STAT 535

11/17/22

Lecture 15

- Heavy ball
- Coordinate descent
- Stopping

SVM

Lecture V posted
HW 6 due Nov 30
Project
choose predictor 1

Lecture IV: Training predictors, Part II

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

October, 2021

Stochastic gradient methods

Examples: Linear classification with hinge loss, Perceptron

Accelerated gradient

No gradient methods: Coordinate descent

Stopping descent algorithms

Reading HTF Ch.: –, Murphy Ch.: 8.5.2-3 Stochastic gradient descent For more advanced treatment Nocedal and Wrigth.

Accelerated gradient: the "heavy ball" method

$$x^{k+1} = x^k - \eta^k d^k + \gamma^k (x^k - x^{k-1}) \tag{14}$$

- Applies to both standard and stochastic gradient methods, i.e.

$$d^k = \begin{cases} \nabla f(x^k) & \text{gradient descent} \\ \text{noisy gradient} & \text{SGD} \\ \nabla f(x^k + \delta^k (x^k - x^{k-1})) & \text{extragradient methods} \end{cases} \tag{15}$$

- Setting the parameters
  - In the extragradient[3] methods, $\eta^k, \delta^k, \gamma^k$ are obtained by search (or knowledge about M, m)
  - For other methods fix $\gamma^k = \gamma \in (0.5, 1]$ OR use smaller γ early in the training and increase it to near 1 when the steps become smaller.
- More intuition
  - for ill conditioned problems $M \ll m$, the heavy ball "accumulates" the components of the step in the correct direction
  - for SGD, the heavy ball approximates the exact gradient

[3] Nesterov's "optimal", FISTA

$$x^{k+1} - x^k = -\eta d^k + \gamma (x^k - x^{k-1})$$

$$-\eta d^{k-1} + \gamma (x^{k-1} - x^{k-2})$$

$$-\eta d^{k-2} + \gamma (\quad)$$

$$= -\eta \left[d^k + \gamma d^{k-1} + \gamma^2 d^{k-2} + \cdots \right]$$

$\gamma < 1$ } fixed
η

Where good?

Near x^*

noise stays bdd

$d^k \approx 0 + \overbrace{\text{noise}}^{\varepsilon^k}$

$$x^{k+1} - x^k = -\eta \left[\varepsilon^k + \gamma \varepsilon^{k-1} + \cdots \right]$$

("~~reduces variance~~")

next page

Far from x^*

- on plateau
- in narow valley

$d^k = d + \varepsilon^k$ amplifies d

Let $d^k = \underline{d} + \underbrace{\varepsilon^k}$

constant (may be = 0)

0-mean noise, independent

$\text{Var}\, \varepsilon^k = \sigma^2$

$$d^k + \gamma d^{k-1} + \gamma^2 d^{k-2} + \dots = \overbrace{d\,(1 + \gamma + \gamma^2 + \dots)}^{1/1-\gamma}$$

$$+ \underbrace{[\varepsilon^k + \gamma \varepsilon^{k-1} + \dots]}_{\bar{\varepsilon}}$$

$$\text{Var}\, \bar{\varepsilon} = \sigma^2 (1 + \gamma^2 + \gamma^4 \dots)$$

$$= \sigma^2 \frac{1}{1-\gamma^2}$$

$$\Rightarrow \text{noise amplitude} = \sigma \cdot \frac{1}{\sqrt{1-\gamma^2}}$$

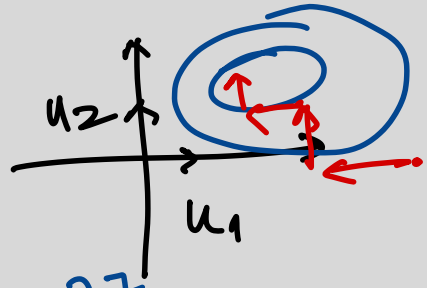
noise amplification > 1

signal amplification $= \frac{1}{1-\gamma} > \frac{1}{\sqrt{1-\gamma^2}}$

+ noise remains bounded

Coordinate descent

J = objective



- d^k is always one of the coordinate axes u_{jk}. Hence $x^{k+1} = x^k + \eta_k u_{jk}$. $\frac{\partial J}{\partial x_i}$
- Note that line search is necessary, and that the minimum can be on either side of x^k so η_k can take negative values.

→ line search ✗

Convergence Theoretical and empirical results suggest that coordinate descent has similar convergence rate as the steepest descent (i.e linear in the best case).
While in a general case coordinate descent is suboptimal, there are several situations when it is worth considering

1. When line minimization can be done analytically. This can save one the often expensive gradient computation.
2. When the coordinate axes affect the function value approximately independently, or (in statistics) when the coordinate axes are uncorrelated. Then minimizing along each axis separately is (nearly) optimal.
3. When there exists a natural grouping of the variables. Then one can optimize one group of variables while keeping the other constant. Again, we hope that the groups are "independent", or that optimizing one group at a time can be done analytically, or it's much easier than computing the gradient w.r.t all variables simultaneously. This idea is the basis of many *alternate minimization* methods, including the well known EM algorithm.

- EM Algorithm
- Max Likelihood for models with hidden variables

Loss (z, θ)

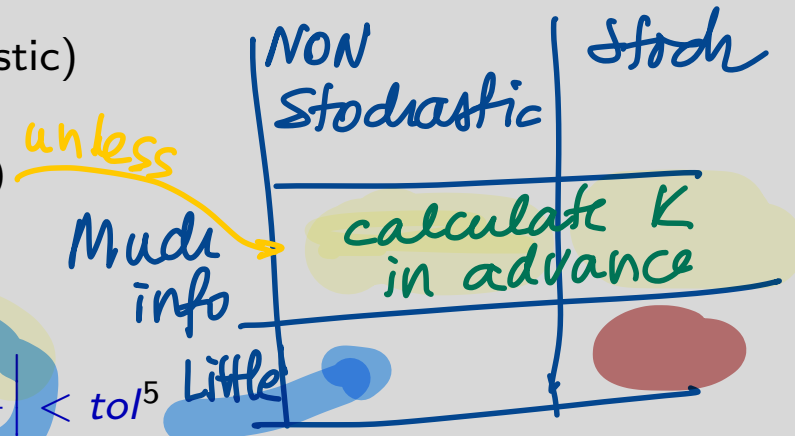
Stopping descent algorithms

Paradigm

- What we would like is to stop when the error $f(x^*) - f(x^i)$ or $||x^* - x^k||$ is "small enough". This is possible for special classes of functions, in particular for convex functions.
- In general, we stop when some other computable quantity is "small enough", i.e smaller than a tolerance tol.

Stopping conditions for Batch algorithms (non-stochastic)

- What **not** to do:
  - stop when $k = 100$ (or any other pre-set number K)
  - stop when $||\nabla f(x^k)|| < tol$ Exercise Why?
  - set $tol < \sqrt{\varepsilon_{machine}} \approx 10^{-8}$
- What to do:
- The "poor man's" stopping condition: $\left|1 - \frac{f(x^{k+1})}{f(x^k)}\right| < tol$[5]
- The "pro's" stopping condition: Newton step $= ||\nabla^2 f(x^k)^{-1} \nabla f(x^k)|| < tol$.
  Note: don't compute it at every step (unless you are actually running Newton method), but only once in a while, depending on n and what descent algorithm you are using.



$\approx$ distance $||x^k - x^*||$

[5] The $||$ are not necessary if the method you use guarantees $f(x^{k+1}) < f(x^k)$ but this is not always the case. Note also that this fails if $f(x^k) = 0$ or changes sign. But it works well for loss functions as they are always positive.

Stopping SGD

[ML framework, minimize J or $\hat{L}$ over parameters, (x, y) =data, f = predictor]

- L is strongly convex, and lower bound on λ known. (Hence you are using "modern" SGD.)
  - Fix K in advance by using the theorem $E[||\theta^* - \theta^k||^2] \leq \frac{c' G^2}{\lambda^2 K^2}$ (c' is a function of c) and setting $tol^2 > \frac{c' G^2}{\lambda^2 K^2}$ → calculate K
- otherwise (old-fashioned SGD, with $n' = 1$ or larger)
  - every M iterations, where M is large enough, test if $\frac{||\bar{\theta}^k - \bar{\theta}^{k-M}||}{||\bar{\theta}^k||} < tol$ ($\bar{\theta}$)

Lecture V: Support Vector Machines

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

November, 2022

Linear SVM's

- The margin and the expected classification error
- Maximum Margin Linear classifiers
- Linear classifiers for non-linearly separable data

Non linear SVM

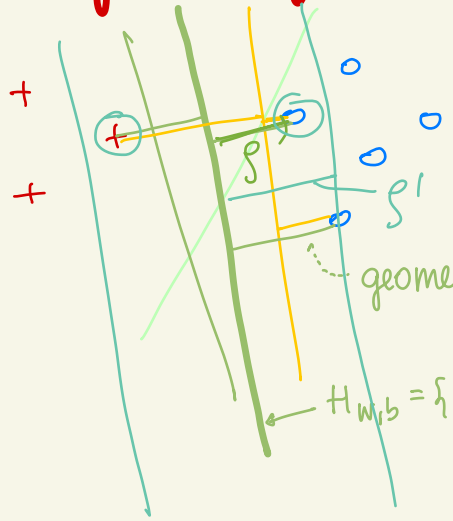
- The "kernel trick"
- Kernels
- Prediction with SVM

Extensions

- L_1 SVM
- Multi-class and One class SVM
- SV Regression

Reading HTF Ch.: Ch. 12.1–3, Murphy Ch.: Ch 14 (14.1,14.2–14.2.4 kernels, 14.4 and equations (14.28,14.29) kernel trick, 14.5.1.–3 Support Vector Machines), Bach Ch.: 7.1–7.4, 7.7
Additional Reading: C. Burges - "A tutorial on SVM for pattern recognition"
These notes: Appendices (convex optimization) are optional.

Large margin classifiers



- linear classifier

$f(x) = w^T x + b$

$x, w \in \mathbb{R}^d$

$b \in \mathbb{R}$

linearly separable $\Leftrightarrow$ $\hat{L} = 0$ for some w, b

assume $\|w\| = 1$

$\Downarrow$

$d(x, H_{w,b}) = |w^T x + b| = |f(x)|$

$\hat{L}_\rho = \frac{1}{n} |\{ i \; f(x^i) y^i < \rho \}|$

margin errors

$\hat{L}_\rho = 0$

$\hat{L}_{\rho'} = \frac{1}{n} \cdot 2$

The margin and the expected classification error

Theorem Let $\mathcal{F} = \{\text{sgn}(w^T x),\ ||w|| \leq \Lambda,\ ||x|| \leq R\}$ and let $\rho > 0$ be any "margin". Then for any $f \in \mathcal{F}$, w.p $1 - \delta$ over training sets

expected Loi

margin error rate

$$L_{01}(f) \leq \hat{L}_\rho + \sqrt{\frac{c}{n}\left(\frac{R^2\Lambda^2}{\rho^2}\ln n^2 + \ln\frac{1}{\delta}\right)} \tag{5}$$

confidence

where c is a universal constant and $\hat{L}_\rho$ is the fraction of the training examples for which

$$\text{margin } y^i f(x^i) \equiv y^i w^T x_i < \rho \tag{6}$$

Pick any $\rho > 0$

- a data point i that satisfies (6) for some ρ is called a **margin error**
- For $\rho = 0$ the margin error rate $\hat{L}_\rho$ is equal to $\hat{L}_{01}$

Next pick max ρ so that $\hat{L}_\rho = 0$

How to find $w, b = ?$

Maximum Margin Linear classifiers

Support Vector Machines appeared from the convergence of Three Good Ideas

Assume (for the moment) that the data are linearly separable.

- Then, there are an infinity of linear classifiers that have $\hat{L}_{01} = 0$. Which one to choose?

irst idea Select the classifier that has **maximum margin** ρ on the training set.

By SRM, we should choose the (w, b) parameters that minimize $\hat{L}(w, b) + R(h_{w,b})$, where $h_{w,b}$ is given by (**??**):

- For any parameters (w, b) that perfectly classify the data $\hat{L}(w, b) = 0$.
- Among these, the best (w, b) is the one that minimizes $R(h_{w,b})$
- $R(h)$ increases with h, and $h_{w,b}$ decreases when ρ increases
- Hence, by SRM we should choose

$$\underset{\rho,w,b:\hat{L}(w,b)=0}{\operatorname{argmax}} \rho, \quad \text{s.t. } d(x, H_{w,b}) \geq \rho \text{ for } i = 1 : n, \tag{7}$$

where $d()$ denotes the Euclidean distance and $H_{w,b} = \{x \mid w^T x + b = 0\}$ is the decision boundary of the linear classifier.

- Because $d(x, H_{w,b}) = \frac{|w^T x + b|}{||w||}$ (proof in a few slides) (7) becomes

$$\underset{\rho,w,b:\hat{L}(w,b)=0}{\operatorname{argmax}} \rho, \quad \text{s.t. } \frac{|w^T x^i + b|}{||w||} \geq \rho \text{ for } i = 1 : n, \tag{8}$$

Maximum Margin Linear classifiers

We continue to transform (8)

- If all data correctly classified, then $y^i(w^T x^i + b) = |w^T x^i + b|$. Therefore (8) has the same solution as

$$\underset{\rho,w,b}{\operatorname{argmax}} \rho, \quad \text{s.t. } \frac{y^i(w^T x^i + b)}{||w||} \geq \rho \text{ for } i = 1:n, \tag{9}$$

- Note now that the problem (9) is underdetermined. Setting $w \leftarrow Cw, b \leftarrow Cb$ with $C > 0$ does not change anything.
- We add a cleverly chosen constraint to remove the indeterminacy; this is$||w|| = 1/\rho$, which allows us to eliminate the variable ρ. We get

$$\underset{w,b}{\operatorname{argmax}} \frac{1}{w}, \quad \text{s.t. } y^i(w^T x^i + b) \geq 1 \text{ for } i = 1:n, \tag{10}$$

Note: the successive problems (7),(8),(9),. . . are equivalent in the sense that their optimal solution is the same.

Alternative derivation of (10)

st idea Select the classifier that has maximum margin on the training set, by the alternative definition of margin.
Formally, define $\min_{i=1:n} y^i f(x^i)$ be the **margin of classifier f on $\mathcal{D}$**. Let $f(x) = w^T x + b$, and choose w, b that

① $\text{maximize}_{w \in \mathbb{R}^n, b \in \mathbb{R}} \left\{ \min_{i=1:n} y^i (w^T x^i + b) \right\}$ → smallest margin

- Remarks
  - (if data is linearly separable), there exist classifiers with margins > 0
  - one can arbitrarily increase the margin of such a classifier by multiplying w and b by a positive constant.
  - Hence, we need to "normalize" the set of candidate classifiers by requiring instead

$$\text{maximize} \min_{i=1:n} d(x, H_{w,b}), \text{ s.t. } y^i (w^T x^i + b) \geq 1 \text{ for } i = 1:n, \quad (11)$$

  where $d()$ denotes the Euclidean distance and $H_{w,b} = \{ x \mid w^T x + b = 0 \}$ is the decision boundary of the linear classifier.
  - Under the conditions of (11), because there are points for which $|w^T x + b| = 1$, maximizing $d(x, H_{w,b})$ over w, b for such a point is the same as

③

$$\max_{w,b} \frac{1}{||w||}, \text{ s.t. } \min_i y_i (w^T x + b) = 1 \quad (12)$$

② $\max_{w,b} \left\{ \min_i \frac{y^i (w^T x^i + b)}{||w||} \right\}$ → $d(x, H_{wb})$

s.t. $y^i (w^T x^i + b) \geq 1$

③ $\max_{w,b} \min_i \frac{|w^T x^i + b|}{\|w\|}$ st. $y^i(w^T x^i + b) \geq 1$

$\min_i \frac{|w^T x^i + b|}{\|w\|} = 1$

$\max_{w,b} \frac{1}{\|w\|}$ st. —— w ——

$\min_{w,b} \|w\|$ s.t —— w ——

④ $\min_{w,b} \frac{1}{2}\|w\|^2$ s.t. $y^i(w^T x^i + b) \geq 1$

quadratic

linear constraints

Second idea

The **Second idea** is to formulate (10) as a **quadratic** optimization problem.

$$\min_{w,b} \frac{1}{2}||w||^2 \text{ s.t } y^i(w^T x^i + b) \geq 1 \text{ for all } i = 1 : n \tag{13}$$

This is the **Linear SVM (primal) optimization problem**

- This problem has a strongly convex **objective** $||w||^2$, and **constraints** $y^i(w^T x^i + b)$ linear in (w, b).
- Hence this is a convex problem, and can be studied with the tools of convex optimization.

The distance of a point x to a hyperplane $H_{w,b}$

$$d(x, H_{w,b}) = \frac{|w^T x + b|}{||w||} \tag{14}$$

Intuition: denote

$$\tilde{w} = \frac{w}{||w||}, \ \tilde{b} = \frac{b}{||w||}, \ x' = \tilde{w}^T x. \tag{15}$$

Obviously $H_{w,b} = H_{\tilde{w},\tilde{b}}$, and x' is the length of the projection of point x on the direction of w.

The distance is measured along the normal through x to H; note that if $x' = -\tilde{b}$ then $x \in H_{w,b}$ and $d(x, H_{w,b}) = 0$; in general, the distance along this line will be $|x' - (-\tilde{b})|$.