Optimization Methods for SVMs

- Stochastic gradient method
- Block-coordinate descent
- Active set method
Support Vector Machines
Classification SVM Problem

Given a training set of \((x_1, y_1), \ldots, (x_n, y_n)\),
\(x_i \in \mathbb{R}^d, y \in \{+1, -1\}\)

\[
\min_{\xi, w} \quad \frac{1}{2} w^\top w + c \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad y_i (w^\top x_i) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\
\xi_i \geq 0, \quad i = 1, \ldots, n.
\]
Classification SVM Problem

Given a training set of \((x_1, y_1), \ldots, (x_n, y_n)\), 
\(x_i \in \mathbb{R}^d, y \in \{+1, -1\}\)

\[
\min_{\xi, w} \quad \frac{1}{2} w^\top w + c \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad y_i(w^\top x_i) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\
\xi_i \geq 0, \quad i = 1, \ldots, n.
\]

What happened to \(\beta\)?

\[
w^\top x + \beta = (w, \beta)^\top (x, 1)
\]
Stochastic gradient approach
Unconstrained formulation of the SVM problem

Given a training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^d$, $y \in \{+1, -1\}$

$$\min_w f(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} \ell(w, (x, y))$$

where

$$\ell(w, (x, y)) = \max\{0, 1 - y_i (w^\top x_i)\}$$

Find $f(w) \leq f(w^*) + \epsilon$ - $\epsilon$-optimal solution.
Subgradient step

Consider the training set $S$ and for a given $w$ define $S^+ = \{(x, y) \in S \mid y(w^\top x) < 1\}$

$$f(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{|S|} \sum_{(x,y) \in S} \ell(w, (x,y))$$

an app, subgradient of $f(w)$:

$$\partial_w f(w) = \lambda w_t - \frac{1}{|S|} \sum_{(x,y) \in S^+} yx$$

Compute a subgradient step of length $\eta_t$.

$$w_{t+\frac{1}{2}} = w_t - \eta_t \partial_w f(w)$$

It can be shown that at optimality $\|w\| \leq 1/\sqrt{\lambda}$, hence we can project $w_{t+\frac{1}{2}}$ onto the ball to obtain $w_{t+1}$. 

SVM problem using Huber loss function

Given a training set \( S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), \( x_i \in \mathbb{R}^d \), \( y \in \{+1, -1\} \)

\[
\min_w f(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} \phi_\mu(w, (x_i, y_i))
\]

where

\[
\phi_\mu(w, (x, y)) = \begin{cases} 
0 & y(w^T x) \geq 1 \\
\frac{(y(w^T x) - 1)^2}{2\mu} & 1 - \mu < y_i(w^T x) < 1 \\
1 - y(w^T x) - \frac{\mu}{2} & y(w^T x) \leq 1 - \mu 
\end{cases}
\]

Find \( f(w) \leq f(w^*) + \epsilon \) - \( \epsilon \)-optimal solution in \( O(\frac{1}{\epsilon}) \) iterations
Approximate subgradient step

Consider a subset of the training set $A_t \subseteq S$ and for a given $w$ define $A_t^+ = \{(x, y) \in A^t : y(w^\top x) < 1\}$

$$f_t(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{|A_t|} \sum_{(x,y) \in A_t} \ell(w,(x,y))$$

an "approximate" subgradient of $f(w)$:

$$\partial_w f_t(w) = \lambda w_t - \frac{1}{|A_t|} \sum_{(x,y) \in A_t^+} yx$$

Compute a subgradient step of length $\eta_t$.

$$w_{t+\frac{1}{2}} = w_t - \eta_t \partial_w f(w, A_t)$$

It can be shown that at optimality $\|w\| \leq 1/\sqrt{\lambda}$, hence we can project $w_{t+\frac{1}{2}}$ onto the ball to obtain $w_{t+1}$. 
Stochastic Gradient Method

Choose $w_1$, such that $\|w_1\| \leq \frac{1}{\sqrt{\lambda}}$.

For $t = 1, 2, \ldots, T$

- Choose $A_t \subset S$, where $|A_t| = k$.
- Set $A_t^+ = \{(x, y) \in A_t : y(w^\top x) < 1\}$.
- $\eta_t = \frac{1}{\chi t}$
- $w_{t+\frac{1}{2}} = (1 - \eta_t \lambda)w_t + \frac{\eta_t}{k} \sum_{(x, y) \in A_t^+} yx$
- $w_{t+1} = \min\{1, \frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|}\}w_{t+\frac{1}{2}}$
Convergence in expectation

Find $E(f(\bar{w})) \leq f(w^*) + \epsilon$

$\epsilon$-optimal solution in expectation, where $\bar{w} = \frac{1}{t} \sum_{i=1}^{t} w_i$. 
Why does this work?

- Each iteration of the algorithm takes $O(n_t s)$ operations, where $s$ is the number of nonzeros attributes of each data point $x_i$ and $n_t$ is the size of $A_t$. There are no subproblems to solve.

- When $A_t = S$ and hence $n_t = n$, the algorithm takes at most $\tilde{O}(\frac{R^2}{\lambda \epsilon})$, iterations where $R = \max_i \|x_i\|$.

- When $|A_t| < n$, then we need an assumption that elements in $A_t$ a drawn from $S$ as i.i.d. samples.

- With probability $1 - \delta$ the algorithm achieves $\epsilon$-optimal solution in at most $\tilde{O}(\frac{R^2}{\delta \lambda \epsilon})$, iterations.

- This means that the probabilistic complexity of this method does not depend on the size of the training set at all!
Stochastic Approximation for Machine Learning

$$\min_w \mathbb{E}[\ell(\langle w, x \rangle, y)]$$

- Our previous approach was a mixed approach:
  - SAA: collect sample of size $m$ and minimize empirical error (w/ norm constraint):
    $$\min_{\|w\|_2 \leq B} \tilde{L}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(\langle w, x_i \rangle, y_i)$$
  - Optimize this with SGD, i.e. applying SA to the empirical objective
    - At each SGD iteration, pick random $(x,y)$ from empirical sample
  - SGD guarantee is on empirical suboptimality:
    $$\tilde{L}(\bar{w}^{(k)}) \leq \tilde{L}(\bar{w}) + O\left(\sqrt{\frac{X^2 B^2}{k}}\right)$$
  - To get guarantee on $L(w^{(k)})$, need to combined with uniform concentration:
    $$\sup_{\|w\| \leq B} \left| \tilde{L}(w) - L(w) \right| \leq O\left(\sqrt{\frac{X^2 B^2}{m}}\right)$$

- Pure SA approach:
  - Optimize $L(w)$ directly
  - Same SGD guarantee, but directly to the generalization error:
    $$L(\bar{w}^{(k)}) \leq L(w^*) + O\left(\sqrt{\frac{X^2 \|w^*\|_2^2}{k}}\right)$$
More Data ⇒ More Work?

10k training examples
10k training examples

1 hour
2.3% error
(when using the predictor)

1 week (or more...)
2.29% error

Can always sample and get same runtime:
1 hour
2.3% error

Can we leverage the excess data to reduce runtime?
10 minutes
2.3% error

But I really care about that 0.01% gain
Study runtime increase as a function of target accuracy

My problem is so hard, I have to crunch 1M examples
Study runtime increase as a function of problem difficulty (e.g. small margin)
Error Decomposition

- Approximation error:
  - Best error achievable by large-margin predictor
  - Error of population minimizer
    \[ w_0 = \text{argmin } E[f(w)] = \text{argmin } \lambda|w|^2 + E_{x,y}[\text{loss}(\langle w, x \rangle; y)] \]
- Estimation error:
  - Extra error due to replacing \( E[\text{loss}] \) with empirical loss
    \[ w^* = \text{arg min } f_n(w) \]
- Optimization error:
  - Extra error due to only optimizing to within finite precision
The Double-Edged Sword

- When data set size increases:
  - **Estimation error** decreases
  - Can increase **optimization error**, i.e. optimize to within lesser accuracy ⇒ fewer iterations
  - But handling more data is expensive
    e.g. runtime of each iteration increases

- **Stochastic Gradient Descent**, e.g. PEGASOS (Primal Efficient Sub-Gradient Solver for SVMs)
  [Shalev-Shwartz Singer Srebro, ICML’07]
  - Fixed runtime per iteration
  - Runtime to get fixed accuracy does not increase with \( n \)
Optimization Problem

\[ w^* = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad 0 \leq \alpha_i \leq c \]

\[
\min_{\alpha, \beta, \xi} \quad \frac{1}{2} \alpha^\top Q \alpha + c \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad -Q \alpha + y \beta + s_i - \xi_i = -1, \quad i = 1, \ldots, n \\
\quad s_i \geq 0, \xi \geq 0, \quad 0 \leq \alpha_i \leq c, \quad i = 1, \ldots, n,
\]

\[ Q_{ij} = y_i y_j x_i^\top x_j \quad \text{or} \quad Q_{ij} = y_i y_j K(x_i, x_j) \]

Linear formulation \hspace{1cm} Kernel formulation

\[
\min_{\alpha} \quad \frac{1}{2} \alpha^\top Q \alpha - e^\top \alpha \\
\text{s.t.} \quad y^\top \alpha = 0, \\
\quad 0 \leq \alpha \leq c,
\]
Decomposition Methods
Dual Optimization Problem

\[ \min_{\alpha} \quad \frac{1}{2} \alpha^\top Q \alpha - e^\top \alpha \]

s.t. \quad \begin{align*}
    y^\top \alpha &= 0, \\
    0 &\leq \alpha \leq c,
\end{align*}
Decomposition approach

Given any dual feasible solution, \((\alpha, \beta)\), we partition \(I = \{1, \ldots, n\}\) into \(B\) and \(N\):

- \(\forall i \in B\), \(0 < \alpha_i < c\).
- \(\forall i \in N\), \(0 \leq \alpha_i \leq c\).

\[ B \cup N = I \text{ and } B \cap N = \emptyset. \]

Based on the partition \((B, N)\) we define \(Q_{BB} (Q_{BN}, Q_{NB}, Q_{NN}) y_B (y_N)\) and \(\alpha_B (\alpha_N)\)
Active set method for convex QP

Solution of an LP is always at the vertex. In the case of QP it can be anywhere.

$$Q = \begin{bmatrix} Q_{BB} & Q_{NB} \top \\ Q_{NB} & Q_{NN} \end{bmatrix}.$$ 

Idea: temporarily fix all $\alpha_N$ to their current values and solve the reduced problem in terms of $\alpha_B$ only.

$$\min_\alpha \quad \frac{1}{2} \alpha_B \top Q_{BB} \alpha_B + e_B \top \alpha_B + \alpha_N \top Q_{NB} \alpha_B + \frac{1}{2} \alpha_N \top Q_{NN} \alpha_N - e_N \top \alpha_N$$

s.t. $y_B \top \alpha_B = -y_N \top \alpha_N,$

$0 \leq \alpha_B \leq c,$

Solve this “small” QP problem by any method
Decomposition Method

How to determine the next set $B$? Look for steepest descent direction of size $|B|$.

$$\min_d \nabla_{\alpha} f(\alpha)^\top d = \nabla_{\alpha} \left( \frac{1}{2} \alpha^\top Q\alpha - e^\top \alpha \right) d = (\alpha^\top Q - e) d$$

$$\min_d \quad (\alpha^\top Q - e) d$$
$$\text{s.t.} \quad y^\top d = 0$$
$$\quad -e \leq d \leq e$$
$$\quad d_i \leq 0 \text{ if } \alpha_i = C$$
$$\quad d_i \geq 0 \text{ if } \alpha_i = 0$$
$$\quad |\{i : d_i \neq 0\}| = |B|$$
Finding the new set B

Ordered vector

$$\nabla_\alpha f(\alpha) = Y(\alpha^\top Q - e)$$

Pick the same number of $d_i = y_i$ and $d_i = i \ y_i$, making sure that $d_i$ satisfy the conditions from prev. slide.

$$d_i = -y_i$$

$$d_i = y_i$$
Workload of a decomposition method

\[
\min_\alpha \quad \frac{1}{2} \alpha_B^\top Q_{BB} \alpha_B + e_B^\top \alpha_B + \alpha_N^\top Q_{NB} \alpha_B \\
\text{s.t.} \quad y_B^\top \alpha_B = -y_N^\top \alpha_N, \\
0 \leq \alpha_B \leq c,
\]

If using an interior point method, empirical complexity is \(O(n_B^3)\).

- Computing \(\alpha^\top Q - e\) is almost equivalent to computing \(\alpha_B^\top Q\) which is \(O(n_B n)\).

- The complexity of the second step can be reduced by ”shrinking” - considering only ”important” part of \(\alpha^\top Q - e\) vector.
Reducing the cost of finding the new set $B$

Reduce the size of the vector

$$Y(\alpha^\top Q - e)$$

by ignoring the elements that are likely to be in the middle (for example because they were in the middle last 100 iterations)

$$d_i > -y_i$$

But we do need to compute the entire vector to verify optimality!

$$d_i < -y_i$$
Complexity

Per iteration:

- Need to solve $Q_{ss}p = r$ at each iteration, where $Q_{ss}$ is $n_s \times n_s$, $n_s$ number of active support vectors for ASMs, but can be any number (2 or more) for the DMs.
- In ASMs, by updating the Cholesky of $Q_{ss}$ the work reduced to $O(n_s^2)$. For DMs have to solve each subproblem independently.
- Need to search for negative $s$ and $x_i$, $O(n_s n)$ operations.
- By considering only a small number of “promising” candidates, the work is substantially reduced.

Bound on the number of iterations

- Active set method - finite to obtain the exact solution, but could be exponential.
- Decomposition methods - $O(n^2/\epsilon)$ - not polynomial.
<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
<th>Solution Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interior point methods</td>
<td>$O(k^2n) \times O(n \log(1/\epsilon))$</td>
<td>$O(k^2n)$ in practice, very accurate solutions</td>
</tr>
<tr>
<td>Cplex, OOQP, OOPS, Mosek</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active set method</td>
<td>Exponential in theory</td>
<td>$O(n_s n^2)$ in practice, very accurate solutions</td>
</tr>
<tr>
<td>SVM-QP, Cplex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decomposition methods</td>
<td>$O(n^2/\epsilon)$, reasonably accurate solutions</td>
<td></td>
</tr>
<tr>
<td>SMO, SVM$^\text{light}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cutting plane methods</td>
<td>$O(Rns/\epsilon)$, no accurate solutions</td>
<td></td>
</tr>
<tr>
<td>SVM$^\text{perf}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stochastic Gradient</td>
<td>$O(Rs/\epsilon)$, probabilistic results, requires i.i.d samples, no accurate solutions</td>
<td></td>
</tr>
<tr>
<td>Pegasos</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Optimality Conditions

\[ \min_{\alpha} \quad \frac{1}{2} \alpha^\top Q \alpha - e^\top \alpha \]

s.t. \[ y^\top \alpha = 0, \]
\[ 0 \leq \alpha \leq c, \]

KKT conditions

\[ \alpha_i s_i = 0, \quad i = 1, \ldots, n, \]
\[ (c - \alpha_i) \xi_i = 0, \quad i = 1, \ldots, n, \]
\[ y^\top \alpha = 0, \]
\[ -Q \alpha + y \beta + s - \xi = -e, \]
\[ 0 \leq \alpha \leq c, \quad s \geq 0, \quad \xi \geq 0. \]
Active Set

Given a dual basic feasible solution, \((\alpha, \beta, s, \xi)\), we partition \(I = \{1, \ldots, n\}\) into \(I_0\), \(I_c\) and \(I_s\):

- \(\forall i \in I_0\) \(\xi_i = 0\) and \(\alpha_i = 0\), \((s_i \geq 0?)\)
- \(\forall i \in I_c\) \(s_i = 0\) and \(\alpha_i = c\), \((\xi_i \geq 0?)\)
- \(\forall i \in I_s\) \(s_i = \xi_i = 0\) and \(0 < \alpha_i < c\).

\[I_0 \cup I_c \cup I_s = I\text{ and } I_0 \cap I_c = I_c \cap I_s = I_0 \cap I_s = \emptyset.\]

Based on the partition \((I_0, I_c, I_s)\) we define \(Q_{ss}\) \((Q_{cs}, Q_{sc}, Q_{cc}, Q_{0s}, Q_{00})\), \(y_s\) \((y_c, y_0)\) and \(\alpha_s\) \((\alpha_c, \alpha_0)\)
Partitioning of matrix $Q$
Active Set Method

**Step 1**

(i) Solve

\[
\min_{\alpha_s} \quad \frac{1}{2} \alpha_s^\top Q_{ss} \alpha_s + c e^\top Q_{cs} \alpha_s - e^\top \alpha_s \\
\text{s.t.} \quad y_s^\top \alpha_s = -y_c^\top \alpha_c
\]

(ii) From the current iterate make a step toward the solution until for some \(i \in I_s\) \((\alpha_s)_i = 0\) or \((\alpha_s)_j = c\) or until solution is reached.

(iii) If for some \(i \in I_s\), \((\alpha_s)_i = 0\)

Then update \(I_s = I_s \setminus \{i\}, \ I_0 = I_0 \cup \{i\}\), and go to step (i).

(iv) If for some \(i \in I_s\), \((\alpha_s)_i = c\)

then update \(I_s = I_s \setminus \{i\}, \ I_c = I_c \cup \{i\}\), and go to step (i).

(v) If the optimum is reached in step (ii), proceed to **Step 2**.
Active Set Method

Step 2

(i) Compute $s_0$

$$s_0 = -Q_0 s \alpha_s - y_0 \beta + 1 - cQ_0 c e$$

and $\xi_c$

$$\xi_c = Q cs \alpha_s + y_c \beta - 1 + cQ c c e$$

(ii) Find $i_0 = \text{argmin}_i \{s_i : i \in I_0\}$.

Find $i_c = \text{argmin}_i \{\xi_i : i \in I_c\}$.

(iii) If $s_{i_0} \geq 0$ and $\xi_{i_c} \geq 0$ then the current solution is optimal, Exit.

If $s_{i_0} \leq \xi_{i_c}$, then $I_s = I_s \cup \{i_0\}$ and $I_0 = I_0 \setminus \{i_0\}$.

Else, $I_s = I_s \cup \{i_c\}$ and $I_c = I_c \setminus \{i_c\}$.

Go to Step 1.
Active Set Method

Step 1

(i) Solve a system with matrix

\[
\begin{bmatrix}
Q_{ss} & y \\
y^\top & 0
\end{bmatrix}.
\]

If factorization $Q_{ss} = G_s G_s^\top$ is available, then work is $O(n_s^2)$.

(ii) Step toward solution. $O(n_s)$

(iii) If for some $i \in I_s$, $(\alpha_s)_i = 0$, then update $I_s = I_s \setminus \{i\}$, $I_0 = I_0 \cup \{i\}$, update $G_s$ by removing a row. $O(n_s^2)$

(iv) If for some $i \in I_s$, $(\alpha_s)_i = c$ then update $I_s = I_s \setminus \{i\}$, $I_c = I_c \cup \{i\}$, update $e^\top Q_{cs}$ and $G_s$ by removing a row. $O(n_s^2) + O(n_c)$
Active Set Method

Step 2

(i) \[ s_0 = -Q_{0s} \alpha_s - y_0 \beta + 1 - cQ_{0c} \epsilon \]
\[ \xi_c = Q_{cs} \alpha_s + y_c \beta - 1 + cQ_{cc} \epsilon \]
\[ \mathcal{O}(n_s n) \]

(ii) Find \( i_0 = \arg \min_i \{ s_i : i \in I_0 \} \), \( i_c = \arg \min_i \{ \xi_i : i \in I_c \} \). \( \mathcal{O}(n) \)

(iii) If \( s_{i_0} \leq \xi_{i_c} \), then \( I_s = I_s \cup \{ i_0 \} \) and \( I_0 = I_0 \backslash \{ i_0 \} \).
Update \( G_s \) by adding a row
Else, \( I_s = I_s \cup \{ i_c \} \) and \( I_c = I_c \backslash \{ i_c \} \).
Update \( e^\top Q_{cs} \) and \( G_s \) by adding a row
\[ \mathcal{O}(n_s^2) + \mathcal{O}(n_c) \]
Complexity

Active set method:

- Need to solve $Q_{ss}p = r$ at each iteration, where $Q_{ss}$ is completely dense, $k_s \times k_s$.
- By updating the Cholesky of $Q_{ss}$ the work reduced to $O(k_s^2)$.
- Need to search for negative $s$ and $x_i$, $O(k_sn)$ operations.
- By considering only a small number of “promising” candidates, the work is substantially reduced.