Greedy Sparse Least-Squares SVM

Сон Артём

11 апреля 2024 г.

Постановка задачи

- Пусть дано множество тренировочных данных:
  $\mathcal{D} = (x_i, y_i)_{i=1}^{l}, x_i \in \mathcal{X} \subset \mathbb{R}^d, y_i \in \mathcal{Y} \subset \mathbb{R}$
- Необходимо найти функцию $f(x)$, которая наилучшим образом предсказывает новые наблюдения.
- Модель нелинейной регрессии с квадратичной функцией потерь задается следующим образом:
  - $\mathcal{K}(x, x') = \phi(x) \cdot \phi(x')$, - ядерная функция
    - $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
    - $\phi : \mathcal{X} \to \mathcal{F}$, $\mathcal{F}$ - пространство признаков высшей размерности
  - $W_{LS-SVM}(w, b) = \frac{1}{2}||w||^2 + \frac{\gamma}{l}\sum_{i=1}^{l}(y_i - w \cdot \phi(x_i) + b)^2$ - целевая функция модели, $w = \sum_{i=1}^{l} \alpha_i \phi(x_i)$
  - $f(x) = \sum_{i=1}^{l} \alpha_i \mathcal{K}(x_i, x) + b$ - решающая функция, полученная минимизацией целевой функции.

Идея алгоритма

- Пусть даны тренировочные данные:
  $\mathcal{D} = (x_i, y_i)_{i=1}^{l}, x_i \in \mathcal{X} \subset \mathbb{R}^d, y_i \in \mathcal{Y} \subset \mathbb{R}$
- Дана модель нелинейной регресси с квадратичной функции потерь, с ядерной функцией $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, определяющую коэффициенты $\{b, \alpha_i\}_{i=1}^{l} \subset \mathbb{R}$ решающей функции $f(x) = \sum_{i=1}^{l} \alpha_i \mathcal{K}(x_i, x) + b$, минимизируя целевую функцию модели

Мы хотим найти такое приближение, что для некоторого подмножества $S \subset \{1, \ \dots, l\}$, коэффициенты $\{b, \beta_i\}_{i=1}^{l} \subset \mathbb{R}$ будут минимизировать функцию цели $f(x) = \sum_{i \in S} \beta_i \mathcal{K}(x_i, x) + b$ нашей GSLS-SVM модели с параметром регуляризации γ:

$$\mathcal{L}(\beta, b) = \frac{1}{2} \sum_{i,j \in S} \beta_i \beta_j \mathcal{K}(x_i, x_j) + \frac{\gamma}{l} \sum_{i=1}^{l} (y_i - \sum_{j \in S} \beta_j \mathcal{K}(x_i, x_j))^2$$

Алгоритм. Жадность

- Каждую итерацию GSLS выбирается новый вектор из датасета в качестве опорного
- Вычисляется целевая функция, и лучший на данной итерации опорный вектор добавляется к результирующему подмножеству.
- Процесс завершается, когда в подмножество достигло некоторого предопределенного размера.

Алгоритм. Вычисления (1)

$$\mathcal{L}(\beta, b) = \frac{1}{2}\sum_{i,j\in S}\beta_i\beta_j\mathcal{K}(x_i, x_j) + \frac{\gamma}{l}\sum_{i=1}^{l}(y_i - \sum_{j\in S}\beta_j\mathcal{K}(x_i, x_j))^2$$

Если приравнять частные производные по β и b нулю, и разделить на $2\gamma/l$, получим:

$$\sum_{i\in S}\beta_i\sum_{j=1}^{l}k_{ij} + lb = \sum_{j=1}^{l}y_j$$

и

$$\sum_{i\in S}\beta_i\{\frac{l}{2\gamma}k_{ir} + \sum_{j=1}^{l}k_{jr}k_{ji}\} + b\sum_{i=1}^{l}k_{ir} = \sum_{i=1}^{l}y_ik_{ir} \quad \forall r\in S$$

Алгоритм. Вычисления (2)

Эти уравнения представляют из себя СЛАУ с $|S|+1$ уравнениями и неизвестными. В матричной форме:

$$H \begin{bmatrix} \beta \\ b \end{bmatrix} = \begin{bmatrix} \Omega & \Phi \\ \Phi^T & l \end{bmatrix} \begin{bmatrix} \beta \\ b \end{bmatrix} = \begin{bmatrix} c \\ \sum_{k=1}^{l} y_k \end{bmatrix},$$

$$\text{где} \quad \Omega = [\frac{l}{2\gamma} k_{ij} + \sum_{r=1}^{l} k_{rj} k_{ri}]_{i,j \in S},$$

$$\Phi = (\sum_{j=1}^{l} k_{ij})_{i \in S},$$

$$c = (\sum_{j=1}^{l} y_j k_{ij})_{i \in S},$$

$$k_{ij} = \mathcal{K}(x_i, x_j)$$

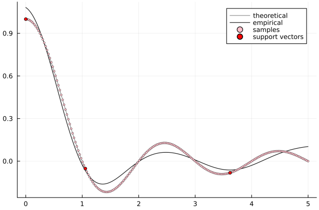
Псевдокод

Data: $\mathcal{K},\ \mathcal{X},\ \mathcal{Y},\ sv\_num$
Result: $S,\ \beta_{best},\ b_{best}$
$l = |\mathcal{X}|,\ \mathcal{L}_{best} = \infty;$
$S,\ \beta_{best},\ b_{best},\ index_{best} = \{\};$
while $|S| \neq sv\_num$ **do**
  for $j \in \mathcal{X} \backslash S$ **do**
    $S \leftarrow j;$
    compute $\Omega,\ \Phi,\ c,\ H;$
    $(\beta,\ b) = H^{-1} * (\sum_{k=1}^{l} y_k,\ c);$
    L = $\mathcal{L}(\beta, b);$
    if $L < \mathcal{L}_{best}$ **then**
      $\beta_{best},\ b_{best},\ \mathcal{L}_{best},\ index_{best} = \beta,\ b,\ L,\ j\ ;$
    $S \rightarrow j\ ;$
  $S \leftarrow index_{best};$

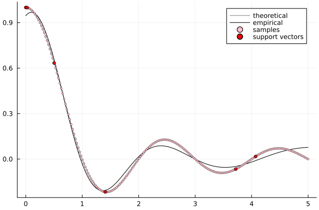
Результаты

- Во всех эксперементах использовалось Гауссовское ядро $\mathcal{K}(x, x') = exp(-\frac{|x-x'|}{2\sigma^2})$
- Восстанавливалсь функция $sinc(x), \; l = 200$
- Регрессия восстанавливалась для количества опорных векторов $n = 3, \; 6, \; 7$ по точным данным, и данным с шумом
- Шум добавлялся нормальным распределенимем $N(0, 0.1)$
- Зависимость СКО от количества опорных векторов проводилась для $n = 1, ..., 12$
  - $\sigma = 0.7, \; \gamma = 10^5$

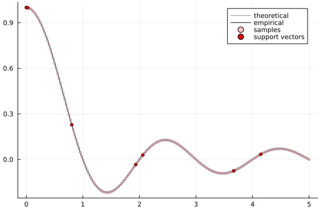
Результаты. Графики (1)



(a) 3 опорных вектора



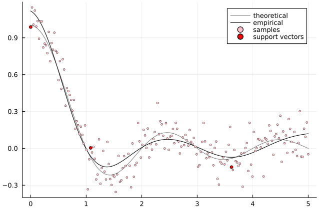
(b) 6 опорных вектора



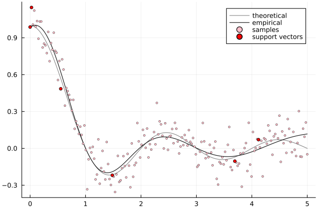
(c) 7 опорных вектора

Рис.: Регрессия без шума

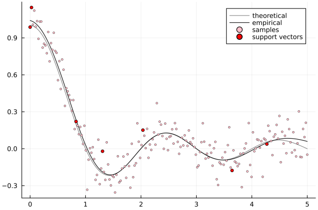
Результаты. Графики (2)



(a) 3 опорных вектора



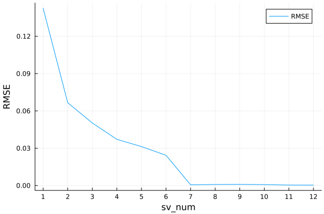
(b) 6 опорных вектора



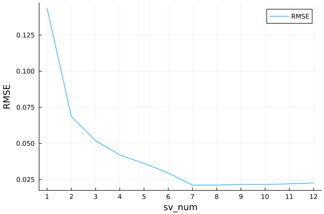
(c) 7 опорных вектора

Рис.: Регрессия с шумом

Результаты. Графики (3)



(a) Без шумов



(b) С шумом

Рис.: Графики зависимости ошибки от количества опорных векторов

Выводы

- Из графиков видно, что 7 - оптимальное количество опорных векторов для этих данных, так как без шума модель идентична $sinc(x)$, также по графикам зависимости СКО от количества опорных векторов видно, что после 7 ошибка особо не уменьшается.
- На графике зависимости ошибки от количества опорных векторов на данных с шумом, можно заметить, что после 10 ошибка немного начинает возрастать. Это можно связать с тем, что минимизация целевой функции происходит жадно, то есть, мы не всегда получаем истинный минимум.
- В общем и целом, алгоритм показал себя хорошо, взамен на небольшую ошибку в точности из-за жадности, мы получаем большой прирост в производительности.