Regresión y optimización

Lección 07

Dr. Pablo Alvarado Moya

CE5506 Introducción al reconocimiento de patrones
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

II Semestre, 2019

Contenido

Recapitulando
Regresión lineal

- $(\underline{\mathbf{x}}^{(i)}, y^{(i)})$: i-ésimo dato de entrenamiento
- $h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}^{(i)})$: predicción de hipótesis $h_{\underline{\boldsymbol{\theta}}}$ para dato de entrada $\underline{\mathbf{x}}^{(i)}$

$$h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)} = \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}$$

donde asumimos $x_0 = 1$. El número de características es n.

- Función cuadrática de costo:

$$J(\underline{\boldsymbol{\theta}}) = \frac{1}{2} \sum_{i=1}^{m} \left(h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}^{(i)}) - y^{(i)} \right)^2$$

con m el número de datos en el conjunto de entrenamiento.

- Analíticamente: $\underline{\boldsymbol{\theta}}^* = \arg\min_{\underline{\boldsymbol{\theta}}} J(\underline{\boldsymbol{\theta}}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\underline{\mathbf{y}}$

Función cuadrática de costo

- ¿Por qué usamos una función *cuadrática* de costo?

$$J(\underline{\boldsymbol{\theta}}) = \frac{1}{2} \sum_{i=1}^{m} \left(h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}^{(i)}) - y^{(i)} \right)^2$$

  con m el número de datos en el conjunto de entrenamiento.
- Pudimos usar otras cosas: valor absoluto, o potencia 4...
- Vamos a analizar el problema de regresión desde una perspectiva probabilística, como *posible* argumentación para esto

Regresión lineal
Interpretación probabilística

(1)

- Supongamos que la salida y las entradas siguen el modelo de regresión

$$y^{(i)} = \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}^{(i)} + \epsilon^{(i)} \qquad \Rightarrow \qquad \epsilon^{(i)} = y^{(i)} - \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}$$

- Término de error $\epsilon^{(i)}$ captura:
  - aspectos no modelados (en caso de casas, p. ej. distintos del área)
  - ruido aleatorio
- Supondremos que $\epsilon^{(i)}$ son independientes e identicamente distribuidos (**i.i.d**)

Regresión lineal
Interpretación probabilística

(2)

- Como proceso es complejo, supondremos que $\epsilon^{(i)} \sim \mathcal{N}(0,\sigma^2)$ y por tanto la PDF de $\epsilon^{(i)}$ es

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

- Introduciendo el modelo $\epsilon^{(i)} = y^{(i)} - \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}$ derivamos

$$p(y^{(i)}|\underline{\mathbf{x}}^{(i)};\underline{\boldsymbol{\theta}}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}})^2}{2\sigma^2}\right)$$

- Nótese que $\underline{\boldsymbol{\theta}}$ está fuera de la condición, pues **no** es una variable aleatoria. La asumimos como dada.
- La distribución $y^{(i)}|\underline{\mathbf{x}}^{(i)};\underline{\boldsymbol{\theta}} \sim \mathcal{N}(\underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}},\sigma^2)$

Verosimilitud
Likelihood

- Con la matriz de diseño $\mathbf{X}$ y los parámetros $\underline{\theta}$, la probabilidad **conjunta** de los datos es $p(\underline{\mathbf{y}}|\mathbf{X};\underline{\theta})$
- $p(\underline{\mathbf{y}}|\mathbf{X};\underline{\theta})$ se interpreta como función de los datos para $\underline{\theta}$ constante.
- Cuando queremos interpretar a $p(\underline{\mathbf{y}}|\mathbf{X};\underline{\theta})$ como función de $\underline{\theta}$ la llamamos **verosimilitud** (*likelihood*)

$$L(\underline{\theta}) = L(\underline{\theta};\mathbf{X},\underline{\mathbf{y}}) = p(\underline{\mathbf{y}}|\mathbf{X};\underline{\theta})$$

- Si suponemos que $\epsilon^{(i)}$ son **i.i.d.** (independientes e idénticamente distribuidos), entonces

$$L(\underline{\theta}) = \prod_{i=1}^{m} p(y^{(i)}|\underline{\mathbf{x}}^{(i)};\underline{\theta}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \underline{\theta}^T\underline{\mathbf{x}})^2}{2\sigma^2}\right)$$

Máxima verosimilitud

- El principio de **máxima verosimilitud** (*maximum likelihood*) indica elegir $\underline{\boldsymbol{\theta}}$ de modo que los datos sean lo más probables que se pueda:

$$\underline{\boldsymbol{\theta}}^* = \arg\max_{\underline{\boldsymbol{\theta}}} L(\underline{\boldsymbol{\theta}})$$

- Producto de muchas probabilidades es numéricamente inestable
- Podemos maximizar verosimilitud indirectamente a través de función monotónicamente creciente: en particular se usa el **logaritmo natural**

Verosimilitud logarítimica

- La **verosimilitud logarítmica** (*log likelihood*) es

$$\begin{aligned}\ell(\underline{\boldsymbol{\theta}}) &= \ln L(\underline{\boldsymbol{\theta}}) \\ &= \ln \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}})^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}})^2}{2\sigma^2}\right) \\ &= m \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}\right)^2\end{aligned}$$

- Observe que maximizar $\ell(\underline{\boldsymbol{\theta}})$ ¡es lo mismo que minimizar $J(\underline{\boldsymbol{\theta}})$!
- Con suposiciones probabilísticas: regresión de mínimos cuadrados es equivalente a estimación de máxima verosimilitud de $\underline{\boldsymbol{\theta}}$ (Note irrelevancia de σ)

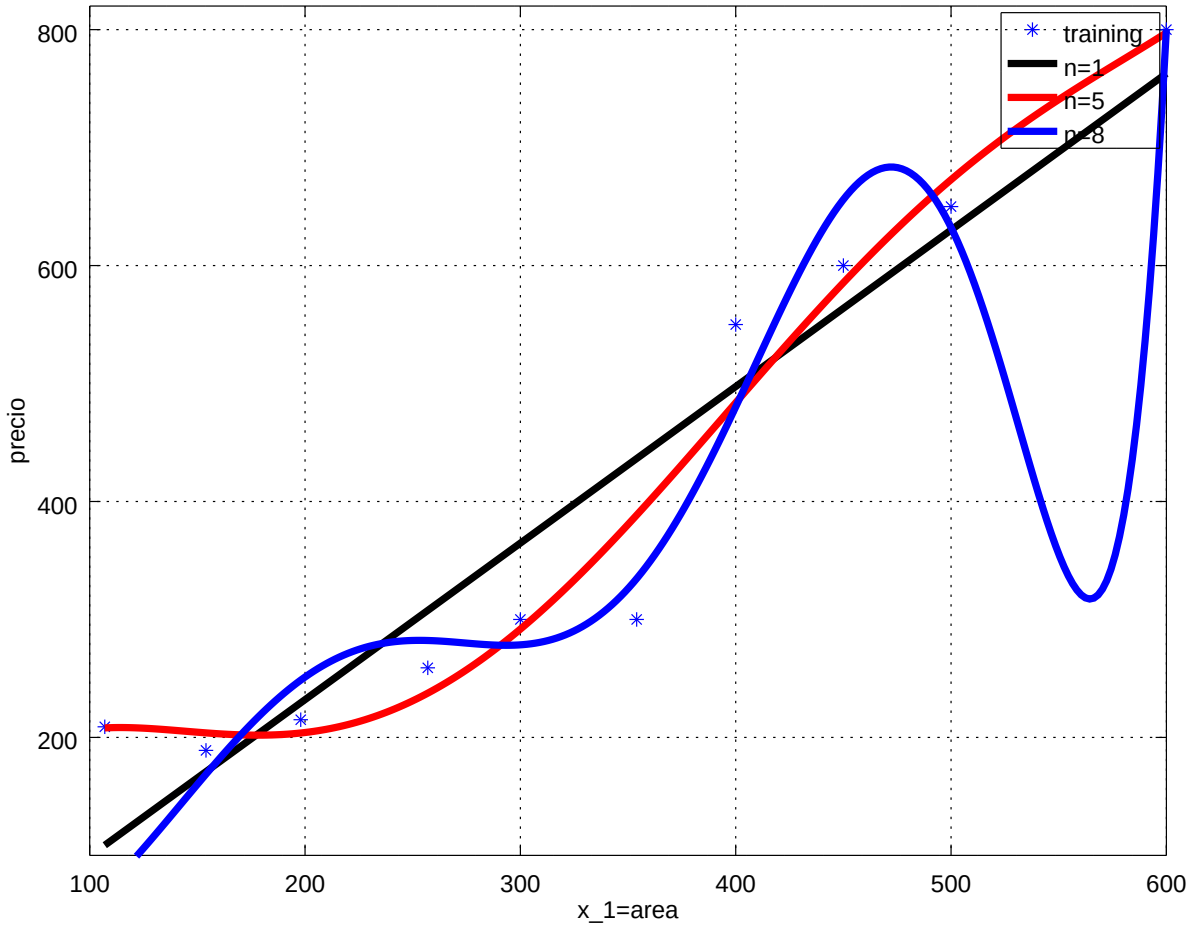
Selección de características

- En ejemplo de precios de casas, tenemos varias **características** (*features*) a disposición:
  - Área habitable
  - Número de pisos
  - Número de habitaciones
- Selección de cuáles características usar es un criterio de diseño
- Es posible introducir características artificiales para introducir no linealidad en un proceso en principio lineal:

$$\underline{\mathbf{x}} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \end{bmatrix}$$

lo que permite modelar aproximaciones de Taylor de n-ésimo orden, de cualquier función no lineal

Ejemplo de regresión con varios órdenes



$$h_{\underline{\theta}}(\underline{\mathbf{x}}) = \underline{\theta}^T \underline{\mathbf{x}} \qquad \underline{\mathbf{x}} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \end{bmatrix}^T$$

Generalización

- Obsérvese que mientras mayor el orden del polinomio, más cerca pasa la curva por los datos de entrenamiento
- Sin embargo, más oscilaciones no plausibles
- Esto está asociado con el concepto de **generalización** que revisaremos más adelante
- Modelos simples: subajustan (*underfitting*)
  - Hay patrones en los datos que el modelo "no ve"
- Modelos complejos sobreajustan (*overfitting*)
  - El modelo se ajusta exactamente al conjunto de entrenamiento particular
- Debe encontrarse el balance
- Selección de características es clave para combatir esos extremos y lograr buena generalización

Algoritmos de aprendizaje paramétricos y no paramétricos

- Un algoritmo de aprendizaje es **paramétrico**, si tiene un número fijo de parámetros que se ajustan para hacer calzar los datos
- El caso de regresión lineal que hemos visto es un caso particular de algoritmo **paramétrico** de aprendizaje, puesto que tamaño de $\underline{\theta}$ se elije a priori.
- En los algoritmos **no paramétricos** el número de parámetros crece con m (tamaño del conjunto de entreniento)
- En otras palabras, en los algoritmos no paramétricos el tamaño del modelo crece linealmente con el conjunto de entrenamiento

Regresión lineal ponderada localmente (1)

Locally weighted linear regression

- **Regresión ponderada localmente**: Primer ejemplo que veremos de método no paramétrico
- Idea es evitar seleccionar un modelo concreto (p.ej. polinomial)
- En regresión lineal (RL):
  1. ajustamos $\underline{\theta}$ para minimizar $J(\underline{\theta})$
  2. predecimos el valor correpondiente a $\underline{\mathbf{x}}$ con $\underline{\theta}^T \underline{\mathbf{x}}$

Regresión lineal ponderada localmente (2)

Locally weighted linear regression

- Idea en LWR (*locally weighted regression*) es seleccionar un vecindario de $\underline{\mathbf{x}}$, y solo en ese vecindario calcular la regresión lineal:
  1. ajustamos $\underline{\boldsymbol{\theta}}$ para minimizar

$$J(\underline{\boldsymbol{\theta}}; \underline{\mathbf{x}}) = \sum_{i=1}^{m} w^{(i)}(\underline{\mathbf{x}}) \left(\underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}^{(i)} - y^{(i)} \right)^2$$

  2. predecimos el valor correpondiente a $\underline{\mathbf{x}}$ con $\underline{\boldsymbol{\theta}}^T \underline{\mathbf{x}}$
- Los pesos $w^{(i)}(\underline{\mathbf{x}})$ son no negativos y permiten ignorar datos lejanos o acentuar datos cercanos a $\underline{\mathbf{x}}$

Regresión lineal ponderada localmente (3)

Locally weighted linear regression

- Con frecuencia se usa el *kernel* gaussiano

$$w^{(i)}(\underline{\mathbf{x}}) = \exp\left(-\frac{(\underline{\mathbf{x}}^{(i)} - \underline{\mathbf{x}})^T \boldsymbol{\Sigma}^{-1} (\underline{\mathbf{x}}^{(i)} - \underline{\mathbf{x}})}{2}\right)$$

- A veces se simplifica además $\boldsymbol{\Sigma} = \tau^2 \mathbf{I}$, con τ el **ancho de banda**, que controla que tan ancho es el vecindario
- En este caso de pesos gaussianos
  - si $|\underline{\mathbf{x}}^{(i)} - \underline{\mathbf{x}}|$ es pequeño, entonces $w^{(i)}(\underline{\mathbf{x}}) \approx 1$
  - si $|\underline{\mathbf{x}}^{(i)} - \underline{\mathbf{x}}|$ es grande, entonces $w^{(i)}(\underline{\mathbf{x}}) \approx 0$
- Note que el método es computacionalmente caro, pues debe recalcular $\underline{\boldsymbol{\theta}}$ cada vez que se quiere predecir con $h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}})$

Técnicas de optimización

Minimización

- Hasta ahora hemos visto **descenso de gradiente** (GD)
- Tuvimos dos variantes del GD: por lotes y estocástico.
- Problemas si no normalizamos el conjunto de entrenamiento $\Rightarrow$ procurar siempre normalizar
- Delicada selección del tamaño de paso (*learning rate*)
- Si la función que queremos normalizar es (aproximadamente) convexa, hay mejores opciones:
  - más rápidas
  - menos evaluaciones de la función a optimizar
- En general, información del gradiente es ventajoso y se usan en ML/PR principalmente este tipo de métodos

Optimización lineal

- Los procesos siguientes requieren optimizar la función en s:

$$J(s) = J(\underline{\boldsymbol{\theta}} - s\underline{\mathbf{d}})$$

  con dirección de minimización $\underline{\mathbf{d}}$ que inicia en $\underline{\boldsymbol{\theta}}$
- Para encontrar $\min_s J(s)$ se utilizan las técnicas de minimización unidimensional
- En particular se utiliza con frecuencia el algoritmo de Brent GNU/octave: función `brent_line_min` del paquete `optim`

Gradiente

- El gradiente $\nabla J(\underline{\boldsymbol{\theta}})$ de una función objetivo indica la dirección de mayor **crecimiento** de la función

$$\nabla J(\underline{\boldsymbol{\theta}}) = \left[\frac{\partial J(\underline{\boldsymbol{\theta}})}{\partial \theta_0} \quad \frac{\partial J(\underline{\boldsymbol{\theta}})}{\partial \theta_1} \quad \cdots \quad \frac{\partial J(\underline{\boldsymbol{\theta}})}{\partial \theta_n} \right]^T$$

  con $\underline{\boldsymbol{\theta}} = [\theta_0, \theta_1, \ldots, \theta_n]^T$

- Extremos ocurren en puntos donde no hay cambio ($\nabla J(\underline{\boldsymbol{\theta}}) = \underline{\mathbf{0}}$)
- El cálculo del gradiente en problemas reales se realiza
  - Analíticamente
  - Por diferenciación numérica
  - Por diferenciación automática

Matriz Hessiana (1)

- Segunda derivada determina tipo de extremo:
  Si $J'(\theta_0) = \left.\frac{dJ(\theta)}{d\theta}\right|_{\theta=\theta_0} = 0$ entonces
  - $J(\theta_0)$ es máximo si $J''(\theta_0) < 0$
  - $J(\theta_0)$ es mínimo si $J''(\theta_0) > 0$
- Equivalente multidimensional de la segunda derivada es la matriz Hessiana (o hessiano):

$$\mathbf{H}(\underline{\boldsymbol{\theta}}) = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1^2} & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_n} \\ \frac{\partial^2 J}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_2^2} & \cdots & \frac{\partial^2 J}{\partial \theta_2 \partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_n \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_n^2} \end{bmatrix}$$

Matriz Hessiana (2)

- Extremos de $J(\underline{\boldsymbol{\theta}})$ se encuentran donde $\nabla J(\underline{\boldsymbol{\theta}}) = 0$ y
  - Si $\mathbf{H}| \succ 0$ entonces $J(\underline{\boldsymbol{\theta}})$ tiene un mínimo local
  - Si $\mathbf{H}| \prec 0$ entonces $J(\underline{\boldsymbol{\theta}})$ tiene un máximo local
  - En otro caso $J(\underline{\boldsymbol{\theta}})$ tiene un punto de silla

Máximo descenso (1)

Steepest descent/ascent

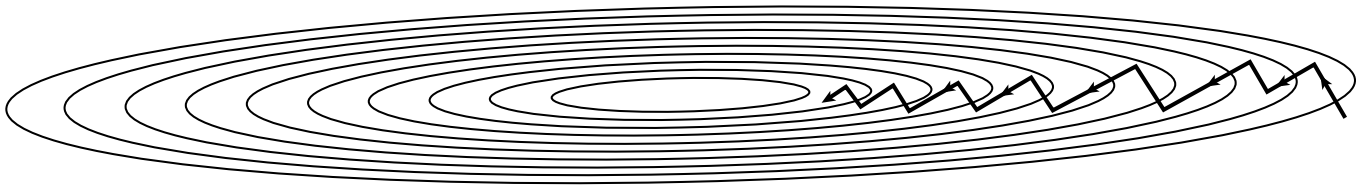
- **Estrategia**: seguir dirección opuesta del gradiente para minimizar

$$\underline{\boldsymbol{\theta}}_{i+1} = \underline{\boldsymbol{\theta}}_i - \nabla J(\underline{\boldsymbol{\theta}}_i) \left(\arg\min_s J(\underline{\boldsymbol{\theta}}_i \pm s \nabla J(\underline{\boldsymbol{\theta}}_i)) \right)$$

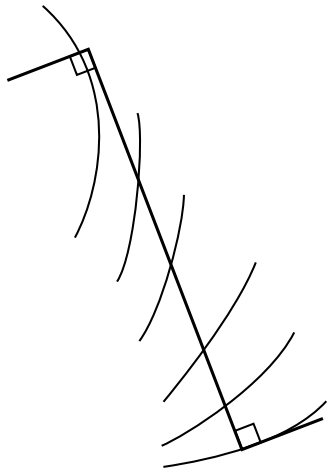
- Requerimos algoritmo de optimización lineal.
- **Problema**: si se minimiza linealmente en dirección del gradiente, nueva dirección del gradiente en mínimo siempre será ortogonal a la última dirección, lo que fuerza un efecto zig-zag:

Máximo descenso (2)

Steepest descent/ascent



(a)



(b)

- Conforme se aproxima el extremo, los desplazamientos lineales disminuyen, lo que hace la convergencia cada vez más lenta

Descenso de gradiente

- El descenso de gradiente que vimos evita el zig-zag:

$$\underline{\boldsymbol{\theta}}_{i+1} = \underline{\boldsymbol{\theta}}_i - \alpha \nabla J(\underline{\boldsymbol{\theta}}_i)$$

  donde la aplicación restringe el valor adecuado de α
- Una mala elección de α conduce a secuencias largas antes de la convergencia (si se elige α muy pequeño), o a oscilaciones indefinidas (si se elige α muy grande)
- Método se usa por facilidad de implementación.

Gradientes conjugados (1)

- Asúmase que la superficie es cuadrática y por tanto

$$J(\underline{\boldsymbol{\theta}}) \approx J(\underline{\boldsymbol{\theta}}_0) + (\underline{\boldsymbol{\theta}} - \underline{\boldsymbol{\theta}}_0)^T \nabla J(\underline{\boldsymbol{\theta}}_0) + \frac{1}{2}(\underline{\boldsymbol{\theta}} - \underline{\boldsymbol{\theta}}_0)^T \mathbf{H}(\underline{\boldsymbol{\theta}}_0)(\underline{\boldsymbol{\theta}} - \underline{\boldsymbol{\theta}}_0)$$

- Esta aproximación de Taylor es más exacta mientras más pequeña sea la vecindad alrededor de $\underline{\boldsymbol{\theta}}_0$
- Bajo esta suposición, la idea es

Direcciones conjugadas

Encontrar direcciones de minimización lineal tales que, la componente del gradiente que el paso anterior ya hizo cero no sea alterada en el nuevo paso

Gradientes conjugados (2)

- Supóngase que ya optimizamos en el paso anterior en la dirección $\underline{\mathbf{d}}_i$, iniciando en $\underline{\boldsymbol{\theta}}_i$, para llegar a un nuevo punto $\underline{\boldsymbol{\theta}}_{i+1}$. Entonces, por estar en un mínimo se debe cumplir

$$\underline{\mathbf{d}}_i^T \nabla J(\underline{\boldsymbol{\theta}}_{i+1}) = 0$$

- Necesitamos que a lo largo de la siguiente dirección $\underline{\mathbf{d}}_{i+1}$ la componente del gradiente paralela a la dirección anterior se mantenga cero:

$$\underline{\mathbf{d}}_i^T \nabla J(\underline{\boldsymbol{\theta}}_{i+1} + \lambda \underline{\mathbf{d}}_{i+1}) = 0$$

- Expandiendo en serie de Taylor con respecto a λ se puede demostrar que esto equivale a

$$\underline{\mathbf{d}}_{i+1}^T \mathbf{H}(\underline{\boldsymbol{\theta}}_{i+1}) \underline{\mathbf{d}}_i = 0$$

Gradientes conjugados (3)

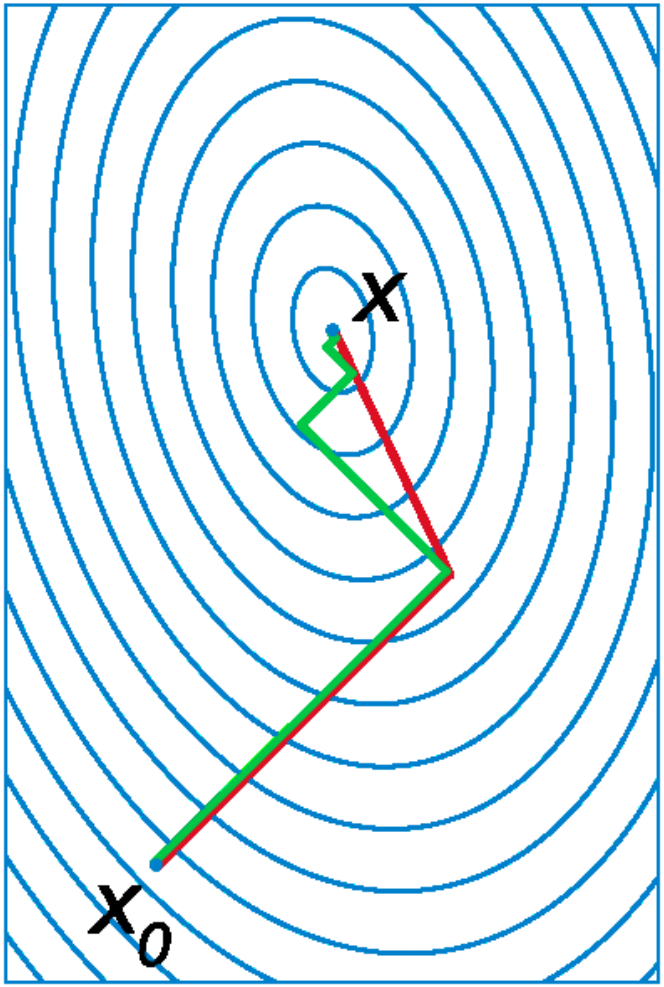
- Direcciones que cumplen esta propiedad se denominan **sin interferencia** o **conjugadas**.
- Polak y Ribiere demostraron que es posible obtener las direcciones conjugadas sin necesidad de calcular la matriz hessiana con:

$$\underline{\mathbf{d}}_{i+1} = \pm\nabla J(\underline{\boldsymbol{\theta}}_{i+1}) + \beta_i \underline{\mathbf{d}}_i$$

$$\beta_i = \frac{(\nabla J(\underline{\boldsymbol{\theta}}_{i+1}) - \nabla J(\underline{\boldsymbol{\theta}}_i))^T \nabla J(\underline{\boldsymbol{\theta}}_{i+1})}{\|\nabla J(\underline{\boldsymbol{\theta}}_i)\|^2}$$

- GNU/octave: función `cg_min` del paquete `optim`

Gradientes conjugados (4)



Otros métodos con gradiente

- Método de Newton

$$\underline{\boldsymbol{\theta}}_{i+1} = \underline{\boldsymbol{\theta}}_i - \mathbf{H}^{-1}(\underline{\boldsymbol{\theta}}_i)\nabla J(\underline{\boldsymbol{\theta}}_i)$$

  - Converge de forma cuadrática cerca del óptimo
  - Requiere cálculo del hessiano e inversión de matriz
- Método de Levenberg-Marquardt
  - Similar al método de Newton, reemplazando el hessiano con

$$\tilde{\mathbf{H}}_i = \mathbf{H}_i + \alpha_i \mathbf{I}$$

    con $\alpha_i \mathbf{I}$ un término de regularización que se adapta durante el algoritmo
  - Al inicio se comporta como algoritmo de máxima inclinación y cerca del óptimo se comporta como algoritmo de Newton

Resumen

1. Regresión lineal
   - Interpretación probabilística
2. Regresión ponderada localmente
   - Selección de características
   - Algoritmo de regresión ponderada localmente
3. Técnicas de optimización
   - Optimización lineal
   - Máximo descenso
   - Gradientes conjugados
   - Otros métodos

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, GNU-Make y Subversion en GNU/Linux