Last time: $f(x) = |x| \Rightarrow \partial f(x) = \begin{cases} 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$

Ex: $f(x) = \|x\|_1$, $x \in \mathbb{R}^d$

$$\partial f(x) = \{ v : f(y) \geq f(x) + v^T (y - x) \}$$

claim

$$= \{ v : \|y\|_1 \geq \|x\|_1 + v^T (y - x) \}$$

$$= \{ \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} : |y_i| \geq |x_i| + v_i (y_i - x_i) \}$$

$\Rightarrow v \in \partial f(x)$ iff for $i = 1, \ldots, d$

$$v_i = \begin{cases} 1 & \text{if } x_i > 0 \\ [-1, 1] & \text{if } x_i = 0 \\ -1 & \text{if } x_i < 0 \end{cases}$$

Recall our motivating ex for subgradient descent:

$$f(\omega) = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i x_i^T \omega)_+ + \lambda \|\omega\|_1$$

for sparse SVM.

how to compute subgrads here?

we know how to compute a subgrad

Q: how to manipulate subgradient

Rules (<u>for cvx functions</u>)

a) Scaling: $\partial(af) = a\,\partial(f)$ if $a > 0$

b) $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$ (set. sum) Minkowski sum

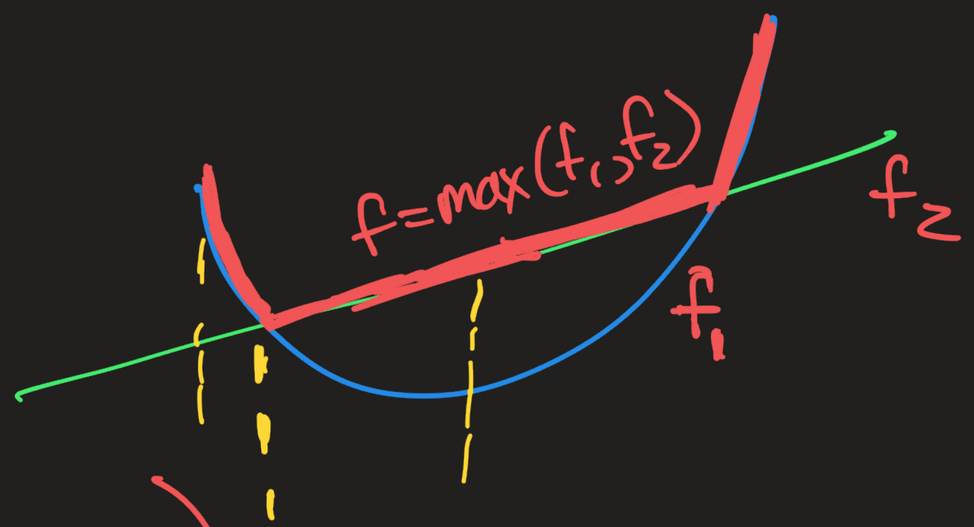
$= \{v + v : v \in \partial f_1, v \in \partial f_2\}$

c) if $g(x) = f(Ax + b)$ then $\partial g(x) = A^T \partial f(Ax + b)$

(chain rule)

d) Finite pointwise max:

$$f(x) = \max_{i=1,\dots,m} f_i(x)$$



$$\partial f(x) = \text{conv}\left(\bigcup_{i:\, f_i(x) = f(x)} \partial f_i(x) \right)$$

which is the convex hull of the union of the subdifferentials of all the active i:

the f_i s.t. $f_i(x) = f(x)$

e) Norm: Given $p \geq 1$ (maybe $p = \infty$)

p-norm
ℓ_p-norm

$$\|x\|_p = \left(\sum |x_i|^p\right)^{1/p}$$

(ex: $p=2$
$\|x\|_2 = \sqrt{\sum x_i^2}$

$p=1$
$\|x\|_1 = \sum |x_i|$)

and $p = \infty$,

$$\|x\|_p = \max_i |x_i|$$

(inf-norm, ℓ_∞-norm, max norm)

Let q be such that $\frac{1}{q} + \frac{1}{p} = 1$ (note $q \geq 1$)

p & q are called a conjugate pair

Note $(1, \infty)$ is a conjugate pair

if $f(x) = \|x\|_p$

then

$$\partial f(x) = \{ y : \|y\|_q \leq 1 \text{ and } y^T x = \|x\|_p \}$$

$(x^{1/2})' = \frac{1}{2} x^{-1/2}$

Ex of this rule

1) $p=2 \Rightarrow q=2$ $\quad f(x) = \|x\|_2 = \sqrt{\sum x_i^2}$

when $x \neq 0$: $\left(\nabla f(x)\right)_i = \frac{1}{2\sqrt{\sum x_j^2}} 2x_i = \frac{x_i}{\|x\|_2}$

$$\nabla f(x) = \frac{x}{\|x\|_2}, \quad x \neq 0$$

To evaluate $\partial f(0)$, use fact

$$\partial \|x\|_2 = \{ y : \|y\|_2 \leq 1 \text{ and } x^T y = \|x\|_2 \}$$

$$\Rightarrow \partial \|0\|_2 = \{ y : \|y\|_2 \leq 1 \text{ and } 0 = 0 \}$$

$$= \{ y : \|y\|_2 \leq 1 \}$$

Ex (using our subgradient rules)

$$f(w) = \frac{1}{n}\sum_{i=1}^{n}(1 - y_i w^T x_i)_+ + \lambda \|w\|_1$$

Then we have

$$\partial f(w) = \frac{1}{n}\sum_{i=1}^{n}\partial(1 - y_i w^T x_i)_+ + \lambda \partial \|w\|_1$$

we know $\partial\|w\|_1$. How to compute $\partial(1 - y_i w^T x_i)_+$

Let $g(w) = (1 - y_i w^T x_i)_+ = (1 - y_i x_i^T w)_+$

$= f(a^T w + b)$ where $f(x) = x_+$

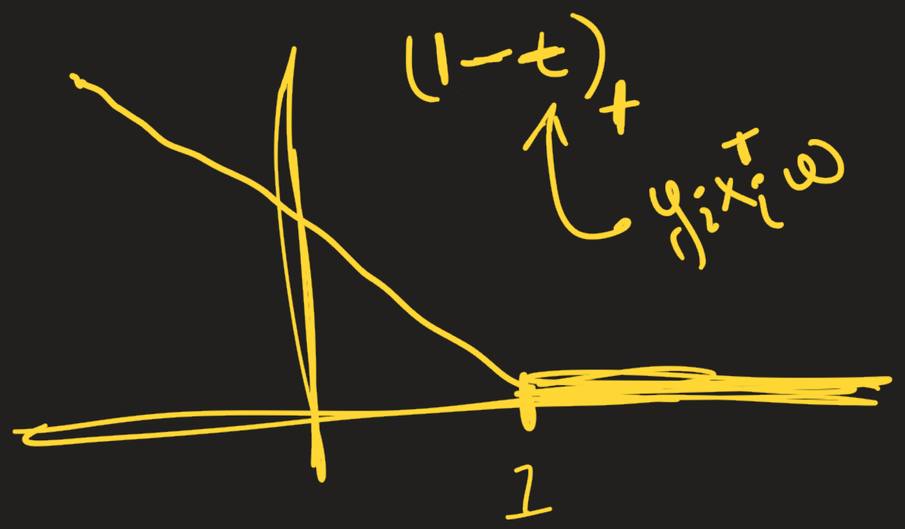
$a = -y_i x_i$

$b = 1$

So $\partial g(w) = a\, \partial f(a^T w + b)$ by chain rule

$$= a \begin{cases} 1 & \text{if } a^T w + b > 0 \\ [0,1] & \text{if } a^T w + b = 0 \\ 0 & \text{if } a^T w + b < 0 \end{cases}$$

$$= -y_i x_i \begin{cases} 1 & \text{if } 1 - y_i x_i^T w > 0 \\ [0,1] & \text{if } 1 - y_i x_i^T w = 0 \\ 0 & \text{if } 1 - y_i x_i^T w < 0 \end{cases}$$

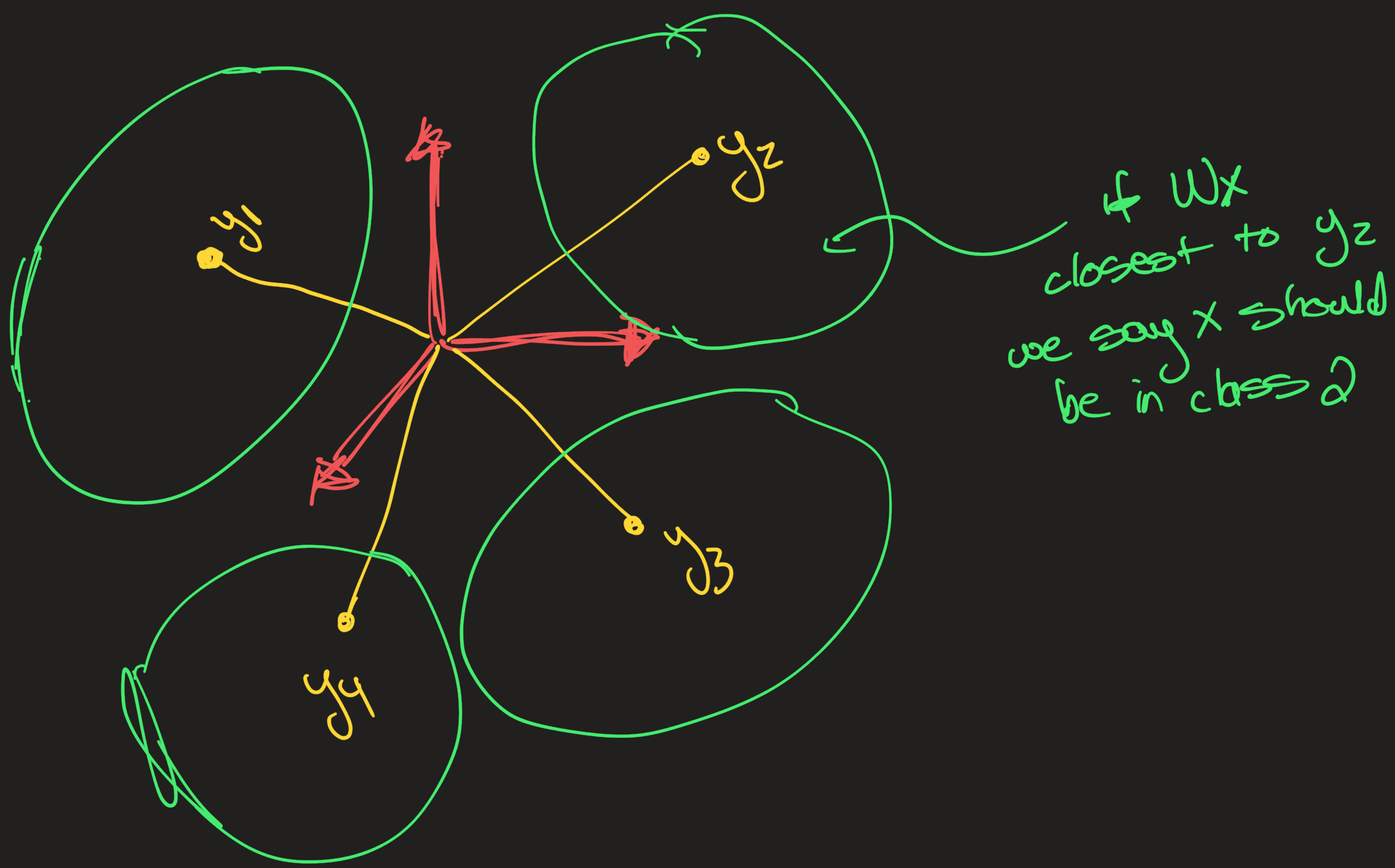


Ex ECOC (Error-correcting output coding)
and linear regression for classification (K-class)

Choose k different code words (vectors $\in \mathbb{R}^q$)
to represent our classes. So represent class i
with code-word $y_i \in \mathbb{R}^q$.

Fit a linear model $y = Wx$ that predicts y
given features x and take the class of a new
point x to be

$$i = \underset{i}{\operatorname{argmin}} \| y_i - Wx \|_2$$


y_1
y_2
y_3
y_4
if Wx closest to y_2 we say x should be in class 2

Train: learn W to minimize the distance between x_i and the codewords for the class y_i

$$f(W) = \frac{1}{n} \sum_{i=1}^{n} \| c_{y_i} - W x_i \|_2^2$$

$W \in \mathbb{R}^{l \times d}$

where c_{y_i} is the codeword for class y_i

then $f(W) = \frac{1}{n} \| Y - WX \|_F^2$

where $Y = \left[c_{y_1} \dots c_{y_n} \right] \in \mathbb{R}^{l \times n}$

$$X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$M = \begin{bmatrix} m_1 & \cdots & m_d \end{bmatrix}$$

$$= \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}$$

Aside $M \in \mathbb{R}^{n \times d}$

$$\|M\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{d} m_{ij}^2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{d} m_{ij}^2 \right)$$

$$= \sum_{i=1}^{n} \|r_i\|_2^2$$

$$= \sum_{j=1}^{d} \left(\sum_{i=1}^{n} m_{ij}^2 \right) = \sum_{j=1}^{d} \|m_j\|_2^2$$

$$\|M\|_F^2 = \sum_{j=1}^{d} \|m_j\|_2^2 \quad \text{where} \quad M = \left[m_1 \mid \cdots \mid m_d \right]$$

Consider $M = Y - WX = \left[c_{y_1} \cdots c_{y_n} \right] - W \left[x_1 \cdots x_n \right]$

$$= \left[c_{y_1} - W x_1 \mid \cdots \mid c_{y_n} - W x_n \right]$$

$$\Rightarrow \|Y - WX\|_F^2 = \sum_{i=1}^{n} \|c_{y_i} - W x_i\|_2^2 = n f(\omega)$$

$$\Rightarrow f(\omega) = \frac{1}{n} \|\varphi - WX\|_F^2$$

Fact $\|M\|_F^2 = Tr(M^T M) = Tr(M M^T)$

recall $Tr(A) = \sum_{i=1}^{n} A_{ii}$ if $A \in \mathbb{R}^{n \times n}$

Facts

1) $Tr(AB^T) = Tr(B^T A)$ if A & B have the same dimensions

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

3) $Tr(\alpha A) = \alpha\, Tr(A)$

These imply that if A & B have the same dimensions, then $\langle A, B \rangle = Tr(AB^T)$ is a valid inner product

An inner-product generalizes the dot-product for two vectors $v, w \in \mathbb{R}^d$:

$$\langle v, w \rangle = v^T w$$

The dot product satisfies

1) $\langle v, w \rangle = \langle w, v \rangle$ symmetry

2) $\langle \alpha v + w, u \rangle = \alpha \langle v, u \rangle + \langle w, u \rangle$ for $\alpha \in \mathbb{R}$

linearity

3) $\langle v, v \rangle = \|v\|_2^2 \geq 0$ and $\langle v, v \rangle = 0$ iff $v = 0$

positive-definiteness

For matrices, $\langle A, B \rangle = \mathrm{Tr}(AB^T)$ has the same properties: symmetry, linearity, and positive-definiteness.

And

$$\| M \|_F^2 = \mathrm{Tr}(MM^T) = \langle M, M \rangle$$

Just as in the vector case $\left(\nabla_v \|v\|_2^2 = 2v \right)$,

$$\nabla_M \|M\|_F^2 = 2M$$

To compute $\nabla f(W)$, note

$$f(W) = \frac{1}{n} \| Y - WX \|_F^2 \Rightarrow$$

$$\nabla f(W) = \frac{-2}{n} (Y - WX) X^T \in \mathbb{R}^{k \times d}$$

With this expression we can use the gradient descent algorithm to fit an ECOC classification model.

How do we choose a stepsize? In this case, f is in fact β-smooth; let's determine β

$f: \mathbb{R}^d \to \mathbb{R}$

Recall that f is β-smooth if for all $\omega, \omega_0 \in \mathbb{R}^d$

$$f(\omega) \leq f(\omega_0) + \langle \nabla f(\omega_0), \omega - \omega_0 \rangle + \beta \|\omega - \omega_0\|_2^2$$

For $f: \mathbb{R}^{l \times d} \to \mathbb{R}$, we need that for all $W, W_0 \in \mathbb{R}^{l \times d}$,

$$f(W) \leq f(W_0) + \langle \nabla f(W_0), W - W_0 \rangle + \beta \|W - W_0\|_F^2$$

To verify this, note

$$f(W) = \frac{1}{n}\|Y - WX\|_F^2 = \frac{1}{n}\|Y - \omega_0 X + (\omega_0 X - \omega X)\|_F^2$$

Now apply the identity

$$\|A+B\|_F^2 = \langle A+B, A+B\rangle = \langle A, A\rangle + 2\langle A, B\rangle + \langle B, B\rangle$$
$$= \|A\|_F^2 + \|B\|_F^2 + 2\langle A, B\rangle$$

with $A = \Psi - \omega_0 X$ and $B = (\omega_0 - \omega)X$ to see that

$$f(\omega) = \frac{1}{n}\|\Psi - \omega_0 X\|_F^2 + \frac{2}{n}\langle \Psi - \omega_0 X, (\omega_0 - \omega)X\rangle + \frac{1}{n}\|(\omega_0 - \omega)X\|_F^2$$

Now apply the identity

$$\langle A, BC\rangle = \mathrm{Tr}(A(BC)^T) = \mathrm{Tr}(BCA^T) = \mathrm{Tr}(B(AC^T)^T)$$
$$= \langle AC^T, B\rangle$$

with $A = \Psi - \omega_0 X$

$B = \omega_0 - \omega$

$C = X$

to see that

$$f(W) = f(W_0) + \frac{2}{n}\langle (Y - W_0 X) X^T, W_0 - W \rangle$$

$$+ \frac{1}{n} \| (W_0 - W) X \|_F^2$$

$$= f(W_0) + \langle \nabla f(W_0), W - W_0 \rangle + \frac{1}{n} \| (W_0 - W) X \|_F^2$$

Now note that

$$\frac{1}{n} \| (W_0 - W) X \|_F^2 \leq \frac{1}{n} \| X \|_2^2 \, \| W_0 - W \|_F^2$$

where $\| X \|_2$ is the spectral norm of X, so we have

$$f(W) \leq f(W_0) + \langle \nabla f(W_0), W - W_0 \rangle + \beta \| W_0 - W \|_F^2$$

where

$$\beta = \frac{1}{n} \| X \|_2^2$$

We therefore see that gradient descent will converge when $\alpha = \frac{1}{\beta} = \frac{n}{\|X\|_2^2}$ is used as the stepsize, at rate

$$f(w_T) - f(w_*) = O\left(\frac{\beta}{\varepsilon}\right)$$

Of course we can directly solve for w_* using OLS:

$$w_* = y X^T (X X^T)^{-1}$$

In practice we would do this as long as d is small enough that the $O(d^3)$ operation of inverting XX^T is feasible. Eg if $d = 50{,}000$ we would then use methods like gradient descent or, as well see next class, more likely stochastic gradient descent