Regresi Logistik

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

aliakbars@live.com

April 5, 2020

Selayang Pandang

Bahan Bacaan

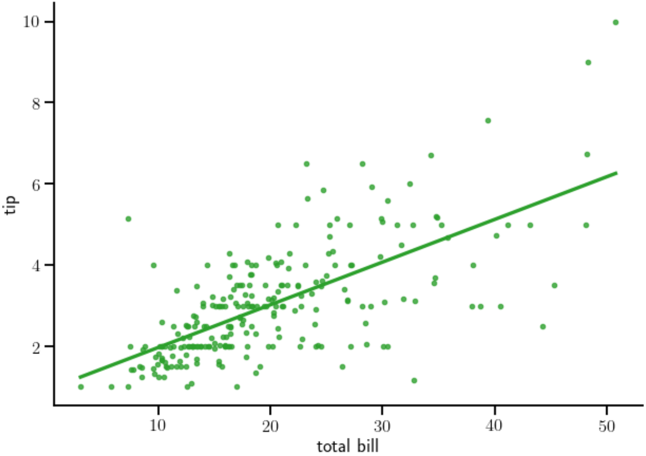
1. Septiandri, A.A. (2019). Artificial Intelligence Kuliah 3: Regresi Logistik. Github.
2. Murray, I. (2016). MLPR class notes. (Regression and Gradients; Logistic Regression) http://www.inf.ed.ac.uk/teaching/courses/mlpr/2016/notes/ (graduate level)

ULASAN

Minggu lalu...

- Generalisasi error
- Bias-variance trade off
- Optimasi model: pembagian dataset
- Metrik evaluasi

Regresi Linear



Gambar: Mencari hubungan $\mathbf{y} = X\mathbf{w}$ dengan meminimalkan $E(\mathbf{w}) = \sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2$

Memprediksi kategori

- Apa yang harus dilakukan jika kita ingin memprediksi *kategori* alih-alih *nilai riil*?

Memprediksi kategori

- Apa yang harus dilakukan jika kita ingin memprediksi *kategori* alih-alih *nilai riil*?
- Contoh: Prediksi apakah komentar-komentar berikut termasuk *spam* atau *ham* (bukan spam) jika dilihat dari kemunculan kata-kata ‘order’ dan ‘password‘.

Memprediksi kategori

- Apa yang harus dilakukan jika kita ingin memprediksi *kategori* alih-alih *nilai riil*?
- Contoh: Prediksi apakah komentar-komentar berikut termasuk *spam* atau *ham* (bukan spam) jika dilihat dari kemunculan kata-kata ‘order’ dan ‘password‘.
- Kita asumsikan spam $= 1$ dan ham $= 0$. Bagaimana memaksa keluaran dari regresi linear $y \in (-\infty, \infty)$ menjadi $y \in \{0, 1\}$?

Regresi Logistik

Mengubah Keluaran

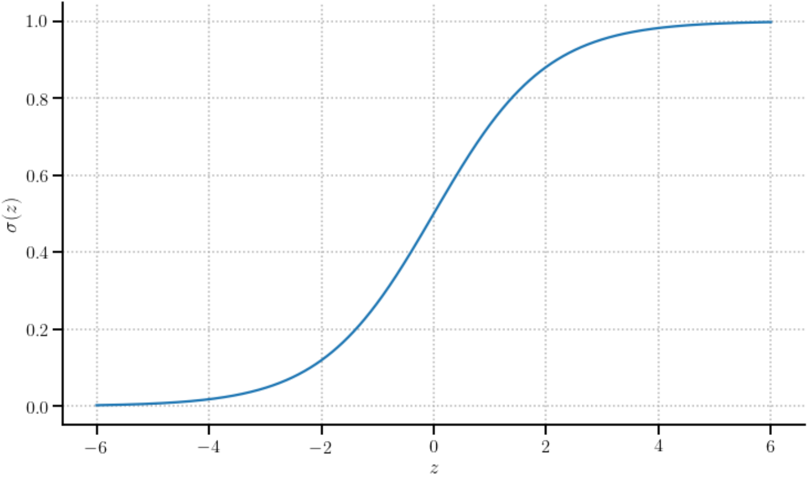
- Berdasarkan keluaran regresi linear, kita bisa memaksanya menjadi [0, 1]
- Gunakan fungsi sigmoid:

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

- Nilai [0, 1] dapat diartikan sebagai probabilitas dari kelas
- Karena probabilitas harus memiliki total 1, maka

$$P(y = 0|\mathbf{x}) = 1 - P(y = 1|\mathbf{x})$$

Fungsi Sigmoid



Gambar: Fungsi sigmoid/logistik $\sigma(z) = \frac{1}{1+exp(-z)}$

Decision Boundary

- Dalam kasus satu variabel prediktor, kemiringan dari batas keputusan diatur oleh nilai w_1, sedangkan w_0 (*intercept*) hanya menggesernya

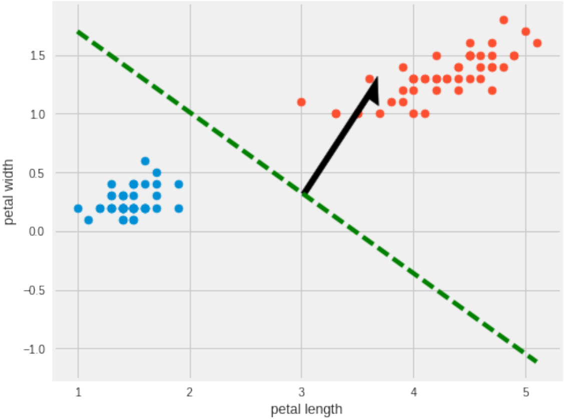
Decision Boundary

- Dalam kasus satu variabel prediktor, kemiringan dari batas keputusan diatur oleh nilai w_1, sedangkan w_0 (*intercept*) hanya menggesernya
- Batas keputusan yang dihasilkan akan berupa *hyperplane* yang akan tegak lurus terhadap vektor $\mathbf{w}$

Decision Boundary

- Dalam kasus satu variabel prediktor, kemiringan dari batas keputusan diatur oleh nilai w_1, sedangkan w_0 (*intercept*) hanya menggesernya
- Batas keputusan yang dihasilkan akan berupa *hyperplane* yang akan tegak lurus terhadap vektor $\mathbf{w}$
- Dari $\mathbf{w}$, kita bisa menggambarkan batas keputusan (*decision boundary*) ketika $p(y = 1|\mathbf{x}) = p(y = 0|\mathbf{x}) = 0.5$, i.e. $\mathbf{w}^T\mathbf{x} = 0$

Decision Boundary



Gambar: Batas keputusan dan vektor bobot untuk klasifikasi dua kelas

Bagaimana cara mencari nilai $\mathbf{w}$?

Likelihood

- Asumsi i.i.d.
- Dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$
- *Likelihood*-nya menjadi

$$
\begin{aligned}
p(\mathcal{D}|\mathbf{w}) &= \prod_{i=1}^{N} p(y = y_i|\mathbf{x}_i, \mathbf{w}) \\
&= \prod_{i=1}^{N} p(y = 1|\mathbf{x}_i, \mathbf{w})^{y_i} (1 - p(y = 1|\mathbf{x}_i, \mathbf{w}))^{1-y_i}
\end{aligned}
$$

- *Log likelihood* $L(\mathbf{w}) = log\ p(\mathcal{D}|\mathbf{w})$

$$
L(\mathbf{w}) = \sum_{i=1}^{N} y_i log\ \sigma(\mathbf{w}^T\mathbf{x}_i) + (1 - y_i) log(1 - \sigma(\mathbf{w}^T\mathbf{x}_i))
$$

Solusi

- Nilai optimum untuk kasus ini unik, i.e. *convex*
- Untuk memaksimalkan nilainya, gunakan gradien

$$\frac{\partial L}{\partial w_j} = \sum_{i=1}^{N} (y_i - \sigma(\mathbf{w}^T \mathbf{x}_i)) x_{ij}$$

- Tidak ada solusi tertutup sehingga harus menggunakan *optimasi numerik*, e.g. dengan *gradient descent*

Optimasi

Mengapa dinamakan *machine learning*?

Alasan Melakukan Optimasi

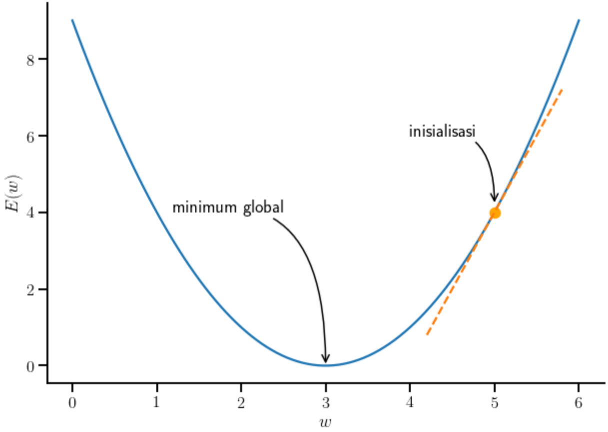
- Belajar $\rightarrow$ masalah optimasi kontinu
- Contoh: regresi linear, regresi logistik, jaringan saraf tiruan, SVM
- Salah satu caranya adalah dengan *maximum likelihood*

"Berapa peluangnya kita melihat data ini jika diketahui parameternya?"

Cara Melakukan Optimasi

- Menggunakan fungsi galat/error $E(\mathbf{w})$ yang akan diminimalkan
- e.g. dapat berupa $-L(\mathbf{w})$
- Beda nilai $\mathbf{w}$, beda besar error
- Belajar $\equiv$ menuruni permukaan error

Menuruni Permukaan Fungsi Error



Gambar: Menuruni lembah fungsi error E(w)

Gradient Descent

begin
- Inisialisasi $\mathbf{w}$
- **while** $E(\mathbf{w})$ *masih terlalu besar* **do**
  - Hitung $\mathbf{g} \leftarrow \frac{\partial E}{\partial \mathbf{w}}$
  - $\mathbf{w} \leftarrow \mathbf{w} - \eta \mathbf{g}$
- **end**
- return $\mathbf{w}$

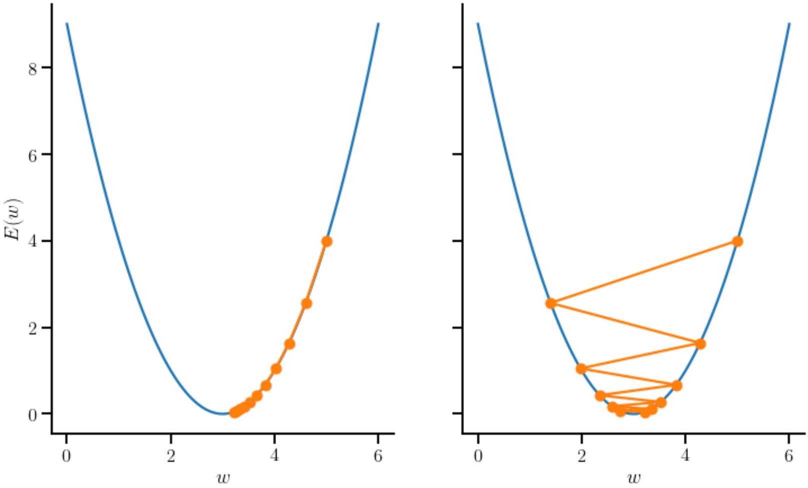
end

Algorithm 1: Melatih dengan gradient descent

Learning Rate

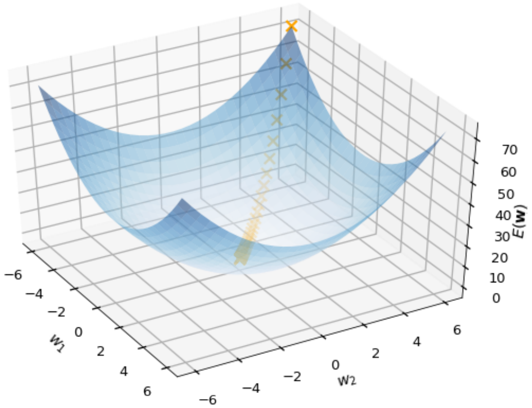
- η (baca: "eta") dikenal sebagai *step size* atau *learning rate* dengan nilai $\eta > 0$
- η terlalu kecil $\rightarrow$ lambat
- η terlalu besar $\rightarrow$ tidak stabil

Pengaruh Nilai η



Gambar: Fungsi $E(w) = w^2 - 6w + 9$, kiri: $\eta = 0.1$, kanan: $\eta = 0.9$

Menuruni Permukaan Fungsi Error



Gambar: Menuruni lembah fungsi error

Batch vs Online

- Untuk data yang sedikit, kita bisa menjumlahkan semua error sebelum memperbarui nilai $\mathbf{w}$ (*batch*)

Batch vs Online

- Untuk data yang sedikit, kita bisa menjumlahkan semua error sebelum memperbarui nilai $\mathbf{w}$ (*batch*)
- Bagaimana untuk 10 juta data?

Batch vs Online

- Untuk data yang sedikit, kita bisa menjumlahkan semua error sebelum memperbarui nilai $\mathbf{w}$ (*batch*)
- Bagaimana untuk 10 juta data?
- Ternyata, kita bisa memperbarui nilai $\mathbf{w}$ untuk setiap satu data (*online*)

Gradient Descent (Batch)

begin
  Inisialisasi $\mathbf{w}$
  while $E(\mathbf{w})$ *masih terlalu besar* **do**
    Hitung $\mathbf{g} \leftarrow \sum_{i=1}^{N} \frac{\partial E_i}{\partial \mathbf{w}}$
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \mathbf{g}$
  end
  return $\mathbf{w}$
end

Algorithm 2: Melatih dengan batch gradient descent

Stochastic Gradient Descent

begin

  Inisialisasi $\mathbf{w}$

  while $E(\mathbf{w})$ *masih terlalu besar* **do**

    Pilih j sebagai integer acak antara 1..N

    Hitung $\mathbf{g} \leftarrow \frac{\partial E_j}{\partial \mathbf{w}}$

    $\mathbf{w} \leftarrow \mathbf{w} - \eta \mathbf{g}$

  end

  return $\mathbf{w}$

end

Algorithm 3: Stochastic gradient descent (SGD)

Kelebihan dan Kekurangan

- **Batch** lebih *powerful*
- **Batch** lebih mudah dianalisis
- **Online** lebih praktikal untuk data yang besar

Pengembangan Gradient Descent (non-examinable)

- "Why **Momentum** Really Works" [Goh, 2017]
- **Performance-dependent** η, e.g. "NewBOB": η berubah menjadi setengahnya saat validation set tidak menjadi lebih baik
- **Time-dependent schedules**, e.g. eksponensial: $\eta(t) = \eta(0) exp(-t/r)$ ($r \sim$ ukuran data latih)

Regresi linear dengan gradient descent

https://github.com/aliakbars/uai/blob/gh-pages/images/line.gif

Tentang Metode Optimasi

- Masih banyak metode optimasi yang tidak dibahas, e.g. linear programming, Newton's method, dll.
- Optimasi merupakan bidang matematika yang kompleks
- Masalah convex: optimum global. Non-convex: optimum lokal.
- Pahami mengapa *gradient descent* bisa mengalami masalah

Klasifikasi

Model Generatif dan Diskriminatif

- Naïve Bayes memodelkan bagaimana kelas "menghasilkan" vektor fitur $p(\mathbf{x}|y)$ untuk kemudian diklasifikasikan dengan

$$p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$$

Model Generatif dan Diskriminatif

- Naïve Bayes memodelkan bagaimana kelas “menghasilkan” vektor fitur $p(\mathbf{x}|y)$ untuk kemudian diklasifikasikan dengan

$$p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$$

- Regresi logistik langsung memodelkan $p(y|\mathbf{x})$, i.e. diskriminatif

Model Generatif dan Diskriminatif

- Naïve Bayes memodelkan bagaimana kelas "menghasilkan" vektor fitur $p(\mathbf{x}|y)$ untuk kemudian diklasifikasikan dengan

$$p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$$

- Regresi logistik langsung memodelkan $p(y|\mathbf{x})$, i.e. diskriminatif
- Keuntungan metode diskriminatif: Buat apa memodelkan $p(\mathbf{x})$? Kita selalu punya input.

Model Generatif dan Diskriminatif

- Naïve Bayes memodelkan bagaimana kelas "menghasilkan" vektor fitur $p(\mathbf{x}|y)$ untuk kemudian diklasifikasikan dengan

$$p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$$

- Regresi logistik langsung memodelkan $p(y|\mathbf{x})$, i.e. diskriminatif
- Keuntungan metode diskriminatif: Buat apa memodelkan $p(\mathbf{x})$? Kita selalu punya input.
- Keuntungan metode generatif: Bisa menangani kasus data yang hilang, mendeteksi pencilan, atau mungkin *memang* perlu menghasilkan input

Klasifikasi Multikelas

- Buat vektor bobot $\mathbf{w}_k$ untuk setiap kelas, untuk mengklasifikasikan k dan bukan-k
- Gunakan fungsi *softmax*

$$p(y = k|\mathbf{x}) = \frac{exp(\mathbf{w}_k^T\mathbf{x})}{\sum_{j=1}^{C} exp(\mathbf{w}_j^T\mathbf{x})}$$

- Perhatikan bahwa $0 \leq p(y = k|\mathbf{x}) \leq 1$ dan $\sum_{j=1}^{C} p(y = j|\mathbf{x}) = 1$

Ikhtisar

- Klasifikasi dengan regresi logistik dan *gradient descent*
- Menggunakan $-L(\mathbf{w})$ sebagai pengganti $E(\mathbf{w})$
- Model generatif vs diskriminatif

Referensi

Gabriel Goh (2017)
"Why Momentum Really Works"
Distill http://distill.pub/2017/momentum/

Terima kasih