Subgradient Descent

Really, the simplest algorithm in the world. Goal:

$$\underset{x}{\text{minimize}} \; f(x)$$

Just iterate

$$x_{t+1} = x_t - \eta_t g_t$$

where η_t is a stepsize, $g_t \in \partial f(x_t)$.

Why subgradient descent?

- Lots of non-differentiable convex functions used in machine learning:

$$f(x) = \left[1 - a^T x\right]_+, \quad f(x) = \|x\|_1, \quad f(X) = \sum_{r=1}^{k} \sigma_r(X)$$

  where σ_r is the rth singular value of X.
- Easy to analyze
- Do not even need true sub-gradient: just have $\mathbb{E} g_t \in \partial f(x_t)$.

Proof of convergence for subgradient descent

Idea: bound $\|x_{t+1} - x^*\|$ using subgradient inequality. Assume that $\|g_t\| \leq G$.

$$\begin{aligned}\|x_{t+1} - x^*\|^2 &= \|x_t - \eta g_t - x^*\|^2 \\ &= \|x_t - x^*\|^2 - 2\eta g_t^T(x_t - x^*) + \eta^2 \|g_t\|^2\end{aligned}$$

Recall that

$$f(x^*) \geq f(x_t) + g_t^T(x^* - x_t) \quad \Rightarrow \quad -g_t^T(x_t - x^*) \leq f(x^*) - f(x_t)$$

so

$$\|x_{t+1} - x^*\|^2 \leq \|x_t - x^*\|^2 + 2\eta\left[f(x^*) - f(x_t)\right] + \eta^2 G^2.$$

Then

$$f(x_t) - f(x^*) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta} + \frac{\eta}{2}G^2.$$

Almost done...

Sum from $t = 1$ to T:

$$\sum_{t=1}^{T} f(x_t) - f(x^*) \leq \frac{1}{2\eta} \sum_{t=1}^{T} \left[\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 \right] + \frac{T\eta}{2} G^2$$

$$= \frac{1}{2\eta} \|x_1 - x^*\|^2 - \frac{1}{2\eta} \|x_{T+1} - x^*\|^2 + \frac{T\eta}{2} G^2$$

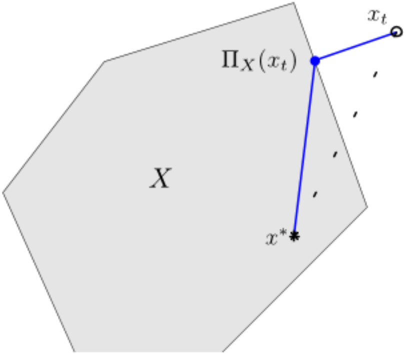
Now let $D = \|x_1 - x^*\|$, and keep track of min along run,

$$f(x_{\text{best}}) - f(x^*) \leq \frac{1}{2\eta T} D^2 + \frac{\eta}{2} G^2.$$

Set $\eta = \frac{D}{G\sqrt{T}}$ and

$$f(x_{\text{best}}) - f(x^*) \leq \frac{DG}{\sqrt{T}}.$$

Extension: projected subgradient descent



Now have a convex constraint set X.
Goal:

$$\underset{x \in X}{\text{minimize}} \quad f(x)$$

Idea: do subgradient steps, project x_t back into X at every iteration.

$$x_{t+1} = \Pi_X(x_t - \eta g_t)$$

Proof:

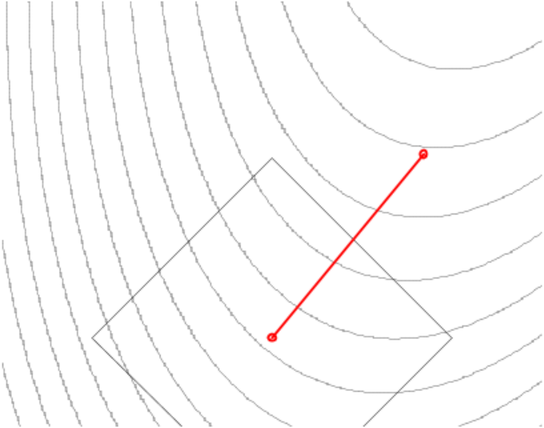
$$\|\Pi_X(x_t) - x^*\| \leq \|x_t - x^*\|$$

if $x^* \in X$.

Projected subgradient descent has been applied to PRIMAL of SVM: http://pages.cs.wisc.edu/~swright/talks/sjw-complearning.pdf Slide #28-30 & dual Slide #18

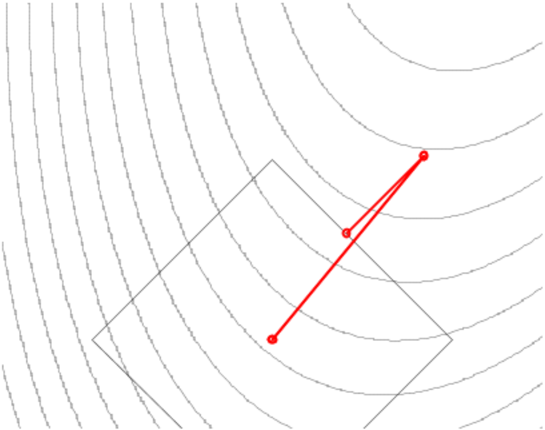
Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



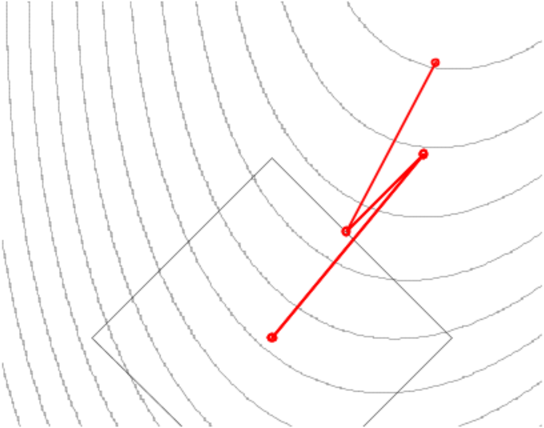
Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



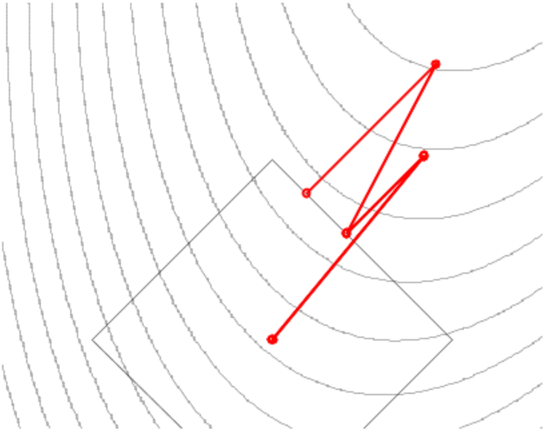
Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



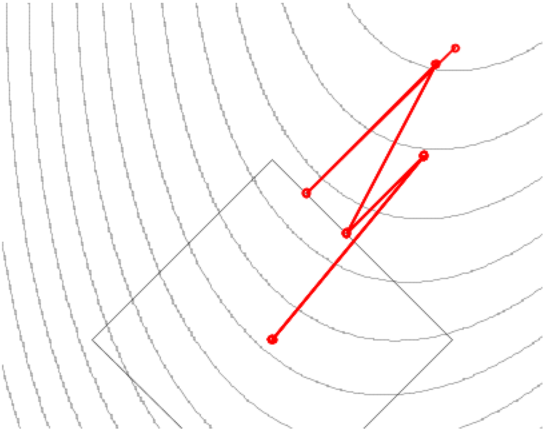
Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \le 1$$



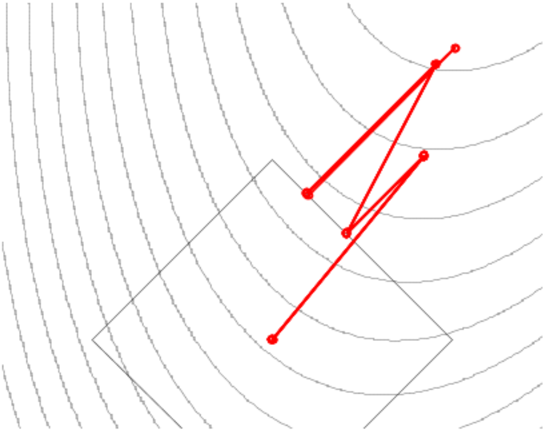
Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



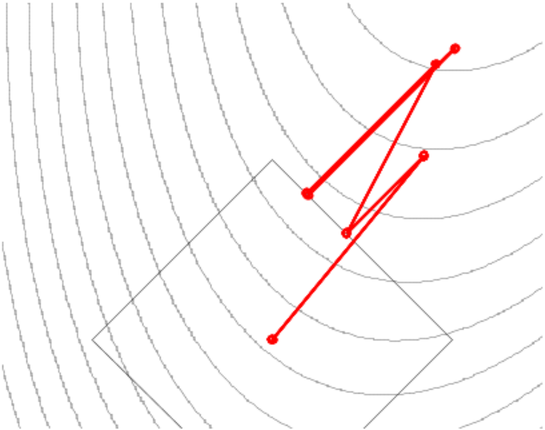
Projected subgradient example

$$\underset{x}{\text{minimize}} \; \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



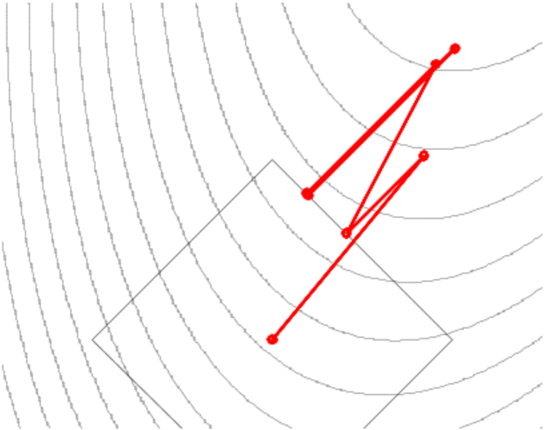
Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



Projected subgradient example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - b\| \quad \text{s.t.} \quad \|x\|_1 \leq 1$$



Convergence results for (projected) subgradient methods

[Part of tutorial]

- Any decreasing, non-summable stepsize $\eta_t \to 0$, $\sum_{t=1}^{\infty} \eta_t = \infty$ gives

$$f\left(x_{\text{avg}(t)}\right) - f(x^*) \to 0.$$

- Slightly less brain-dead analysis than earlier shows with $\eta_t \propto 1/\sqrt{t}$

$$f\left(x_{\text{avg}(t)}\right) - f(x^*) \leq \frac{C}{\sqrt{t}}$$

- Same convergence when g_t is random, i.e. $\mathbb{E} g_t \in \partial f(x_t)$. Example:

$$f(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left[1 - y_i x_i^T w\right]_+$$

Just pick random training example.

Quadratic Optimization: Primal Active-Set Algorithm

Consider the quadratic optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + \beta \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \end{array} \tag{1}$$

where $Q \succ 0$. [assume $A \in \mathbb{R}^{m\times n}$, $\text{rank}(A) = m$ & $m < n$]

Side notes:

① If we had equality constraint $Ax=b$ instead, we could pose an unconstrained optimisation problem by writing x as:

ⓐ $x = x_{\text{particular}} + x_{\text{null space}}$

OR equivalently

ⓑ $x = A^+ b + [I_n - A^+A]\,\vec{\phi}$ → Arbitrary n-dimensional param vector

Moore Penrose pseudo inverse: If $\text{rank}(A)=m$ & $m<n$

$A^+ = A^T(AA^T)^{-1}$ which is right inverse of A

ⓒ Simplifying ⓑ using SVD decomposition of A: $A = U\Sigma V^T$

Σ: $[S, 0]$ & $S = \text{diag}\{\sigma_1 \ldots \sigma_m\}$

U in $\mathbb{R}^{m\times m}$, V in $\mathbb{R}^{n\times n}$ — Both orthogonal

$x = V_r\phi + A^+b$

ϕ → Arbitrary r dimensional vector

V_r: Last $r = n-m$ columns of V

Now suppose we had inequality constraints $Ax \geq b$ instead

①

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} + \beta \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \end{array} \qquad (1)$$

(β: can be ignored)

where $Q \succ 0$.

② KKT conditions are:

$Q\hat{x} + c - \sum_i \hat{\lambda}_i a_i = 0$ - - - - - ⓐ

$(a_i^T \hat{x} - b_i)\hat{\lambda}_i = 0$ for $i = 1 \ldots m$ - - - - - ⓑ

$\hat{\lambda}_i \geq 0$ for $i = 1 \ldots m$ - - - - - ⓒ

$A\hat{x} \geq b$ - - - - - ⓓ

③ If x^* lies in the interior of feasible region (i.e. $A\hat{x} > b$)

then: $\hat{\lambda} = 0$

$\hat{x} = -Q^{-1}c$ ... unique global minimizer of $f(x)$ without constraints if Q is positive definite (so that $\nabla^2 f(x) \succ 0$)

④ What if some $a_i^T x^* = b_i$, $i \in I^*$? **Ans: You need to solve iteratively**

Let x_k be some intermediate point in the iteration and I_k be the intermediate set of active constraints

i.e $a_j^T x_k = 0 \quad \forall j \in I_k$.

Let $x_{k+1} = x_k + \alpha_k d_k$ ($\alpha_k d_k \to d$)

The objective, as a function of d with $x = x_k + d$ is:

$$f_k(d) = \frac{1}{2}(x_k+d)^T Q (x_k+d) + c^T(x_k+d)$$

$$= \frac{1}{2} d^T Q d + (x_k^T Q + c^T) d + \left(\frac{1}{2} x_k^T Q x_k + c^T x_k\right)$$

$(x_k^T Q + c^T) \to g_k$, $\left(\frac{1}{2} x_k^T Q x_k + c^T x_k\right) \to C_k$ (constant)

$$= \frac{1}{2} d^T Q d + g_k^T d + C_k$$

IDEA BEHIND ACTIVE SET ALGO:

$$d_k = \operatorname{argmin} \frac{1}{2} d^T Q d + g_k^T d \quad \text{s.t.} \quad a_j^T d = 0 \quad \forall j \in I_k. \qquad (A)$$

(I) $d_k = 0$ i.e x_k satisfies first order necessary conditions: (i.e rank $[A_{I_k}^T \; g_k] = \text{rank}[A_{I_k}^T]$)

$$Q x_k + c - \sum_{i \in I_k} \lambda_i a_i = 0$$

we already know that: $a_i^T x_k - b_i > 0 \quad \forall i \notin I_k$

and that: $a_i^T x_k - b_i = 0 \quad \forall i \in I_k$

Set: $\lambda_i = 0 \quad \forall i \notin I_k$

- If $\lambda_i \geq 0 \quad \forall i \in I_k$, by KKT sufficient conditions, x_k will be point of global minimum.
- If $\lambda_i < 0$ for some $i \in I_k$ then it can be shown that if i is dropped from I_k, the active set and (A) is solved to get d_k then d_k will be descent direction to reduce objective, i.e $\nabla^T f(x_k) \tilde{d}_k \leq 0$

(II) $d_k \neq 0$

In this case, you need to further determine α_k s.t $x_{k+1} = x_k + \alpha_k d_k$ remains feasible.

$$\alpha_k = \min\left\{1, \min_{\substack{j \notin I^k \\ a_j^T d^k < 0}} \frac{a_j^T x^k - b_j}{-a_j^T d^k}\right\}$$

Projected Subgradient

[Active set for QP with linear inequality constraints applied to dual of SVM

http://pages.cs.wisc.edu/~swright/talks/sjw-complearning.pdf slide # 17-22 & 24]

(Primal) active set method for linearly constrained QP

Step 1
Input a feasible point, $\mathbf{x}^0$, identify the active set $\mathcal{I}^0$, form matrix $A_{\mathcal{I}^0}$, and set $k = 0$.

Step 2
Compute $\mathbf{g}^k = Q\mathbf{x}^k + \mathbf{c}$.
Check the rank condition $rank[A^T_{\mathcal{I}^k} \ \ \mathbf{g}^k] = rank[A^T_{\mathcal{I}^k}]$. If it does not hold, go to **Step 4**.

Step 3
Solve the system $A^T_{\mathcal{I}^k}\widehat{\lambda} = \mathbf{g}^k$. If $\widehat{\lambda} \geq \mathbf{0}$, output $\mathbf{x}^k$ as the solution and stop; otherwise, remove the index that is associated with the most negative Lagrange multiplier (some $\widehat{\lambda}_t$) from $\mathcal{I}^k$.

Step 4
Compute the value of $\mathbf{d}^k$:

$$\mathbf{d}^k = \begin{array}{ll} \underset{\mathbf{d}}{\text{argmin}} & \frac{1}{2}\mathbf{d}^T Q\mathbf{d} + (\mathbf{g}^k)^T\mathbf{d} \\ \text{subject to} & \mathbf{a}_i^T\mathbf{d} = 0 \qquad \text{for } i \in \mathcal{I}^k \end{array} \tag{2}$$

Step 5
Compute α_k:

$$\alpha_k = \min\left\{1, \min_{\substack{j \notin \mathcal{I}^k \\ \mathbf{a}_j^T\mathbf{d}^k < 0}} \frac{\mathbf{a}_j^T\mathbf{x}^k - b_j}{-\mathbf{a}_j^T\mathbf{d}^k}\right\} \tag{3}$$

Set $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{d}^k$.

Step 6
If $\alpha_k < 1$, construct $\mathcal{I}^{k+1}$ by adding the index that yields the minimum value of α_k in (**??**). Otherwise, let $\mathcal{I}^{k+1} = \mathcal{I}^k$.

Step 7
Set $k = k + 1$ and repeat from **Step 2**.

Figure 1: Optimization for the quadratic problem in (**??**) using Primal Active-set Method.

(Kelley's) Cutting plane algos for general Convex Programs

Consider the general convex optimization problem[1]:

$$\begin{array}{lll} \text{minimize} & \mathbf{c}^T\mathbf{x} & \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 & \text{for } i = 1, 2, \ldots, m \end{array} \tag{1}$$

where $g_i(\mathbf{x})$ are convex functions.

[Recall that any convex optimation problem can be cast equivalently with linear objective and an additional convex inequality constraint]

① Let $s_j(x^i)$ be a subgradient for g_j at the point x^i

∴ By definition of subgradient

$$g_j(x) \geq g_j(x^i) + s_j^T(x^i)(x - x^i) \quad \text{- - - - (A)}$$

For example, if $s_j(x^i) = \nabla^T g_j(x^i)$

$$g_j(x) \geq g_j(x^i) + \nabla^T g_j(x^i)(x - x^i)$$

② If point x^i is feasible, ie

$$g_j(x^i) \leq 0 \quad \forall j \quad \text{- - - - (B)}$$

then by (A) and (B)

$$0 \geq g_j(x^i) + s_j^T(x^i)(x - x^i) \quad \text{- - - - (C)}$$

Note that (C), when enumerated for different values of i & j, give us several linear constraints

$i = 0 \ldots k$ [points from all previous iterations including current one]

$j = 1 \ldots m$

$$A^k = \begin{bmatrix} A_0 \\ A_1 \\ . \\ . \\ A_k \end{bmatrix} \quad \mathbf{b}^k = \begin{bmatrix} A_0\mathbf{x}^0 + \mathbf{g}_0 \\ A_1\mathbf{x}^1 + \mathbf{g}_1 \\ . \\ . \\ A_k\mathbf{x}^k + \mathbf{g}_k \end{bmatrix}$$

where,

subgradients for diff g_j's at common pt x^i

$$A_i = \begin{bmatrix} \mathbf{s}_1(\mathbf{x}^i) \\ \mathbf{s}_2(\mathbf{x}^i) \\ \cdot \\ \cdot \\ \mathbf{s}_m(\mathbf{x}^i) \end{bmatrix} \qquad \mathbf{g}_i = \begin{bmatrix} g_1(\mathbf{x}^i) \\ g_2(\mathbf{x}^i) \\ \cdot \\ \cdot \\ g_m(\mathbf{x}^i) \end{bmatrix}$$

→ function values at common pt x^i

that is: $A^k x \geq b^k$

(3) Solve the LP problem:

$$x_*^k = \underset{x}{\operatorname{argmin}} \; c^T x$$

$$\text{s.t} \quad A^k x \geq b^k$$

(4) Recall that $g_j(x) \leq 0 \Rightarrow g_j(x^i) + s_j^T(x^i)(x - x^i)$

But not vice versa

∴ Solution to LP might violate $g_j(x_*^k) \leq 0$ for some j.

↳ If so, set $k = k+1$ and go back to step (2)

↳ If no violation is found for any j, (i.e $g_j(x_*^k) \leq 0 \;\forall j$) then convergence is understood to have been achieved.

[Cutting plane algo applied to SVM primal at http://pages.cs.wisc.edu/~swright/talks/sjw-complearning.pdf, slides # 26-27]

Kelley's cutting plane algo summarised

Step 1
Input an initial feasible point, $\mathbf{x}^0$ and set $k = 0$.
Step 2
Evaluate

$$A^k = \begin{bmatrix} A_0 \\ A_1 \\ . \\ . \\ A_k \end{bmatrix} \qquad \mathbf{b}^k = \begin{bmatrix} A_0\mathbf{x}^0 + \mathbf{g}_0 \\ A_1\mathbf{x}^1 + \mathbf{g}_1 \\ . \\ . \\ A_k\mathbf{x}^k + \mathbf{g}_k \end{bmatrix} \tag{2}$$

where,

$$A_i = \begin{bmatrix} \mathbf{s}_1(\mathbf{x}^i) \\ \mathbf{s}_2(\mathbf{x}^i) \\ . \\ . \\ \mathbf{s}_m(\mathbf{x}^i) \end{bmatrix} \qquad \mathbf{g}_i = \begin{bmatrix} g_1(\mathbf{x}^i) \\ g_2(\mathbf{x}^i) \\ . \\ . \\ g_m(\mathbf{x}^i) \end{bmatrix} \tag{3}$$

where $\mathbf{s}_j(\mathbf{x}^i)$ is a subgradient of g_j at the point $\mathbf{x}^i$. Remember[a] every gradient is a subgradient.
Step 3
Solve the LP problem

$$\begin{aligned} \mathbf{x}_*^k = \quad & \underset{\mathbf{x}}{\text{argmin}} \quad \mathbf{c}^T\mathbf{x} \\ & \text{subject to} \quad A^k\mathbf{x} \geq \mathbf{b}^k \end{aligned}$$

Step 4
If $\max\{g_j(\mathbf{x}_*^k),\ 1 \leq j \leq m\} \leq \epsilon$ output $\mathbf{x}_* = \mathbf{x}_*^k$ as the point of optimality and stop. Otherwise, set $k = k + 1$, $\mathbf{x}^{k+1} = \mathbf{x}_*^k$, update A^k and $\mathbf{b}^k$ from (2) using (3) and repeat from **Step 3.**

[a]Recall that we are only dealing with convex functions.

Figure 1: Optimization for the convex problem in (1) using Kelly's cutting plane algorithm.

Examples from ML

http://www.cse.iitb.ac.in/~ CS709/notes/constrainedOpt/ConvexOptimisationForMachineLearningSlides.pdf

- Maximum likelihood estimation:

$$\underset{\theta}{\text{maximize}} \quad \sum_{i=1}^{n} \log p_\theta(x_i)$$

- Collaborative filtering:

$$\underset{w}{\text{minimize}} \quad \sum_{i \prec j} \log\left(1 + \exp(w^T x_i - w^T x_j)\right)$$

- k-means:

$$\underset{\mu_1,\ldots,\mu_k}{\text{minimize}} \quad J(\mu) = \sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu_j\|^2$$

- And more (graphical models, feature selection, active learning, control)

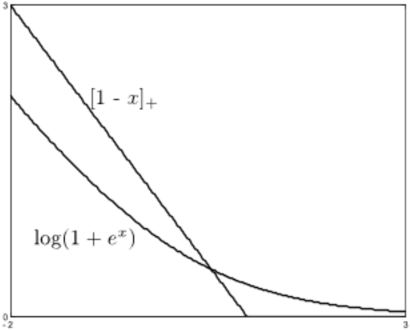
Important examples in Machine Learning

- SVM loss:

$$f(w) = \left[1 - y_i x_i^T w\right]_+$$

- Binary logistic loss:

$$f(w) = \log\left(1 + \exp(-y_i x_i^T w)\right)$$



Summary of constraint opt techniques applied to SVM

Ref: http://pages.cs.wisc.edu/~swright/talks/sjw-complearning.pdf

1. Interior point applied to SVM dual
   Slide #23 Also see http://www.cse.iitb.ac.in/~CS709/notes/SachinJayadevaGaneshSureshNeurocomputing2012.pdf
2. Projected gradient descent applied to SVM dual: Slide #18
3. Projected (stochastic) (sub)gradient descent applied to SVM primal: Slide #28-29
4. Active set and its variants applied to SVM dual: Slide #17-22 & 24.
5. Cutting plane applied to SVM primal: Slide #26-27