



# Aprendizaje de Máquina

---

ITAM

Semestre agosto-diciembre 2017



# Menú

---

- Métodos Lineales
  - Regresión
    - Mínimos Cuadrados Estático
    - Mínimos Cuadrados Iterativo



# Regresión Lineal

---

- Tipo de Método
  - Supervisado
  - Regresión: valor numérico
- Qué suponemos de los datos?
  - Atributos numéricos
  - Muestras de datos tomadas, preferiblemente, i.i.d (independientes e idénticamente distribuidas)
- Aplicaciones
  - Predicción de tendencias, precios,...



# Regresión Lineal

---

- Como es un método de aprendizaje supervisado, requerimos que cada dato de entrenamiento tenga un valor numérico asociado
  - $(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)$
  - Donde cada  $X_i$  es un registro completo
    - $X_i = \langle \text{atrib}_1, \text{atrib}_2, \dots, \text{atrib}_p \rangle$
    - Cada atributo tiene que ser numérico (o haber sido transformado en numérico)
  - Donde las  $y_i$  son valores numéricos y representan el valor de entrenamiento de la función objetivo



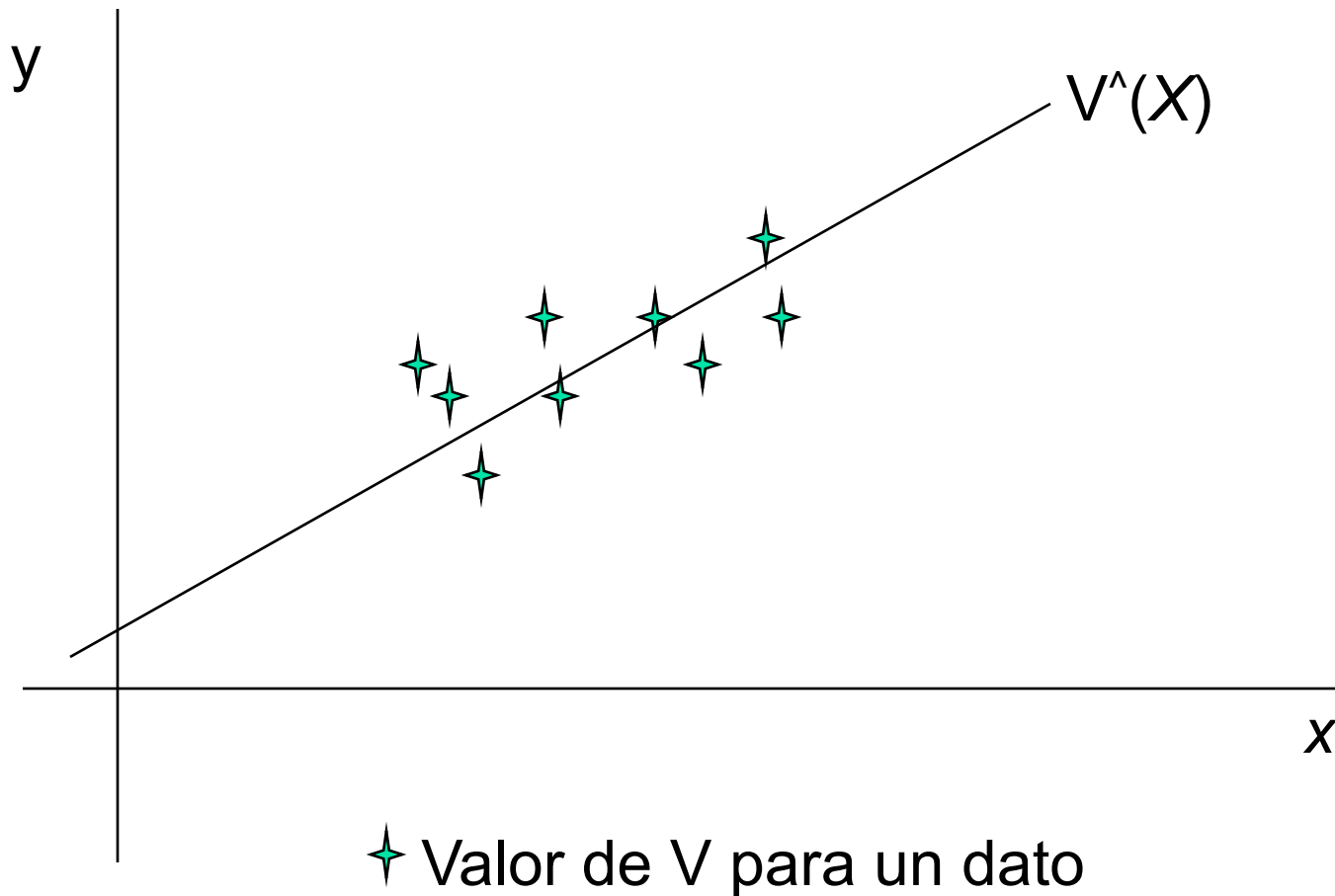
# Regresión Lineal

---

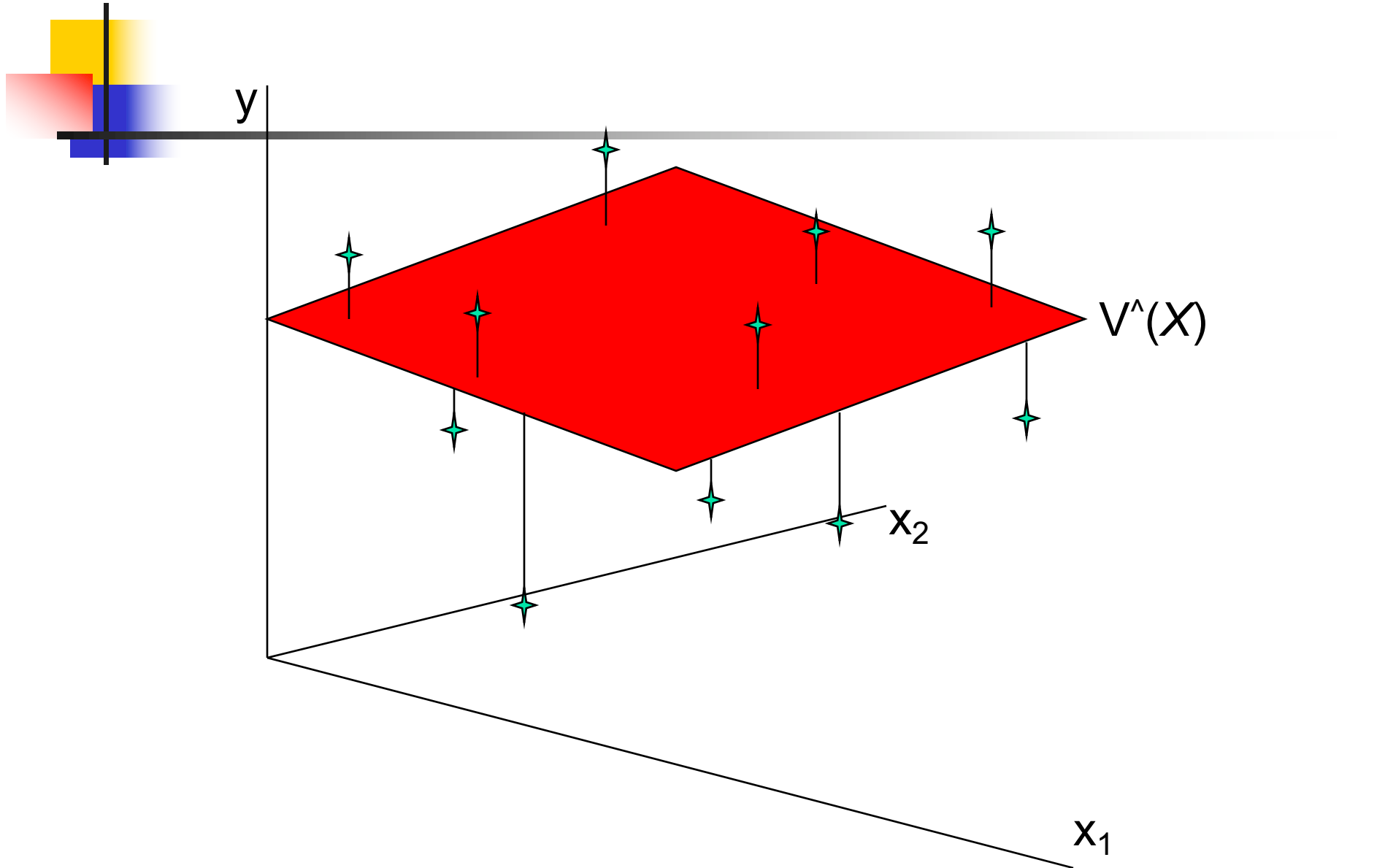
- El objetivo es poder determinar el valor de  $y$  para datos nuevos, por ejemplo:
  - Datos:
    - $\langle \text{Línea de crédito, saldo} \rangle$  y tenemos la deuda asociada a esos datos ( $\langle \text{Línea de crédito, saldo} \rangle, \text{deuda}$ )
  - Generar un modelo para:
    - “predecir” la deuda de un individuo del cual sólo conocemos su línea de crédito y saldo
- Estos métodos asumen que la función que se intenta estimar  $V$  es adecuadamente aproximada por una recta, un plano o hiper-plano
  - $V$  es una función desconocida (que suponemos existe) que determina la relación entre las  $X$  y las  $y$



# Aproximación Lineal de $V$



# Aproximación Lineal de $V$





# Regresión Lineal

---

- El modelo de regresión lineal tiene la siguiente forma:
  - $V^{\wedge}(X) = w_0 + \sum x_j w_j$
  - $V^{\wedge}$  es la aproximación lineal a  $V$ , la función que verdaderamente describe los datos
  - La suma es sobre todos los atributos  $x_j$  del dato  $X$
  - Las  $w_j$ 's son los coeficientes de la función
- Los métodos de regresión lineal buscan encontrar valores para los parámetros  $w_j$ 
  - Encontrar los valores de las  $w$  es aprender

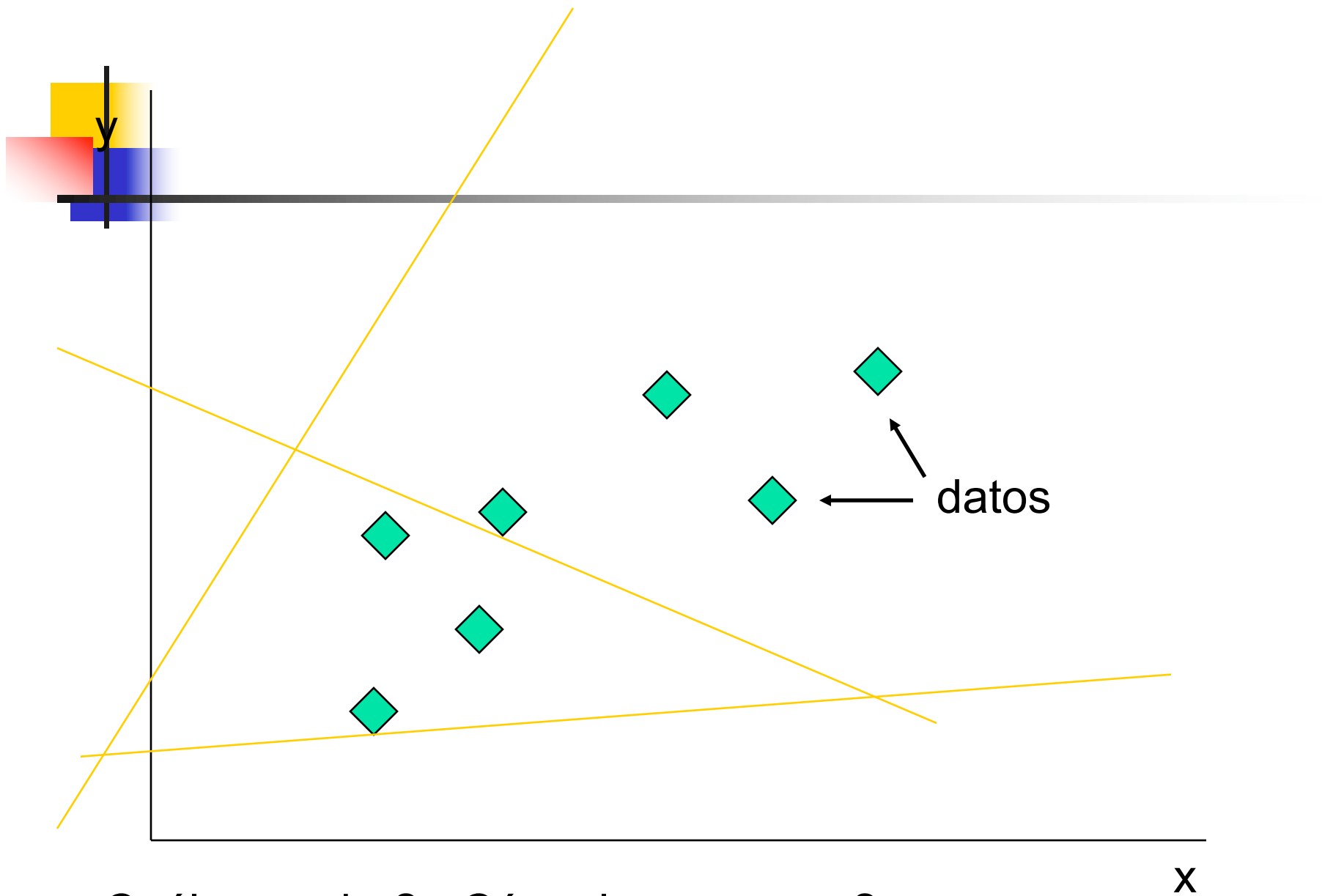




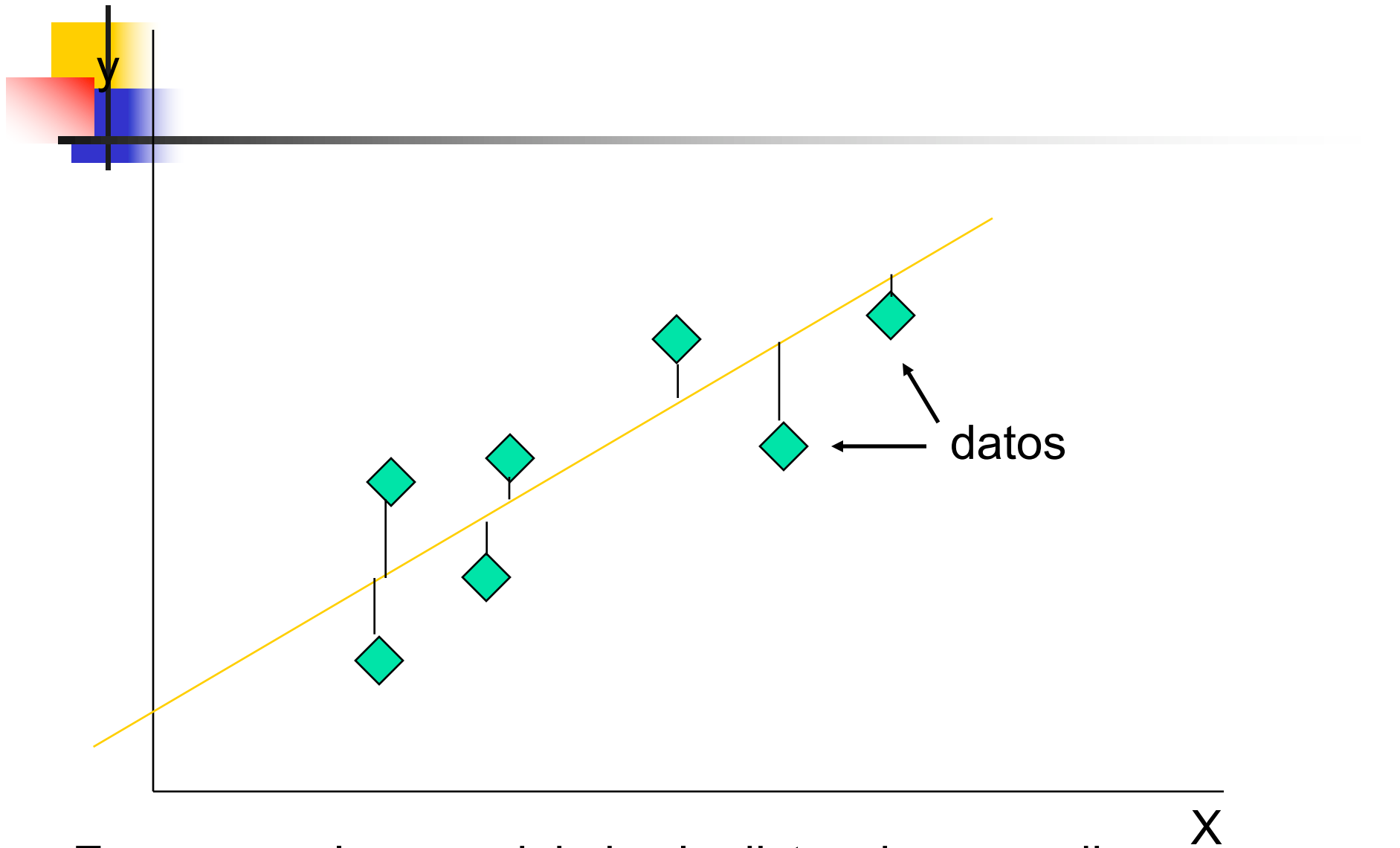
# Regresión Lineal

---

- Para esto necesitamos definir una medida de “ajuste” de nuestro modelo a los datos
  - Para valores dados de las  $w$ , qué tan bien se ajusta a los datos de entrenamiento
- Requerimos, pues, una medida de error
  - Una vez definida dicha medida, el algoritmo de aprendizaje la utiliza para ajustar el modelo y minimizar el error



¿Cuál es mejor? ¿Cómo la movemos?



Escogemos la que minimice la distancia promedio



# Regresión Lineal

## Mínimos Cuadrados

---

- La medida más común para definir el grado de ajuste se conoce como mínimos cuadrados
  - Tomamos las diferencias al cuadrado de cada valor  $y_i$  con lo calculado por  $V^{\wedge}$  para los atributos correspondientes al dato  $x_i$
  - Intentamos minimizar la suma de esta medida para todos los datos de entrenamiento

$$MinCuad(B) = \sum_{i=1}^N \left( y_i - W_0 - \sum_{j=1}^p x_{i,j} W_j \right)^2$$

- Donde N es el número de datos y p el número de atributos en cada dato
- ¿Cómo lo minimizamos?



# Regresión Lineal

## Mínimos Cuadrados

---

- Supongamos por un momento que sólo tenemos un atributo  $x$  y su correspondiente valor  $y$  para cada uno de los  $N$  datos:
  - $(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)$
  - $(\langle x_{1,1} \rangle, y_1), (\langle x_{2,1} \rangle, y_2), \dots, (\langle x_{N,1} \rangle, y_N)$
- El modelo es una recta
  - $V^{\wedge}(X_i) = w_0 + x_{i,1} w_1$
  - $w_0$  es la ordenada al origen y  $w_1$  es la pendiente



# Regresión Lineal

## Mínimos Cuadrados

---

- Y el error:
  - $\text{Mincuad}(W) = \sum^N (y_i - V^{\wedge}(X_i))^2$
  - $\text{Mincuad}(W) = \sum^N (y_i - w_0 - x_{i,1}w_1)^2$
- Para minimizarlo, primero derivamos
  - $d/w_0 = - \sum^N 2(y_i - w_0 - x_{i,1}w_1)$
  - $d/w_1 = - \sum^N 2(y_i - w_0 - x_{i,1}w_1)(x_{i,1})$
- Segundo, igualamos la primer derivada a cero
  - $- \sum^N 2(y_i - w_0 - x_{i,1}w_1) = 0$
  - $- \sum^N 2(y_i - w_0 - x_{i,1}w_1)(x_{i,1}) = 0$



# Regresión Lineal

## Mínimos Cuadrados

---

- Resolvemos para  $w_0$  y  $w_1$

- $w_0 N + w_1 \sum^N x_{i,1} = \sum^N y_i$

- $w_0 \sum^N x_{i,1} + w_1 \sum^N x_{i,1}^2 = \sum^N x_{i,1} y_i$



# Regresión Lineal

## Mínimos Cuadrados

---

- Generalización a p atributos
  - $\text{MinCuad}(\mathbf{W}) = (\mathbf{y} - \mathbf{XW})^T (\mathbf{y} - \mathbf{XW})$
  - Donde  $\mathbf{X}$  es una matriz con N renglones y p+1 columnas. La primer columna tiene 1's y se utilizan para multiplicar a  $W_0$
  - Donde  $\mathbf{W}$  es el vector de p+1  $w_i$ 's
  - Donde  $\mathbf{y}$  es el vector de valores para la función objetivo de las  $\mathbf{X}$





# Regresión Lineal

## Mínimos Cuadrados

---

- La derivada es:
  - $d/d\mathbf{W} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{XW})$
- Igualando a cero y resolviendo para  $\mathbf{W}$ 
  - $\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$
- Una vez que se tienen los valores para las  $w_i$ 's, encontrar el valor de  $V^\wedge$  para un nuevo dato se lo logra sustituyendo los valores de las  $w_i$ 's y los valores de las  $X$  del dato en:
  - $V^\wedge(X_i) = w_0 + \sum x_{i,j} w_{i,j}$



# Ejemplo

---

- Haga una regresión lineal utilizando los datos provistos por el profesor (reglin)
  - Utilize sklearn (LinearRegression) y el ipython Notebook
  - Siga el procedimiento para generar modelos delineado por el profesor en clase
  - Grafique los datos y el resultado del modelo
  - Grafique como cambia el error con el valor de los pesos  $W$ 
    - Para diferentes valores de  $W$  calcule el error del modelo y grafique. Dónde quedan las  $W$  encontradas por sklearn
- Repita para los archivos reglin2 y 3
  - Anote sus comentarios
  - Qué podemos hacer?



# Métodos de Regresión Lineal

## Comentarios

---

- Estos métodos asumen que la función objetivo puede aproximarse linealmente
  - En muchas ocasiones esta simplificación da muy buenos resultados
- Es importante tener una buena muestra de datos y de los correspondientes valores de la función de evaluación para entrenar



# Regresión Lineal

## Método Dinámico

---

- Supongamos que los datos no se encuentran todos disponibles al mismo tiempo, que nuevos datos se hacen disponibles en cualquier momento
- Lo que deseamos es un método incremental que sea capaz de integrar la información contenida en una nueva observación, sin necesidad de reentrenar con todos los datos
  - En ocasiones no solamente queremos incorporar información nueva, sino que también deseamos que información vieja pierda influencia



# Regresión Lineal

## Método Dinámico

---

- Este es el algoritmo que va a aproximar incrementalmente la función objetivo  $V$ , i.e., aprende  $V^{\wedge}$
- Requiere de ejemplos con un valor asociado para entrenar. Cada ejemplo es una tupla:
  - $(X, y)$
  - Donde:
    - $X$  es un datos con  $p$  atributos
    - $y$  es el valor que se le asigna a un dato durante el entrenamiento



# Regresión Lineal

## Método Dinámico

---

- Nuestra función es la siguiente combinación lineal
  - $V^{\wedge}(X_i) = w_0 + \sum w_{i,j} x_{i,j}$
  - $V^{\wedge}(X_i) = w_0 x_{i,0} + w_1 x_{i,1} + w_2 x_{i,2} + w_3 x_{i,3} + w_4 x_{i,4} + w_5 x_{i,5}$ 
    - Truco, insertar en todos los datos el atributo  $X_0=1$
- Donde
  - las  $w$ 's son coeficientes numéricos que determinan la importancia relativa de cada término
  - Las  $x_{i,j}$  son los atributos del dato  $X_i$



# Regresión Lineal

## Método Dinámico

---

- Al igual que en el método estático necesitamos ajustar los pesos  $w_i$  de nuestra función
  - $V^{\wedge}(X_i) = w_0x_{i,0} + w_1x_{i,1} + w_2x_{i,2} + w_3x_{i,3} + w_4x_{i,4} + w_5x_{i,5}$
  - Ajustar los pesos correctamente es, en este caso, aprender
- Nuestro objetivo es encontrar el valor de los pesos  $w_i$  que hagan que nuestra función se ajuste a los ejemplos de entrenamiento
- ¿Qué es que se ajuste bien?



# Método Dinámico

## Mínimos Cuadrados

---

- Una manera posible de medir esto, es encontrar los valores para las  $w_i$ 's que minimicen el error cuadrático
  - $(y - \hat{V}(X))^2$
  - Es la diferencia entre el valor de entrenamiento y lo que estima nuestra función (al cuadrado)
- Hay muchos algoritmos que realizan esto.
- Hoy veremos la regla LMS “least mean squares” (El menor error cuadrático medio)  
También conocida como la regla Widrow-Hoff





# Algoritmo de Aprendizaje LMS

---

- Para cada ejemplo de entrenamiento  $(X, y)$ 
  - Calcule  $V^{\wedge}$  con las  $w'$  s actuales
  - Para cada  $w_i$  ,
    - $w_i \leftarrow w_i + \eta(y - V^{\wedge}(X)) x_i$
- Donde  $\eta$  es una constante pequeña menor a 1, e.g. 0.1
- La regla se aplica iterativamente un número fijo de veces ó hasta que se logran los errores deseados ó si no se detecta progreso o en cada ocasión que se presenta un dato

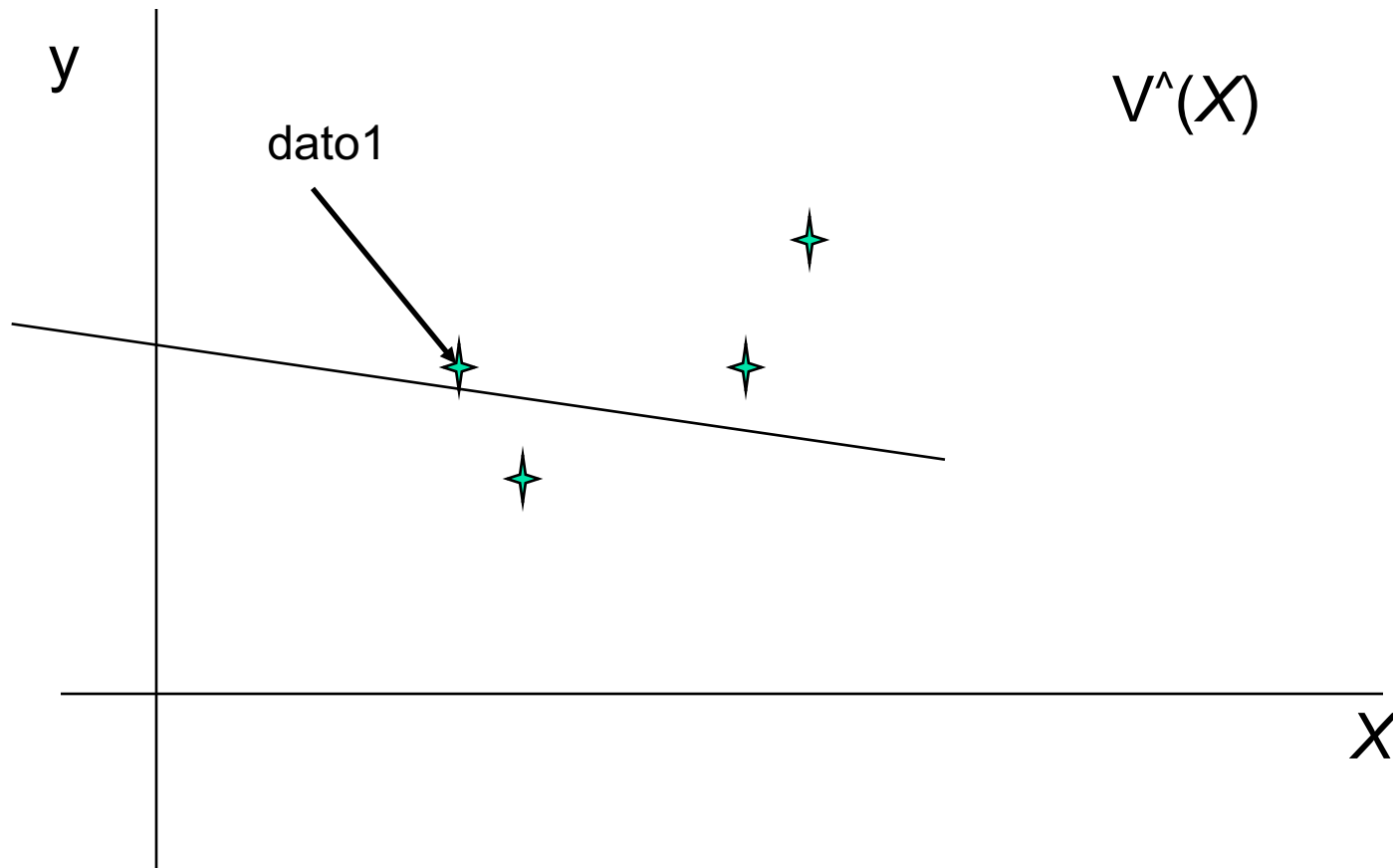


# Algoritmo de Aprendizaje LMS

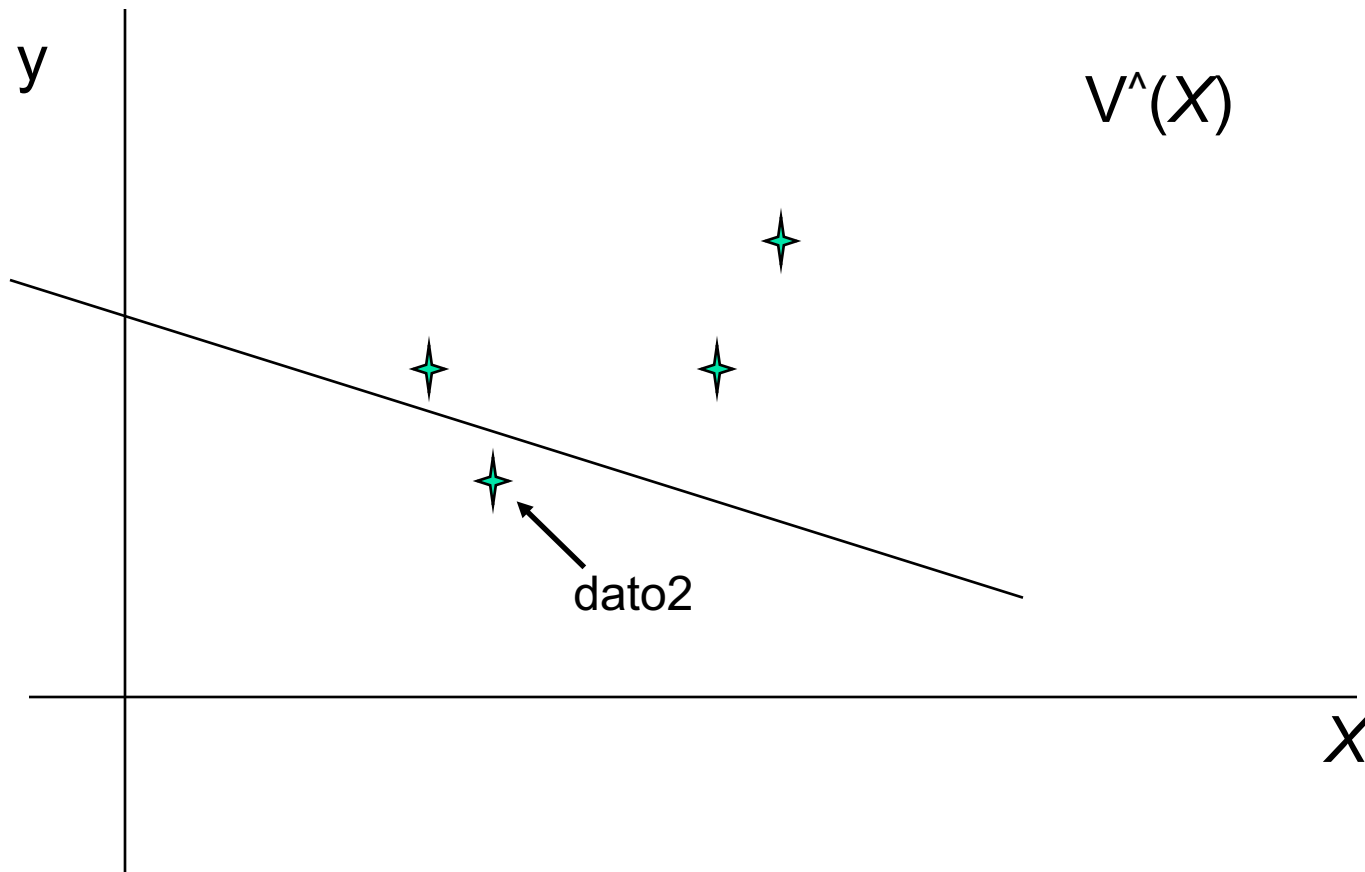
---

- ¿Porqué sirve?  $w_i \leftarrow w_i + \eta(y - V^{\wedge}(X)) x_i$ 
  - Cuando el error  $(y - V^{\wedge}(X))$  es cero nada cambia
  - Cuando es positivo ( $V^{\wedge}(X)$  es muy bajo) cada peso se modifica en proporción al valor de la  $x_i$  correspondiente. Esto aumenta  $V^{\wedge}(X)$  y reduce el error
  - Análogamente cuando es negativo
  - Nótese que si alguna  $x_i$  vale cero el peso correspondiente no cambia. Sólo se ajustan los pesos de las variables que contribuyen en la instancia

# Aproximación Lineal de $V$

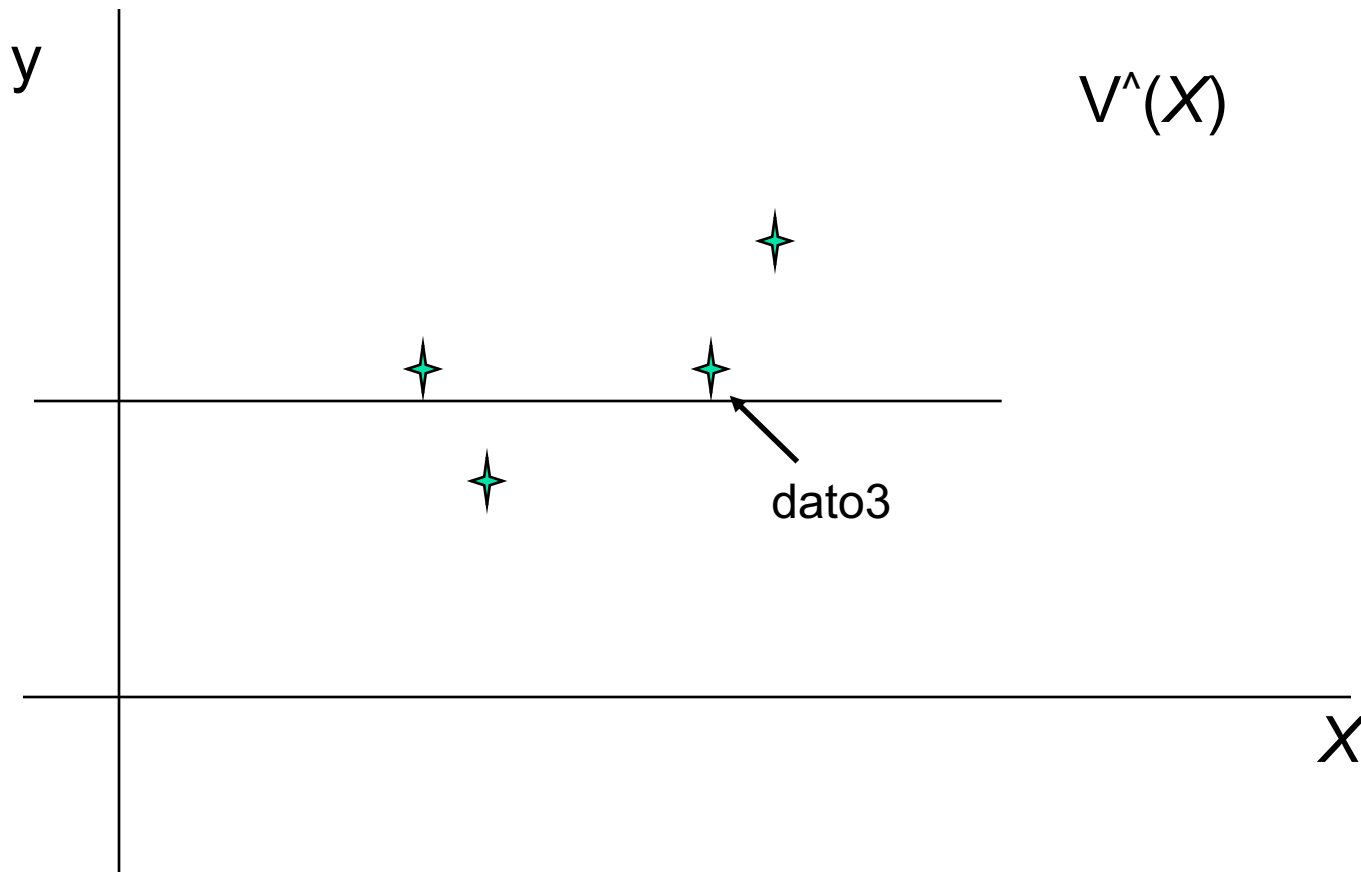


# Aproximación Lineal de $V$

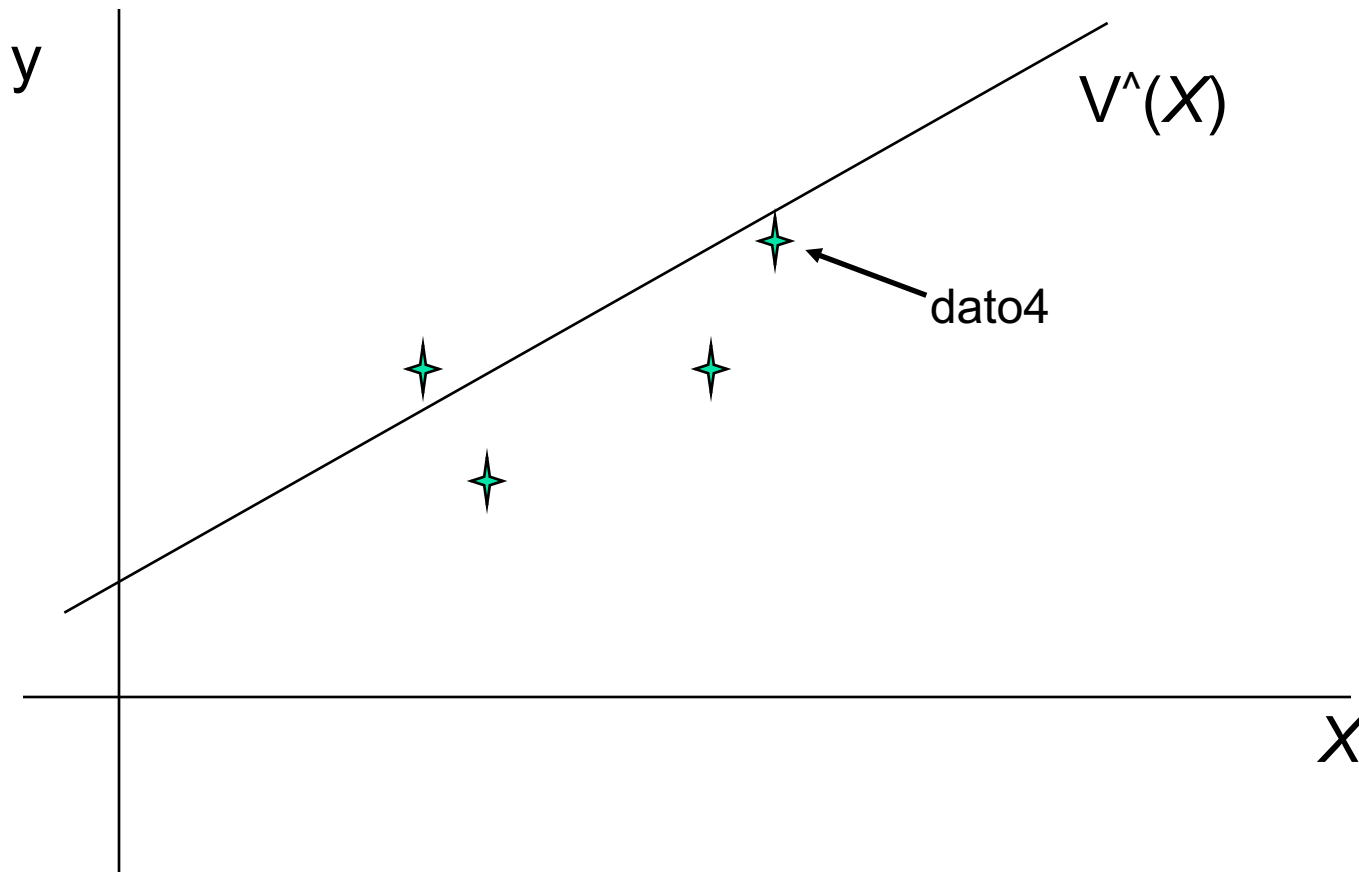




# Aproximación Lineal de $V$



# Aproximación Lineal de V





# Algoritmo Dinámico

## Ejemplo

---

- Datos:
  - $(\langle 1, 0, 1, 2, 0, 0 \rangle, 1)$
  - $(\langle 1, 1, 0, 2, 0, 1 \rangle, -1)$



# Algoritmo Dinámico

## Ejemplo

	x0	x1	x2	x3	x4	x5
$x's$	1	0	1	2	0	0
$w's$	1	1	1	1	1	1
$x_i w_i$	1	0	1	2	0	0

$y=1$ ,  $V^{\wedge}(X)=4$ ,  $\text{Error}=1-4=-3$ ,  $\eta =0.1$

$$w_0 = 1 + 0.1(-3)1 = 0.7$$

$$w_2 = 1 + 0.1(-3)1 = 0.7$$

$$w_3 = 1 + 0.1(-3)2 = 0.4$$

El nuevo error con este ejemplo es  $1-2.2=-1.2$





# Algoritmo Dinámico

## Ejemplo

	x0	x1	x2	x3	x4	x5
x's	1	1	0	2	0	1
w's	0.7	1	0.7	0.4	1	1
$x_i w_i$	0.7	1	0	0.8	0	1

$y = -1$ ,  $V^*(X) = 3.5$ ,  $\text{Error} = -1 - 3.5 = -4.5$ ,  $\eta = 0.1$

$$w_0 = 0.7 + 0.1(-4.5)1 = 0.25$$

$$w_1 = 1 + 0.1(-4.5)1 = 0.55$$

$$w_3 = 0.4 + 0.1(-4.5)2 = -0.5$$

$$w_5 = 1 + 0.1(-4.5)1 = 0.55$$

El nuevo error con este ejemplo es  $-1 - 0.35 = -1.35$  (¿Cuál es el error con el anterior?)



# Ejercicio

---

- Implementar LMS en Python
- Probar con reglin.csv con  $\eta = 0.05$
- Graficar como va cambiando el error para cada  $w$
- Pruebe enstandarizar los datos
- Repita el experimento para  $\eta = 0.1$  y  $1$
- Escriba sus observaciones



# Notas acerca de regresión lineal

---

- Vimos que podemos utilizar la regresión lineal para ajustar funciones no lineales
  - Esto se logra haciendo no lineales los regresores, las variables de entrada, y agregando múltiples de estas
  - El problema es que no siempre sabemos la forma adecuada de la función ni cuantas variables extras incluir. Si incluimos demasiado no generalizaremos bien
  - Una solución interesante es el método de regresión lineal local ponderada (LOWESS).
  - Otra es lo que veremos a continuación



# Tarea

---

- Leer el artículo de Searle, objeción a Turing