Consideraciones prácticas

Lección 17

Dr. Pablo Alvarado Moya

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

II Semestre, 2019

Contenido

Motivación

- Con herramientas probabilísticas como las revisadas, si dos personas intentan aplicar el mismo algoritmo a los mismos datos, con frecuencia una persona lo va a poder configurar bien y otra no.
- Objetivo ahora es enfocar la atención a los puntos claves para hacer funcionar los algoritmos.
- Estos son reglas empíricas y por tanto debatibles
- Algunas reglas son de aplicación y no tendrán sentido en investigación

Ideas clave

1. Diagnósticos para depurar algoritmos de aprendizaje
2. Análisis de error y análisis ablativo
3. Cómo iniciar con problemas de aprendizaje automático
   - Optimización (estadística) prematura

Depuración de algoritmos de aprendizaje

Caso de estudio

- Supongamos que queremos construir un filtro anti-spam
- Usted eligió un conjunto pequeño de 100 palabras como características, en vez de usar 50 000 palabras
- Usando regresión logística Bayesiana con descenso de gradiente, obtuvo un 20 % de error de prueba (muy alto)

$$\max_{\underline{\boldsymbol{\theta}}} \sum_{i=1}^{m} \log p(y^{(i)} | \underline{\mathbf{x}}^{(i)}, \underline{\boldsymbol{\theta}}) - \lambda \|\underline{\boldsymbol{\theta}}\|^2$$

- ¿Qué sigue?

Posibles mejoras

- Mejorar el algoritmo de distintas formas:
  1. Buscar más datos de entrenamiento
  2. Reducir el conjunto de características
  3. Probar usando más características
  4. Probar nuevas características "más poderosas"
  5. Usar más iteraciones en el descenso de gradiente
  6. Usar otros métodos de optimización (Newton, gradientes conjugados)
  7. Usar otro parámetro de regularización λ
  8. Probar SVM
- Probar arbitrariamente consume tiempo
- Es un asunto de suerte llegar a corregir el problema
- Cada una de las mejoras ataca distintos problemas
- Podemos diagnosticar problemas sistemáticamente y dar solución eficientemente

Estrategia recomendada

- ¡No busque mejorar algoritmo aleatoriamente!
- Mejor estrategia:
  1. Ejecute diagnósticos para detectar el problema
  2. Corrija el problema

Ejemplo de diagnóstico

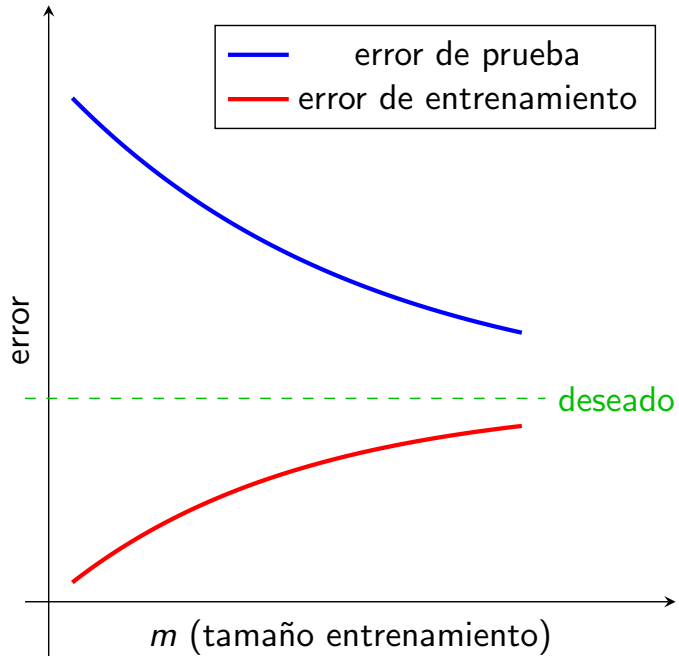
- El error de prueba de la regresión logística Bayesiana de 20 % es excesivamente alto.
- Supongamos que usted sospecha:
  1. Sobreajuste (alta varianza)
  2. Muy pocas características para clasificar spam (alto sesgo)
- ¿Cómo detectamos si es alta varianza o alto sesgo?

Ejemplo de diagnóstico

- El error de prueba de la regresión logística Bayesiana de 20 % es excesivamente alto.
- Supongamos que usted sospecha:
  1. Sobreajuste (alta varianza)
  2. Muy pocas características para clasificar spam (alto sesgo)
- ¿Cómo detectamos si es alta varianza o alto sesgo?
- Alta varianza: error de entrenamiento mucho más bajo que error de prueba
- Alto sesgo: error de entrenamiento también será alto

Diagnóstico de alta varianza

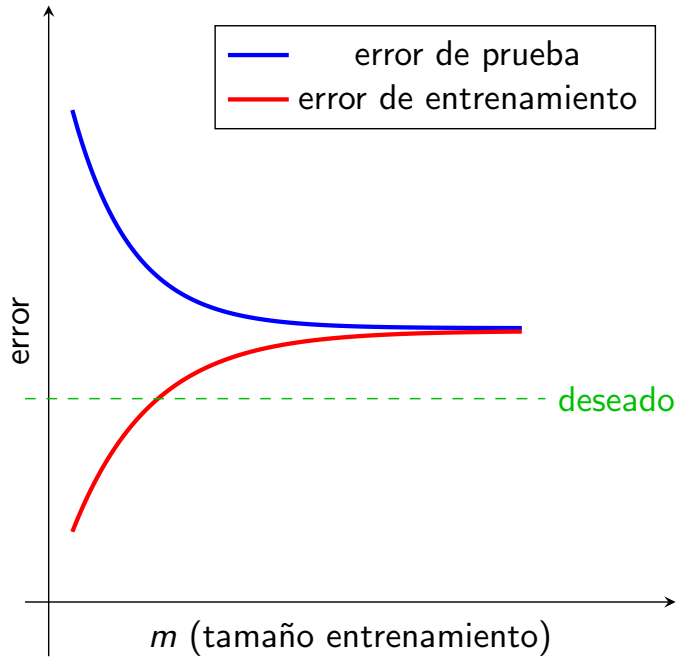
- Comportamiento típico de aprendizaje con alta varianza



- El error de prueba baja conforme m crece
- El error de entrenamiento sube conforme m crece
- Esto sugiere que el aumentar m ayuda
- Característica es la brecha fuerte entre ambos tipos de error

Diagnóstico de alto sesgo

- Comportamiento típico de aprendizaje con alto sesgo



- Curva del error de entrenamiento se estanca en valor alto
- Brecha entre ambos tipos de error es baja

Tipos de corrección

Corrige:

1. Más datos de entrenamiento
2. Reducir # de características
3. Más características
4. Características "más poderosas"
5. Más iteraciones en optimizador
6. Otros métodos de optimización
7. Parámetro de regularización λ
8. Usar otro clasificador

Tipos de corrección

	Corrige:
1. Más datos de entrenamiento	alta varianza
2. Reducir # de características	
3. Más características	
4. Características "más poderosas"	
5. Más iteraciones en optimizador	
6. Otros métodos de optimización	
7. Parámetro de regularización λ	
8. Usar otro clasificador	

Tipos de corrección

	Corrige:
1. Más datos de entrenamiento	alta varianza
2. Reducir # de características	alta varianza
3. Más características	
4. Características "más poderosas"	
5. Más iteraciones en optimizador	
6. Otros métodos de optimización	
7. Parámetro de regularización λ	
8. Usar otro clasificador	

Tipos de corrección

	Corrige:
1. Más datos de entrenamiento	alta varianza
2. Reducir # de características	alta varianza
3. Más características	alto sesgo
4. Características "más poderosas"	
5. Más iteraciones en optimizador	
6. Otros métodos de optimización	
7. Parámetro de regularización λ	
8. Usar otro clasificador	

Tipos de corrección

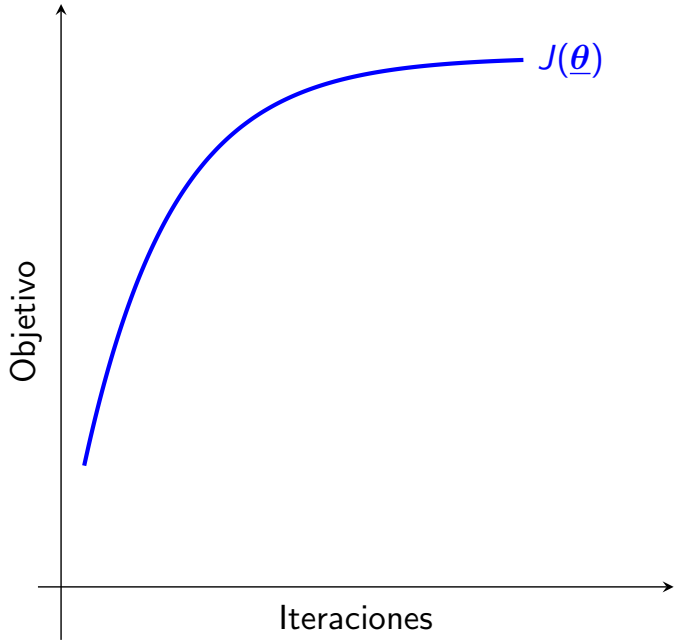
	Corrige:
1. Más datos de entrenamiento	alta varianza
2. Reducir # de características	alta varianza
3. Más características	alto sesgo
4. Características "más poderosas"	alto sesgo
5. Más iteraciones en optimizador	
6. Otros métodos de optimización	
7. Parámetro de regularización λ	
8. Usar otro clasificador	

Otros diagnósticos

- El diagnóstico de sesgo contra varianza es bastante común
- Otros problemas requieren de ingenio para contruir diagnósticos a la medida que detecten el problema
- Otro ejemplo:
  - Regresión logística Bayesiana tiene 2 % de error en spam y 2 % de error en no-spam (error muy alto para no spam)
  - SVM con kernel lineal tiene un 10 % de error en spam y un 0,01 % de error en no-spam (desempeño aceptable)
  - Usted quiere usar regresión logística por ser más eficiente
- ¿Qué se puede hacer?

Otras preguntas:

- ¿El algoritmo de optimización está convergiendo?



Otras preguntas:

- ¿El algoritmo de optimización está convergiendo?
- ¿Estamos optimizando la función correcta?

Otras preguntas:

- ¿El algoritmo de optimización está convergiendo?
- ¿Estamos optimizando la función correcta?
- En regresión logística Bayesiana, ¿Es el regularizador λ correcto?

$$\max_{\underline{\boldsymbol{\theta}}} J(\underline{\boldsymbol{\theta}}) = \max_{\underline{\boldsymbol{\theta}}} \sum_{i=1}^{m} \ln p(y^{(i)}|\underline{\mathbf{x}}^{(i)},\underline{\boldsymbol{\theta}}) - \lambda \|\underline{\boldsymbol{\theta}}\|^2$$

Otras preguntas:

- ¿El algoritmo de optimización está convergiendo?
- ¿Estamos optimizando la función correcta?
- En regresión logística Bayesiana, ¿Es el regularizador λ correcto?
- En SVM ¿usamos el C correcto?

$$\min_{\underline{\mathbf{w}},b} \|\underline{\mathbf{w}}\|^2 + C \sum_{i=1}^{m} \xi_i$$

$$\text{sujeto a } y^{(i)}(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(i)} + b) \geq 1 - \xi_i$$

Diagnóstico (1)

- En el caso que estamos estudiando: SVM es mejor que regresión logística Bayesiana (BLR)
- Queremos usar la regresión logística Bayesiana
- Sea $\underline{\boldsymbol{\theta}}_{SVM}$ los parámetros que aprendió SVM
- Sea $\underline{\boldsymbol{\theta}}_{BLR}$ los parámetros que aprendió BLR
- Nos interésa la exactitud ponderada (*weighted accuracy*):

$$a(\underline{\boldsymbol{\theta}}) = \max_{\underline{\boldsymbol{\theta}}} \sum_i \underline{\mathbf{w}}^{(i)} 1\left\{ h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}^{(i)}) = y^{(i)} \right\}$$

- Como $\underline{\boldsymbol{\theta}}_{SVM}$ es mejor que $\underline{\boldsymbol{\theta}}_{BLR}$ entonces $a(\underline{\boldsymbol{\theta}}_{SVM}) > a(\underline{\boldsymbol{\theta}}_{BLR})$

Diagnóstico (2)

- Como SVM y BLR optimizan funciones distintas, definamos una medida objetivo común a maximizar basada en BLR, por ejemplo:

$$\bar{J}(\underline{\theta}) = \sum_{i=1}^{m} \ln p(y^{(i)}|\underline{\mathbf{x}}^{(i)}, \underline{\theta})$$

- (encontrar un planteo común $\bar{J}(\underline{\theta})$ a ambos métodos es complejo)

Diagnóstico: función objetivo u optimización

- Caso 1:
$$a(\underline{\theta}_{SVM}) > a(\underline{\theta}_{BLR})$$
$$\bar{J}(\underline{\theta}_{SVM}) > \bar{J}(\underline{\theta}_{BLR})$$
Como BLR maximiza $\bar{J}(\underline{\theta})$, entonces problema en algoritmo de optimización, que no pudo encontrar un buen $\underline{\theta}_{BLR}$
- Caso 2:
$$a(\underline{\theta}_{SVM}) > a(\underline{\theta}_{BLR})$$
$$\bar{J}(\underline{\theta}_{SVM}) \leq \bar{J}(\underline{\theta}_{BLR})$$
El algoritmo de optimización encontró un buen valor de $\bar{J}(\underline{\theta})$ pero no parece ser la mejor opción, así que el problema es la función objetivo que no refleja el verdadero objetivo a optimizar

Tipos de corrección

	Corrige:
1. Más datos de entrenamiento	alta varianza
2. Reducir # de características	alta varianza
3. Más características	alto sesgo
4. Características "más poderosas"	alto sesgo
5. Más iteraciones en optimizador	
6. Otros métodos de optimización	
7. Parámetro de regularización λ	
8. Usar otro clasificador	

Tipos de corrección

	Corrige:
1 Más datos de entrenamiento	alta varianza
2 Reducir # de características	alta varianza
3 Más características	alto sesgo
4 Características "más poderosas"	alto sesgo
5 Más iteraciones en optimizador	algoritmo de optimización
6 Otros métodos de optimización	algoritmo de optimización
7 Parámetro de regularización λ	función objetivo
8 Usar otro clasificador	función objetivo

Más diagnósticos

- Con frecuencia es necesario idear diagnósticos propios para encontrar problemas
- Aun si un algoritmo de aprendizaje funciona bien, es importante correr diagnósticos para
  1. Para comprender el problema de aplicación.
     Si se trabaja en un problema de aprendizaje automático por varios meses/años, es valioso comprender qué funciona y qué no
  2. Para publicar (tesis, artículos, patentes, etc.)
     Diagnósticos y análisis de errores revelan detalles del problema para justificar resultados científicos
  3. Para poder argumentar ante usuarios/clientes
     Es importante las razones por las cuales un algoritmo seleccionado funciona para poder justificar su uso
- Una de las buenas prácticas es el análisis de error: cuáles son las fuentes de error

Análisis de error

- Sistemas reales compuestos por varios subsistemas
- Reconocimiento de caras:
  1. Captura de imágenes
  2. Preprocesamiento (fondo)
  3. Detección de caras
  4. Segmentación de ojos
  5. Segmentación de nariz
  6. Segmentación de boca
  7. Clasificador

Análisis de error

- Sistemas reales compuestos por varios subsistemas
- Reconocimiento de caras: (total 85 %)
  1. Captura de imágenes
  2. Preprocesamiento (fondo) 85,1 %
  3. Detección de caras 91 %
  4. Segmentación de ojos 95 %
  5. Segmentación de nariz 96 %
  6. Segmentación de boca 97 %
  7. Clasificador 100 %
- ¿Cuánto error es atribuible a cada sección?
  (inserte datos "perfectos" para evaluar cada componente)

Análisis ablativo

- Análisis de error: intenta explicar diferencia entre desempeño actual y desempeño perfecto
- Análisis ablativo: intenta explicar diferencia entre una desempeño de una línea base (mala) y desempeño actual
- Ejemplo: usted contruye un filtro anti-spam
- Con solo regresión logística se alcanza un 94 % de error
- Con varias características buenas y regresión logística 99,9 %
  1. Corrección ortográfica
  2. Características del emisor
  3. Características del encabezado
  4. Características de parser del correo
  5. Parser del Javascript
  6. Características de imágenes embebidas
- ¿Cuánto ayuda cada característica de esas?

Análisis ablativo

- Elimine componentes uno a la vez para observar deterioro
  1. Sistema completo
  2. Corrección ortográfica
  3. Características del emisor
  4. Características del encabezado
  5. Car. parser del texto
  6. Parser del Javascript
  7. Car. imágenes embebidas

Análisis ablativo

- Elimine componentes uno a la vez para observar deterioro
  1. Sistema completo 99,9 %
  2. Corrección ortográfica 99,0 %
  3. Características del emisor 98,9 %
  4. Características del encabezado 98,9 %
  5. Car. parser del texto 95 %
  6. Parser del Javascript 94,5 %
  7. Car. imágenes embebidas 94 %

Análisis ablativo

- Elimine componentes uno a la vez para observar deterioro
  1. Sistema completo 99,9 %
  2. Corrección ortográfica 99,0 %
  3. Características del emisor 98,9 %
  4. Características del encabezado 98,9 %
  5. Car. parser del texto 95 %
  6. Parser del Javascript 94,5 %
  7. Car. imágenes embebidas 94 %
- Conclusión: el parser del texto del correo brinda mayor mejora

Estrategias de diseño

¿Cómo comenzar con un problema?

Hay dos estrategias fundamentalmente:

1. Diseño cuidadoso
   - Invierta tiempo diseñando las características adecuadas, recolectando datos apropiados y diseñando la arquitectura algorítmica apropiada
   - Implemente y ojalá todo funcione
   - Ventaja: Algoritmos bien diseñados. Posíblemente se encuentre un algoritmo nuevo elegante que aporte a la investigación
2. Construya rápido y ajuste
   - Implemente algo rápido (*quick-and-dirty*)
   - Aplique análisis de error y diagnósticos para corregir problemas
   - Ventaja: Usualmente lo lleva a una aplicación funcional más rápido

Optimización estadística prematura

- Con frecuencia no se sabe qué módulos del sistema son críticos
- La única forma de verificar eso es implementando todo y luego optimizar
- (no aplica si el interés es investigación)
- ¡Evite la optimización (estadística) prematura!

Resumen

- Inversión de tiempo en diagnósiticos es siempre buena
- Usualmente se deja a su ingenio encontrar los diagnósiticos adecuados
- Análisis ablativo y de error permiten identificar módulos críticos
- Hay dos estrategias para aplicar los algoritmos de aprendizaje:
  1. Diseño cuidadoso y luego implementación (riesgo de optimización prematura)
  2. Prototipo rápido, diagnósitico y ajuste

Resumen

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