Bundle Methods for Machine Learning

(Teo, Vishwanathan, Smola, Le)
JMLR 2010, NIPS 2007

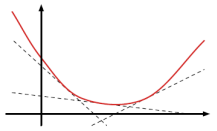
Presented by Kevin Duh
Bayes Reading Group 6/4/2010

1-slide summary

- Many machine learning methods involve solving a minimum regularized risk objective

$$\min_w \; J(w) := \lambda\Omega(w) + R_{\text{emp}}(w),$$
$$\text{where } R_{\text{emp}}(w) := \frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, w)$$

- Cutting-plane algorithm & Bundle methods solve it iteratively by using a piece-wise lower bound



Why I chose this paper

- These optimization methods (invented in 1960s) are becoming popular in supervised learning
  - Very fast
  - Scale to large datasets
- Handles non-smooth convex optimization, so widely applicable
  - Can be used when LBFGS fails

Outline

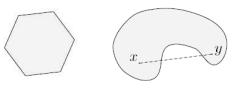
1. Background
2. Cutting plane algorithm
3. Bundle method
4. Different loss functions

Warm-up

- Convex Set:
  - a set is a convex set if it contains the line segment joining any of its points

$$x, y \in S; a, b \geq 0; a + b = 1$$
$$\Rightarrow ax + by \in S$$

  - are these sets convex?

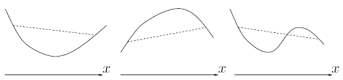


Background: Convex functions

- Convex function:
  - A function is convex is its domain is a convex set and the segment joining any two points on f do not have values lower than f

$$\forall x, y \in dom(f); a, b \geq 0, a + b = 1$$
$$af(x) + bf(y) \geq f(ax + by)$$

- What's convex, what's concave?



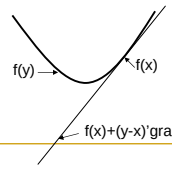
Convex & differentible functions

- Gradient exists if f is differentiable

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$$

- 1st-order condition: a differentiable f is convex iff:

$$f(y) \geq f(x) + \langle y - x, \nabla f \rangle \quad \forall y$$



Gradient provides a global lower bound to f

Nonsmooth (non-differentiable) functions

Derivative at x>1 is 1
Derivative at x<1 is -1
Derivative at x=0?

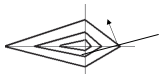
- What if f is not differentiable, e.g.
  - L1-regularizer or |x|
  - Envelope function: $f(x) = \max_{s \in S} \phi_s(x)$
- Subgradient: a vector s is a subgradient if

$$f(y) \geq f(x) + \langle y - x, s \rangle \quad \forall y$$

- There may exist many subgradients at a point
  - The set of subgradients is called the subdifferential
  - The methods we deal with only require one subgradient

Subgradient Method for optimizing non-smooth functions

- Similar as gradient descent, except:
  - works on non-differentiable functions
  - step-lengths not chosen by line-search, but fixed
  - it is not a descent method
    - descent direction d can only be defined by <d,s> <0 for all s in sub-differential



- Pseudo-code:
  - Repeat until convergence:
    1. s = subgradient at f(x)
    2. x = x – stepsize * s
    3. keep track of best x so far

Outline

1. Background
2. Cutting plane algorithm
3. Bundle method
4. Different loss functions

Cutting-plane algorithm for optimizing non-smooth functions

- Recall subgradient forms a lowerbound on f
- Main Idea:
  - If we have multiple subgradients at different points, we get a tighter lowerbound
  - The lowerbound improves with each iteration, so minimizing the lowerbound eventually minimizes the desired objective

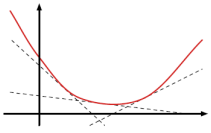
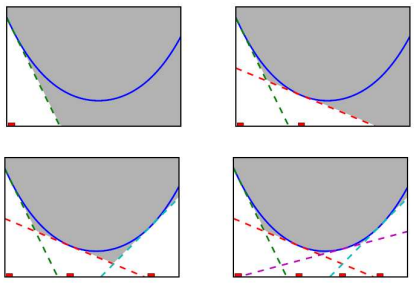


Figure: lower bound improves after each iteration



The math

- Overall optimization goal:

$$\min_w J(w) := \lambda\Omega(w) + R_{\text{emp}}(w),$$
$$\text{where } R_{\text{emp}}(w) := \frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, w)$$

- Lower bound:
  - Given sequence of iterates w and subgradients s, the (piecewise-linear) lower bound is:

$$J(w) \geq J_t^{\text{CP}}(w) := \max_{1\leq i\leq t}\{J(w_{i-1}) + \langle w - w_{i-1}, s_i\rangle\}$$

  - Because $J(w) \geq J(w') + \langle w - w', s'\rangle$
  - At each iteration, compute $w_t := \operatorname{argmin}_w J_t^{\text{CP}}(w).$

Note: Slight change of notation starting now: function f(x) → J(w)

13

Cutting-plane pseudocode

1. Compute J(w_t) and its subgradient s_t
2. Compute error $\epsilon_t := \min_{0\leq i\leq t} J(w_i) - J_t^{\text{CP}}(w_t)$
3. If error < threshold, stop
4. Update bound $J_t^{\text{CP}}(w) := \max_{1\leq i\leq t}\{J(w_{i-1}) + \langle w - w_{i-1}, s_i\rangle\}$
5. Optimize it to get new iterate $w_t := \operatorname{argmin}_w J_t^{\text{CP}}(w)$
6. Goto step 1

14

Why does this work?

Let w* be optimal solution, then

$$J(w_i) \geq J(w^*) \Rightarrow \min_{0\leq i\leq t} J(w_i) \geq J(w^*)$$

By construction,

$$J(w) \geq J_t^{CP}(w), \forall w \Rightarrow J(w^*) \geq J_t^{CP}(w_t)$$

So optimal point is sandwiched:

$$\min_{0\leq i\leq t} J(w_i) \geq J(w^*) \geq J_t^{CP}(w_t)$$

This error is monotonically decreases.
When it reaches zero, we have w*

$$\epsilon_t := \min_{0\leq i\leq t} J(w_i) - J_t^{\text{CP}}(w_t)$$

15

Final word on cutting plane-algorithm

- It has nice stopping criteria
  - (better than subgradient method)
- Cost is solving linear programs per iteration:
  - Size of this subproblem grows with each iteration
  - But usually this can be solved quickly
- Speed depends critically on the set of cutting planes
  - Zig-zag behavior possible, slowing down convergence

16

Outline

1. Background
2. Cutting plane algorithm
3. Bundle method
4. Different loss functions

17

Standard Bundle Method

- Zig-zag in cutting-plane is caused by taking large steps and neglecting previous solutions
- Bundle methods extend cutting-plane by ensuring new iterate is not too far

$$w_t := \operatorname{argmin}_w\{\tfrac{\zeta_t}{2}\|w - \hat{w}_{t-1}\|^2 + J_t^{\text{CP}}(w)\}$$

18

(Standard) Bundle method pseudo-code

Algorithm 1 Proximal Bundle Method

input & initialization: $\epsilon \geq 0$, $\rho \in (0,1)$, w_1, $t \leftarrow 0$, $\hat{w}_0 \leftarrow w_0$
loop
  $t \leftarrow t+1$
  Compute $J(w_{t-1})$ and $s_t \in \partial_w J(w_{t-1})$
  Update model $J_t^{\mathrm{CP}}(w) := \max_{1\leq i\leq t}\{J(w_{i-1}) + \langle w - w_{i-1}, s_i\rangle\}$
  $\bar{w}_t \leftarrow \mathrm{argmin}_w\, J_t^{\mathrm{CP}}(w) + \frac{\zeta_t}{2}\|w - \hat{w}_{t-1}\|^2$
  $\epsilon_t \leftarrow J(\bar{w}_{t-1}) - \left[J_t^{\mathrm{CP}}(\bar{w}_t) + \frac{\zeta_t}{2}\|\bar{w}_t - \hat{w}_{t-1}\|^2\right]$
  if $\epsilon_t < \epsilon$ **then return** $\bar{w}_t$
  Linesearch: $\eta_t \leftarrow \mathrm{argmin}_{\eta\in\mathbb{R}} J(\hat{w}_{t-1} + \eta(\bar{w}_t - \hat{w}_{t-1}))$ (if expensive, set $\eta_t = 1$)
  $w_t \leftarrow \hat{w}_{t-1} + \eta_t(\bar{w}_t - \hat{w}_{t-1})$
  if $J(\hat{w}_{t-1}) - J(w_t) \geq \rho\epsilon_t$ **then**
    SERIOUS STEP: $\hat{w}_t \leftarrow w_t$
  else
    NULL STEP: $\hat{w}_t \leftarrow \hat{w}_{t-1}$
  end if
end loop

This paper argues that some parameters are hard to tune:

ζ_t ρ

Proposed Bundle Method (BMRM)

- Regularized risk minimization objective already has a regularization term:

$$\min_w \; J(w) := \lambda\Omega(w) + R_{\mathrm{emp}}(w),$$

- So optimize this subproblem: $J_t(w) := \lambda\Omega(w) + R_t^{\mathrm{CP}}(w)$, $w_t := \min_w J_t(w)$
- No need for serious/null step

Algorithm 2 BMRM

input & initialization: $\epsilon \geq 0$, w_0, $t \leftarrow 0$
repeat
  $t \leftarrow t+1$
  Compute $a_t \in \partial_w R_{\mathrm{emp}}(w_{t-1})$ and $b_t \leftarrow R_{\mathrm{emp}}(w_{t-1}) - \langle w_{t-1}, a_t\rangle$
  Update model: $R_t^{\mathrm{CP}}(w) := \max_{1\leq i\leq t}\{\langle w, a_i\rangle + b_i\}$
  $w_t \leftarrow \mathrm{argmin}_w\, J_t(w) := \lambda\Omega(w) + R_t^{\mathrm{CP}}(w)$
  $\epsilon_t \leftarrow \min_{0\leq i\leq t} J(w_i) - J_t(w_t)$
until $\epsilon_t \leq \epsilon$
return w_t

(subgradient of R_{emp})	$a_t \in \partial_w R_{\mathrm{emp}}(w_{t-1})$
(offset)	$b_t := R_{\mathrm{emp}}(w_{t-1}) - \langle w_{t-1}, a_t\rangle$
(piecewise linear lower bound of R_{emp})	$R_t^{\mathrm{CP}}(w) := \max_{1\leq i\leq t}\{\langle w, a_i\rangle + b_i\}$
(piecewise convex lower bound of J)	$J_t(w) := \lambda\Omega(w) + R_t^{\mathrm{CP}}(w)$
(iterate)	$w_t := \min_w J_t(w)$
(approximation gap)	$\epsilon_t := \min_{0\leq i\leq t} J(w_i) - J_t(w_t)$

In more detail: how to solve subproblem in step 6

- Reformulate as constrained optimization:

$$w_t = \mathrm{argmin}_w\, J_t(w) := \lambda\Omega(w) + \max_{1\leq i\leq t}\langle w, a_i\rangle + b_i$$

$$\min_{w,\xi}\; \lambda\Omega(w) + \xi$$
$$\text{subject to } \xi \geq \langle w, a_i\rangle + b_i \text{ for } i = 1,\ldots,t$$

- Then call linear/quadratic program depending on regularizer
  - # constraints = #iterations, unrelated to #samples!
  - Dual program for L2 regularizer:

$$\alpha_t = \mathrm{argmax}_{\alpha\in\mathbf{R}^t}\{-\tfrac{1}{2\lambda}\alpha^\top A^\top A\alpha + \alpha^\top b \mid \alpha \geq 0,\ \|\alpha\|_1 = 1\}$$

Convergence Analysis

Theorem 4 *Assume that* $\max_{u\in\partial_w R_{\mathrm{emp}}(w)}\|u\| \leq G$ *for all* $w \in \mathrm{dom}\, J$. *Also assume that* Ω^* *has bounded curvature, i.e.,* $\left\|\partial_\mu^2\Omega^*(\mu)\right\| \leq H^*$ *for all* $\mu \in \{-\lambda^{-1}\sum_{i=1}^{t+1}\alpha_i a_i$ *where* $\alpha_i \geq 0,\ \forall i$ *and* $\sum_{i=1}^{t+1}\alpha_i = 1\}$. *In this case we have*

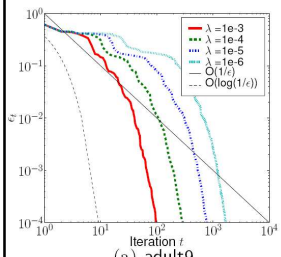
$$\epsilon_t - \epsilon_{t+1} \geq \frac{\epsilon_t}{2}\min(1, \lambda\epsilon_t/4G^2H^*). \tag{27}$$

Furthermore, if $\left\|\partial_w^2 J(w)\right\| \leq H$, *then we have*

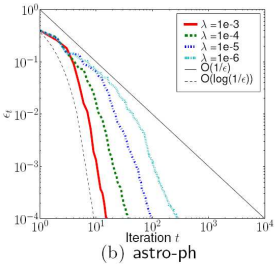
Every iteration the error is halved!!

$$\epsilon_t - \epsilon_{t+1} \geq \begin{cases} \epsilon_t/2 & \text{if } \epsilon_t > 4G^2H^*/\lambda \\ \lambda/8H^* & \text{if } 4G^2H^*/\lambda \geq \epsilon_t \geq H/2 \\ \lambda\epsilon_t/4HH^* & \text{otherwise} \end{cases} \tag{28}$$

Experiments in actual speed



(a) adult9



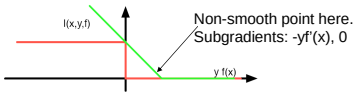
(b) astro-ph

Outline

1. Background
2. Cutting plane algorithm
3. Bundle method
4. Different loss functions

Binary classification

- Accuracy-based loss: $\Delta(y, y') = \begin{cases} 0 & \text{if } y = y' \\ 1 & \text{otherwise} \end{cases}$
- Convex upper-bounds:
  - Soft margin loss: $l(x, y, f) = \max(0, 1 - yf(x))$



  - Logisitic: $\log(1+\exp(-yf(x)))$
  - MCE (Katagiri et. al.) – sigmoid, with adjustable parameter
  - Gaussian process classifier, MAP solution:
    - Minimize $\frac{1}{2}\mathbf{f}^\top K^{-1}\mathbf{f} - \sum_{i=1}^{n} \log p(y_i|f_i)$,

Structured Prediction

- Similar to previous slide, convex upper bounds for structured loss $\Delta(y, y')$
  - CRF:

$$l(x, y, w) = \log \sum_{y' \in \mathcal{Y}} \exp\left(\langle w, \phi(x, y')\rangle\right) - \langle w, \phi(x, y)\rangle$$
$$\partial_w l(x, y, w) = \mathbf{E}_{y' \sim p(y'|x)}\left[\phi(x, y')\right] - \phi(x, y).$$

  - Structured SVM:

$$l(x, y, w) = \max_{y' \in \mathcal{Y}} \Gamma(y, y')\left\langle w, \phi(x, y') - \phi(x, y)\right\rangle + \Delta(y, y')$$
$$\partial_w l(x, y, w) = \Gamma(y, \bar{y}(x))\left[\phi(x, \bar{y}(x)) - \phi(x, y)\right]$$
$$\bar{y}(x) := \operatorname*{argmax}_{y'} \Gamma(y, y')\left\langle w, \phi(x, y') - \phi(x, y)\right\rangle + \Delta(y, y')$$

For bundle methods, just collect the vectors $\partial_w l(x, y, w)$ and give to the LP/QP

ROC score

- AUC is not continuous in w:

$$\mathrm{AUC}(x, y, w) = \frac{1}{m_+ m_-} \sum_{y_i < y_j} \mathbf{I}(\langle w, x_i\rangle < \langle w, x_j\rangle),$$

- But this nonsmooth convex bound is:

$$R_{\mathrm{emp}}(w) = \frac{1}{m_+ m_-} \sum_{y_i < y_j} \max(0, 1 + \langle w, x_i - x_j\rangle)$$

- We can directly calculate subgradients in closed-form, but we can also obtain from an algorithm if it's more efficient
  - See algorithm 7 in JMLR paper

Do you see the pattern?

1. Give me any problem, with any evaluation metric (may be difficult to optimize)
2. Think of a convex upper bound for the metric
   - This bound does not need to be smooth
   - Just need to get subgradients from it
3. Solve with:
   - Gradient descent, LFBGS; or
   - Subgradient method, Bundle method, etc.
4. Done: submit NIPS paper

Discussions

- Fear not non-smooth convex functions
- What about non-convex optimization?
  - EM-style training where M is solved by bundle
    - Yu & Joachims, Learning Structural SVMs w/ Latent Variables (ICML09)
  - Modified bundle method:
    - Do & Artieres, Large margin training of HMMs w/ partially-observed states (ICML09)