SVM recap, optimization methods intro

Administrative/Meta:

- **New room!** Siebel 1214.
- Office hours **tomorrow**, Saturday 9/24, during 1-3pm.
- No class next Wednesday.

SVM recap/conclusion from last time

The SVM material last time was presented sloppily, so let's recap some main points before starting on convex optimization algorihtm.

First, here is a restatement of the main result from last time.

Theorem (Baby Representer Theorem from previous lecture)**.** Suppose $\lambda > 0$. Then

$$\min\left\{\sum_{i=1}^{n}\ell_i(-Aw)+\frac{\lambda}{2}\|w\|_2^2 : w\in\mathbb{R}^d\right\} = \max\left\{-\sum_{i=1}^{n}s_i-\frac{1}{2\lambda}\|A^\top s\|_2^2 : s\in[0,1]^n\times\{0\}^{m-n}\right\}.$$

Primal-dual optimal pairs $(\bar{w},\bar{s})$ always exist. $\bar{s}$ is optimal iff it has the following form:

$$\bar{s}\in\begin{cases}\{0\} & i>n,\\ \{0\} & i\le n, (A\bar{w})_i>1,\\ [0,1] & i\le n, (A\bar{w})_i=1,\\ \{1\} & i\le n, (A\bar{w})_i<1.\end{cases}$$

Lastly, $\bar{w}$ is unique, and has the form $\bar{w}=A^\top\bar{s}/\lambda$.

To prove it, we will first will mainly invoke duality theorem from two lectures ago.

Theorem (Fenchel Duality from two lectures ago)**.** Let convex $f:\mathbb{R}^d\to\mathbb{R}\cup\{\infty\}$, $g:\mathbb{R}^n\to\mathbb{R}\cup\{\infty\}$, and matrix $A:\mathbb{R}^d\to\mathbb{R}^n$ be given. Assume $\inf_x f(x)+g(Ax)>-\infty$, and the **constraint qualification**

$$0\in\mathrm{int}(\mathrm{dom}(g)-A\mathrm{dom}(f)).$$

Then

$$\inf\left\{f(x)+g(Ax) : x\in\mathbb{R}^d\right\} = \max\left\{-f^*(A^\top s)-g^*(-s) : s\in\mathbb{R}^n\right\}.$$

A pair $(\bar{x},\bar{s})$ are optimal iff $A^\top\bar{s}\in\partial f(\bar{x})$ and $-\bar{s}\in\partial g(A\bar{x})$.

We need to evaluate some conjugates and subgradients in order to massage the duality law into the baby representer theorem.

Lemma.

1. If $f:\mathbb{R}^d\to\mathbb{R}\cup\infty$ is closed convex, then f attains a minimum.

2. Given $g : \mathbb{R}^m \to \mathbb{R}$ defined as $g(v) := \sum_{i=1}^n f(v_i)$, then $g^* s(s) = \sum_{i=1}^n f^*(s_i) + \sum_{i>n} \iota_{\{0\}}(s_i)$ and $s \in \partial g(v)$ iff $s_i \in \partial f(v_i)$ for $i \leq n$ and $s_i = 0$ for $i > n$.

3. If $f(x) = g(cx)$ for $c \neq 0$, then $f^*(s) = g^*(s/c)$.

4. $\ell^*(s) = -s + \iota_{[0,1]}(s)$.

The baby representer theorem now follows by plugging all the pieces together, and additionally using the "inverse gradient" property from the last lecture that tells us $A^\top \bar{s} \in \partial f(\bar{w})$ implies $\bar{w} \in \partial f^*(A^\top \bar{s})$, namely a representation for both the primal optimum $\bar{w}$ and the dual optimum $\bar{s}$.

(Filling in the details will be a homework problem! Isn't that great!?)

Next, recall the (confusing) discussion of kernels. This can be summarized as follows. Rather than starting with data $((x_i, y_i))_{i=1}^n$ where $x_i \in \mathbb{R}^d$, suppose we merely have $x_i \in \mathcal{X}$, where $\mathcal{X}$ is just some set, and it's not clear how to take inner products, etc. Here are some options which will allow us to use $\mathcal{X}$ in an SVM.

1. We could construct a **feature map** $\Phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space; this means it can be infinite dimensional, but feel free to treat is as some $\mathbb{R}^d$. Now we can let row $i \leq n$ of matrix A be $y_i \Phi(x_i)$, and everything works. If we let row $i > n$ be $\Phi(x_i)$ where x_i is in a testing set, then our prediction on x_i is
$$(A\bar{w})_i = (AA^\top \bar{s}/\lambda)_i = \sum_{j:\bar{s}_j > 0} \frac{\bar{s}_j y_j}{\lambda} \left\langle \Phi(x_j), \Phi(x_i) \right\rangle .$$

2. Suppose we just have a pairwise similarity function $k(\cdot, \cdot)$ which is symmetric positive semi-definite (**s-psd**) over $\mathcal{X}$, meaning that for any positive integer l and any $(x_1, \ldots, x_l) \in \mathcal{X}^l$, the matrix $G \in \mathbb{R}^{l \times l}$ with $G_{ij} = k(x_i, x_j)$ is s-psd. Then we can construct G over our training sample $(x_1, \ldots, x_n)$ as before, and use that both in our algorithm (via the dual) and at prediction time; namely, we predict with
$$x \mapsto \sum_{j:\bar{s}_j > 0} \frac{\bar{s}_j y_j}{\lambda} k(x_j, x).$$

3. Suppose we are given an space (a Hilbert space) of *functions*, $\mathcal{F}$, with the following properties.
   - There exists a function $k(\cdot, \cdot)$ so that every $f \in \mathcal{F}$ satisfies $f(x) := \left\langle f, k(x, \cdot) \right\rangle$.
   - The norm on the space can be written
   $$\|f\|_{\mathcal{F}} := \inf \left\{ \|w\|_2 : f = \sum_x w_x k(x, \cdot) \right\}.$$
   (And the norm is finite for all $f \in \mathcal{F}$).

   This space is called an **RKHS** (**reproducing kernel hilbert space**).

It turns out these are all the same. As proved in (Steinwart and Christmann 2008, chap. 4) (In the following, implications mean one can be used to construct another):

- $1 \implies 2$, namely $k(x, x') := \left\langle \Phi(x), \Phi(x') \right\rangle$ is s-psd.
- $2 \implies 3$, namely an s-psd kernel k leads to the construction of an RKHS, namely the space $\text{span}(\{k(x, \cdot) : x \in \mathcal{X}\})$.
- $3 \implies 1$, namely via the definition $\Phi(x) := k(x, \cdot)$.

Remark. The notion of an RKHS leads to the true power of the representer theorem: now we write the primal optimization over the function space $\mathcal{F}$, but still we can write the optimum as a finitary object (namely $\sum_{j:\bar{s}_j > 0} \bar{s}_j k(x_j, \cdot)/\lambda$).

[note to future matus: this presentation still seemed too abstract and vague.]

Convex optimization methods

Here are 5 convex optimization scenarios that are covered in this course.

- **Projected (sub)gradient descent** applied to Lipschitz $f : \mathbb{R}^d \to \mathbb{R}$ over a compact convex constraint set; this takes $\mathcal{O}(1/\epsilon^2)$ iterations for accuracy $\epsilon > 0$. We will not prove this here; a minor variant of the sgd proof from lecture 1 does the trick.
- **Frank-Wolfe method** applied to smooth $f : \mathbb{R}^d \to \mathbb{R}$ once again with compact convex constraint. This takes $\mathcal{O}(1/\epsilon)$ iterations, and will be covered today.
- Projected gradient descent applied to smooth $f : \mathbb{R}^d \to \mathbb{R}$ with compact convex constraint. Once again $\mathcal{O}(1/\epsilon)$, discussed today.
- Projected gradient descent applied to smooth and strongly convex $f : \mathbb{R}^d \to \mathbb{R}$, unconstrained. Needs $\mathcal{O}(\ln(1/\epsilon))$ iterations, discussed today.
- **Coordinate descent** applied to smooth $f : \mathbb{R}^d \to \mathbb{R}$, unconstrained; will be analyzed under two disjoint sets of conditions of f to get rate $\mathcal{O}(\ln(1/\epsilon))$. We'll cover these in a future lecture.

Remarks.

1. To learn about more convex optimization methods, I recommend the survey by Bubeck (2014), which was used for many of the analyses here.
2. We'll be analyzing the **batch** setting, namely we should think of applying these methods to the regularized empirical risk $f(w) := \sum_i \ell(y_i, x_i, w) + p(w)$, where ℓ is a loss function and p is a penalty function (e.g., $\ell(y_i, x_i, w) = \max\{0, 1 - y_i \langle w, x_i, \} \rangle$ and $p(w) = \lambda \|w\|_2^2$.

   In order to get a guarantee over the distribution generating $((x_i, y_i))_{i=1}^n$, we must apply a **generalization bound**, as developed in the upcoming third part of the course. By contrast, sgd, as analyzed in lecture 1, directly gives a guarantee on the expected risk.
3. Recall that we defined f β-smooth to mean

$$\forall w, w' \centerdot f(w') \leq f(w) + \langle \nabla f(w), w' - w \rangle + \frac{\beta}{2} \|w - w'\|^2.$$

   Another definition is that $\|\nabla f(w) - \|\nabla f(w')\| \leq \beta \|w - w'\|$. This implies the former definition by the fundamental theorem of calculus and Cauchy-Schwarz, namely

$$\begin{aligned}
\left| f(w') - f(w) - \langle \nabla f(w), w' - w \rangle \right| &= \left| \int_0^1 \langle \nabla f(w + t(w' - w)), w' - w \rangle \, dt - \langle \nabla f(w), w' - w \rangle \right| \\
&\leq \int_0^1 \|\nabla f(w + t(w' - w)) - \nabla f(w)\| \|w' - w\| dt \\
&\leq \int_0^1 \beta \|w + t(w' - w) - \nabla f(w)\| \|w' - w\| dt \\
&= \frac{\beta}{2} \|w' - w\|^2.
\end{aligned}$$

4. There are various procedures to convert non-smooth objective functions into smooth ones. Two generic methods are as follows.
   - Rather than minimizing f directly, we minimize $w \mapsto \mathbb{E}_v(f + \tau v)$, where $\tau > 0$ is small and \$v is uniform over (the interior of) the unit ball. This can be shown to hardly change f in value, but make it Lipschitz, differentiable, and smooth. This is an old idea, but a nice recent summary is provided by (Duchi, Bartlett, and Wainwright 2012, Appendix E).

- We can form the **infimal convolution**

$$g(w) = \inf \left\{ \|v\|^2/2 + f(w - v) : w \in \mathbb{R}^d \right\}.$$

  This is a smooth function, and indeed $\text{epi}(g) = \text{epi}(v \mapsto \|v\|^2/2) + \text{epi}(f)$. For more on infimal convolution, see Hiriart-Urruty and Lemaréchal (2001), and for something recent-ish on its use in optimization, see for instance Shalev-Shwartz, Srebro, and Zhang (2010) (there are many things to cite here).

5. Each method will depend on a choice of **step size** η, or a per-iteration step η_i. The convergence properties are often sensitive to the choice of step size. In practice, one either tries many options, or uses a **line search**, meaning one tries many choices at each descent iteration and uses the one that decreases the cost the most. The line search will hold in the analysis today, as the analysis applies a per-iteration bound iteratively (so the line search per-iteration bound can't be much worse than the custom choice here). With sgd, a line search is not possible, so step sizes must be searched over.
6. In practice, many of the assumptions here will be violated: the setting will often be non-convex, unbounded or bounded so weakly that the theorems are too loose, unregularized or similarly regularized too weakly, etc. But the theory will give some good suggestions and often lead to algorithms that work well outside their analysis.

Frank-Wolfe method

We will start with the easiest analysis, namely of the Frank-Wolfe method. This algorithm approximately solves $\min_{w \in S} f(w)$ as follows.

1. Let $w_0 \in S$ be given.
2. For $i = 1, 2, \ldots, t$:
   1. Choose $v_i := \text{argmin}_{v \in S} \langle \nabla f(w_{i-1}), v \rangle$.
   2. Set $w_i := w_{i-1} + \eta_i(v_i - w_{i-1})$, where $\eta_i := 2/(i+1)$.

Remarks.

- Since $w_0 \in S$ and w_i is formed from a convex combination of w_{i-1} and $v \in S$, then induction grants $w_i \in S$ when S is convex.
- The method requires a **linear optimization subroutine** $\text{argmin}_{v \in S} \langle \nabla f(w_{i-1}), v \rangle$. This is often considered the most attractive aspect of the algorithm, and there are many settings where it can be solved much more cheaply than a projection as in projected subgradient descent.

  For instance, if $S := \text{conv}(U)$ for some finite set U, then we can evaluate the argmin by checking each element of U.
- *(Following is an answer to a question in class; thanks!)* The choice of step size has some meaning here; for instance, the more intuitive choice $\eta_i := 1/t$ (independent of i) adds a $\ln(t)$ term to the rate. To see what's going on, note that

$$w_t = 2 \sum_{i=1}^{t} \frac{i v_i}{t(t+1)} = \frac{\sum_{i=1}^t i v_i}{\sum_{i=1}^t i}.$$

  This can be proved by induction:
  - Base case $t = 1$ holds since

$$w_1 = w_0 + \eta_1(v_1 - w_0) = w_0 + (v_1 - w_0) = v_1.$$

– Inductive step $t > 1$ gives

$$w_{t+1} = (1 - \eta_{t+1})w_t + \eta_{t+1}v_{t+1} = \frac{t}{t+2}(2)\sum_{i=1}^{t}\frac{iv_i}{t(t+1)} + \frac{2v_{t+1}}{t+2}\left(\frac{t+1}{t+1}\right).$$

With that out of the way, the main things to note are that w_0 is erased completely (!), and that this weighting puts more emphasis on *later* gradients. This weighting scheme is sometimes called "polynomial-decay averaging" and has been used with sgd as well (Shamir and Zhang 2013).

Theorem. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is β-smooth, $\sup_{x,x' \in S} \|x - x'\| \leq R$, and $(w_i)_{i=0}^t$ are generated by Frank-Wolfe as above. Then for $t \geq 1$.

$$f(w_t) - f(\bar{w}) \leq \frac{2\beta R^2}{t+1}.$$

Proof. The proof proceeds by induction on iteration i; for $i > 0$, the desired bound is shown, but for $i = 0$, no bound is shown. For the inductive step,

$$\begin{aligned}
f(w_{i+1}) - f(\bar{w}) &\leq f(w_i) - f(\bar{w}) + \eta_{i+1}\left\langle \nabla f(w_i), v_{i+1} - w_i \right\rangle + \frac{\beta}{2}\|\eta_{i+1}(v - w_i)\|^2 \\
&\leq f(w_i) - f(\bar{w}) + \eta_{i+1}\min_{v \in S}\left\langle \nabla f(w_i), v - w_i \right\rangle + \frac{\beta\eta_{i+1}^2 R^2}{2} \\
&\leq f(w_i) - f(\bar{w}) + \eta_{i+1}\left\langle \nabla f(w_i), \bar{w} - w_i \right\rangle + \frac{\beta\eta_{i+1}^2 R^2}{2} \\
&\overset{(\star)}{\leq} (1 - \eta_{i+1})(f(w_i) - f(\bar{w})) + \frac{\beta\eta_{i+1}^2 R^2}{2} \\
&\overset{(\square)}{\leq} 2\beta R^2\left(\frac{i+2-2}{i+2}\left(\frac{1}{i+1}\right) + \frac{\eta_{i+1}^2}{4}\right) \\
&\leq 2\beta R^2\left(\frac{i+1}{(i+2)(i+1)}\right),
\end{aligned}$$

where $(\star)$ used convexity and $(\square)$ used the inductive hypothesis for $i > 0$ and $1 - \eta_{i+1} = 0$ when $i = 0$. □

References

Bubeck, Sébastien. 2014. "Theory of Convex Optimization for Machine Learning."

Duchi, John, Peter L. Bartlett, and Martin Wainwright. 2012. "Randomized Smoothing for Stochastic Optimization." *SIOPT*.

Hiriart-Urruty, Jean-Baptiste, and Claude Lemaréchal. 2001. *Fundamentals of Convex Analysis*. Springer Publishing Company, Incorporated.

Shalev-Shwartz, Shai, Nathan Srebro, and Tong Zhang. 2010. "Trading Accuracy for Sparsity." *SIAM Journal on Optimization* 20 (6): 2807–32.

Shamir, Ohad, and Tong Zhang. 2013. "Stochastic Gradient Descent for Non-Smooth Optimization: Convergence Results and Optimal Averaging Schemes." In *ICML*.

Steinwart, Ingo, and Andreas Christmann. 2008. *Support Vector Machines*. 1st ed. Springer.