,0]

Make predictions given current **w**: $\forall i \;\; \widehat{y}_i = P(y_i = 1|\mathbf{x}_i) = \frac{1}{1+e^{-\mathbf{w}^T \mathbf{x}_i}}$

Calculate gradient for each parameter: $\forall j \;\; \frac{d\, logL}{d\, w_j} = \left[\sum_{i=1}^{n} (-y_i + \hat{y}_i) x_{ij}\right] + \lambda w_j = \nabla_j$

Move parameters in direction of negative gradient: $\forall j \;\; w_j^{new} = w_j - \eta \nabla_j$

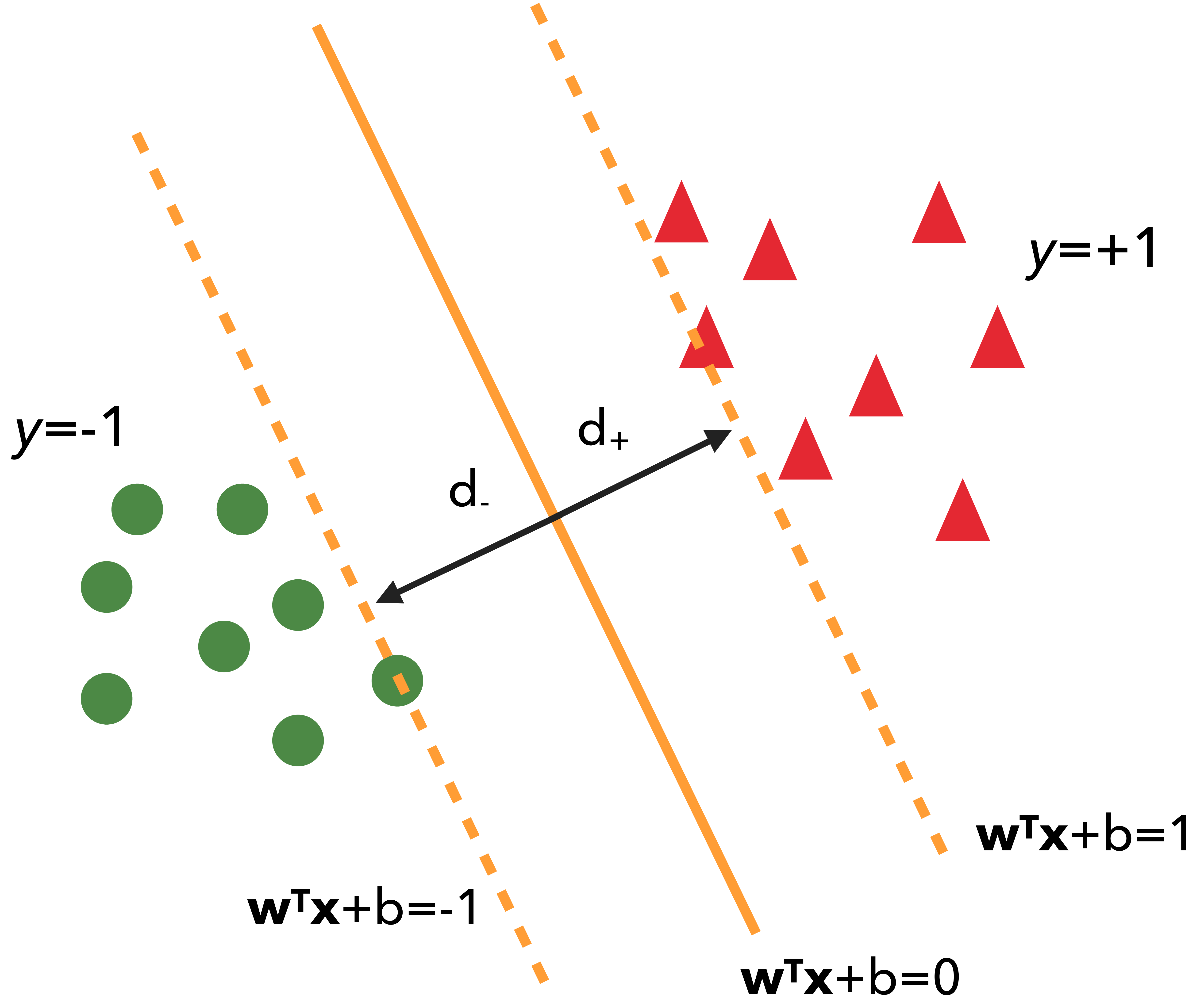
Repeat until stopping criteria is met

SVM: RECAP

SVM: KNOWLEDGE REPRESENTATION AND SCORING FUNCTION

- Linear SVM: $y = sign\left[\sum_{i=1}^{m} w_i x_i + b\right]$
- Margin = $d_+ + d_- = 2/||\mathbf{w}||$
- Optimization problem
  - max $2/||\mathbf{w}||$
  - subject to

    $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i \in \{1,2,\ldots,N\}$



SVM LEARNING

- Equivalent to minimize $||\mathbf{w}||^2/2$ subject to

$$y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) \geq 1, \forall i \in \{1,2,\ldots,N\}$$

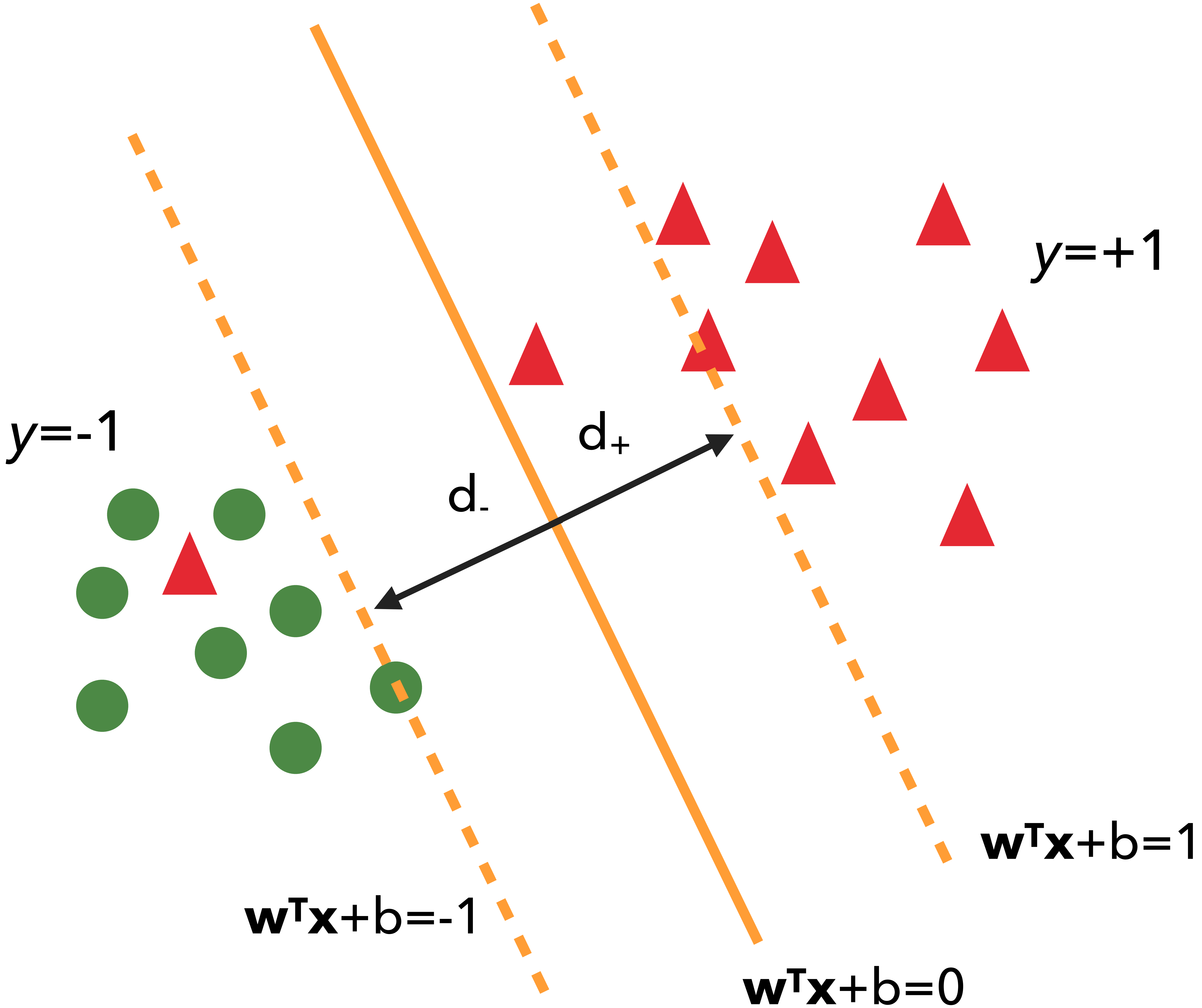
- This is a **quadratic optimization** problem subject to linear constraints, there is a unique minimum
- Lagrangian function $L(\mathbf{w}, b, \lambda_i) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{N} \lambda_i(1 - y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_\mathbf{i} + b))$

WHAT ABOUT LINEARLY NON-SEPARABLE DATA?

- Introduce slack variables $\varepsilon_i \geq 0$ such that:

  $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \varepsilon_i, \forall i \in \{1,2,\ldots,N\}$

- ε_i measures the amount of error
  - When $0 < \varepsilon_i \leq 1$, data is between the margin, but classified correctly
  - When $\varepsilon_i > 1$, data is misclassified



“SOFT” MARGIN OPTIMIZATION

- With slack variables the score function is:

$$\min_{\mathbf{w},\xi} ||\mathbf{w}||^2 + C\sum_{i}^{N} \xi_i$$

- And new constraints:

$$y_i(x_i \cdot w + b) - (1 - \xi_i) \geq 0 \;\; \forall i$$

- If ξ are sufficiently large, then every constraint can be satisfied
- C is regularization parameter
  - Small C means constraints can be ignored in order to find large margin
  - Large C means constraints cannot be ignored and result is small margin (C=∞ enforces hard margin)

SVM OPTIMIZATION

- Constraint can be rewritten as:

$$y_i f(x_i) \geq 1 - \xi_i \quad \forall i$$

- Together with $\xi_i \geq 0$, is equivalent to:

$$\xi_i = max\Big(0, 1 - y_i f(x_i)\Big)$$

- Hence we can use the following score in unconstrained optimization:

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_{i}^{N} \left[max\Big(0, 1 - y_i f(x_i)\Big) \right]$$

NEW OBJECTIVE

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_{i}^{N} \left[\boxed{max\Big(0, 1 - y_i f(x_i)\Big)} \right]$$

Hinge Loss

Points are in three categories:

1. $y_i f(x_i) > 1$
   Point is outside margin.
   No contribution to loss
2. $y_i f(x_i) = 1$
   Point is on margin.
   No contribution to loss.
   As in hard margin case.
3. $y_i f(x_i) < 1$
   Point violates margin constraint.
   Contributes to loss