Math ∩ Programming

Duality for the SVM

Posted on June 12, 2017 by j2kun

This post is a sequel to Formulating the Support Vector Machine Optimization Problem (https://jeremykun.com/2017/06/05/formulating-the-support-vector-machine-optimization-problem/).

The Karush-Kuhn-Tucker theorem

Generic optimization problems are hard to solve efficiently. However, optimization problems whose objective and constraints have special structure often succumb to analytic simplifications. For example, if you want to optimize a linear function subject to linear equality constraints, one can compute the Lagrangian (https://jeremykun.com/2013/11/30/lagrangians-for-the-amnesiac/) of the system and find the zeros of its gradient. More generally, optimizing a linear function subject to linear equality and inequality constraints can be solved using various so-called "linear programming (https://jeremykun.com/2014/06/02/linear-programming-and-the-most-affordable-healthy-diet-part-1/)" techniques, such as the simplex algorithm (https://jeremykun.com/2014/12/01/linear-programming-and-the-simplex-algorithm/).

However, when the objective is not linear, as is the case with SVM, things get harder. Likewise, if the constraints don't form a convex set you're (usually (http://www.offconvex.org/)) out of luck from the standpoint of analysis. You have to revert to numerical techniques and cross your fingers. Note that the set of points satisfying a collection of linear inequalities forms a convex set, provided they can all be satisfied.

We are in luck. The SVM problem can be expressed as a so-called "convex quadratic" optimization problem, meaning the objective is a quadratic function and the constraints form a convex set (are linear inequalities and equalities). There is a neat theorem that addresses such, and it's the "convex quadratic" generalization of the Lagrangian method. The result is due to Karush, Kuhn, and Tucker, (dubbed the KKT theorem) but we will state a more specific case that is directly applicable to SVM.

Theorem [Karush 1939, Kuhn-Tucker 1951]: Suppose you have an optimization problem in $\mathbb{R}^n$ of the following form:

$$\min f(x), \text{ subject to } g_i(x) \leq 0, i = 1, \dots, m$$

Where f is a differentiable function of the input variables x and $g_1, \dots, g_m$ are affine (degree-1 polynomials). Suppose z is a local minimum of f. Then there exist constants (called KKT or Lagrange multipliers) $\alpha_1, \dots, \alpha_m$ such that the following are true. Note the parenthetical labels contain many intentionally undefined terms.

1. $-\nabla f(z) = \sum_{i=1}^{m} \alpha_i \nabla g_i(z)$ (gradient of Lagrangian is zero)
2. $g_i(z) \leq 0$ for all $i = 1, \dots, m$ (primal constraints are satisfied)
3. $\alpha_i \geq 0$ for all $i = 1, \dots, m$ (dual constraints are satisfied)
4. $\alpha_i g_i(z) = 0$ for all $i = 1, \dots, m$ (complementary slackness conditions)

We'll discuss momentarily how to interpret these conditions, but first a few asides. A large chunk of the work in SVMs is converting the original, geometric problem statement, that of maximizing the margin of a linear separator, into a form suitable for this theorem. We did that last time (https://jeremykun.com/2017/06/05/formulating-the-support-vector-machine-optimization-problem/). However, the conditions of this theorem also provide the structure for a more analytic algorithm, the Sequential Minimal Optimization algorithm, which allows us to avoid numerical methods. We'll see how this works explicitly next time when we implement SMO.

You may recall (https://jeremykun.com/2013/11/30/lagrangians-for-the-amnesiac/) that for the basic Lagrangian, each constraint in the optimization problem corresponds to one Lagrangian multiplier, and hence one term of the Lagrangian. Here it's largely the same—each constraint in the SVM problem (and hence each training point) corresponds to a KKT multiplier—but in addition each KKT multiplier corresponds to a *constraint* for a new optimization problem that this theorem implicitly defines (called the dual problem). So the pseudocode of the Sequential Minimal Optimization algorithm is to start with some arbitrary separating hyperplane w, and find any training point x_j that corresponds to a violated constraint. Fix w so it works for x_j, and repeat until you can't find any more violated constraints.

Now to interpret those four conditions. The difficulty in this part of the discussion is in the notion of primal/dual problems. The "original" optimization problem is often called the "primal" problem. While a "primal problem" can be either a minimization or a maximization (and there is a corresponding KKT theorem for each) we'll use the one of the form:

$$\min f(x), \text{subject to } g_i(x) \leq 0, i = 1, \dots, m$$

Next we define a corresponding "dual" optimization problem, which is a *maximization* problem whose objective and constraints are related to the primal in a standard, but tedious-to-write-down way. In general, this dual maximization problem has the guarantee that its optimal solution (a max) is a lower bound on the optimal solution for the primal (a min). This can be useful in many settings. In the most pleasant settings, including SVM, you get an even stronger guarantee, that the optimal solutions for the primal and dual problems have *equal objective value*. That is, the bound that the dual objective provides on the primal optimum is tight. In that case, the primal and dual are two equivalent perspectives on the same problem. Solving the dual provides a solution to the primal, and vice versa.

The KKT theorem implicitly defines a dual problem, which can only possibly be clear from the statement of the theorem if you're intimately familiar with duals and Lagrangians already. This dual problem has variables $\alpha = (\alpha_1, \dots, \alpha_m)$, one entry for each constraint of the primal. For KKT, the dual constraints are simply non-negativity of the variables

$$\alpha_j \geq 0 \text{ for all } j$$

And the objective for the dual is this nasty beast

$$d(\alpha) = \inf_x L(x, \alpha)$$

where $L(x, \alpha)$ is the generalized Lagrangian (which is simpler in this writeup because the primal has no equality constraints), defined as:

$$L(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x)$$

While a proper discussion of primality and duality could fill a book, we'll have to leave it at that. If you want to journey deeper into this rabbit hole, these notes (https://people.eecs.berkeley.edu/~klein/papers/lagrange-multipliers.pdf) give a great introduction from the perspective of the classical Lagrangian, without any scarring.

But we can begin to see why the KKT conditions are the way they are. The first requires the generalized Lagrangian has gradient zero. Just like with classical Lagrangians, this means the primal objective is at a local minimum. The second requires the constraints of the primal problem to be satisfied. The third does the same for the dual constraints. The fourth is the interesting one, because it says that at an optimal solution, the primal and dual constraints are intertwined.

4. $\alpha_i g_i(z) = 0$ for all $i = 1, \dots, m$ (complementary slackness conditions)

More specifically, these "complementary slackness" conditions require that for each i, either the dual constraint $\alpha_i \geq 0$ is actually *tight* ($\alpha_i = 0$), or else the primal constraint g_i is tight. At least one of the two must be exactly at the limit (equal to zero, not strictly less than). The "product equals zero means one factor is zero" trick comes in handy here to express an OR, despite haunting generations of elementary algebra students. In terms of the SVM problem, complementary slackness translates to the fact that, for the optimal separating hyperplane w, if a data point doesn't have functional margin exactly 1, then that data point isn't a support vector. Indeed, when $\alpha_i = 0$ we'll see in the next section how that affects the corresponding training point x_i.

The nitty gritty for SVM

Now that we've recast the SVM into a form suitable for the KKT theorem, let's compute the dual and understand how these dual constraints are related to the optimal solution of the primal SVM problem.

The primal problem statement is

$$\min_w \frac{1}{2}\|w\|^2$$

Subject to the constraints that all m training points $x_1, \dots, x_m$ with training labels $y_1, \dots, y_m$ satisfy

$$(\langle w, x_i \rangle + b) \cdot y_i \geq 1$$

Which we can rewrite as

$$1 - (\langle w, x_i \rangle + b) \cdot y_i \leq 0$$

The generalized Lagrangian is

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + \sum_{j=1}^{m} \alpha_j(1 - y_j \cdot (\langle w, x_j\rangle + b))$$
$$= \frac{1}{2}\|w\|^2 + \sum_{j=1}^{m} \alpha_j - \sum_{j=1}^{m} \alpha_j y_j \cdot \langle w, x_j\rangle - \sum_{j=1}^{m} \alpha_j y_j b$$

We can compute each component of the gradient ∇L, indexed by the variables $w_i, b,$ and α_j. First, since this simplifies the Lagrangian a bit, we compute $\frac{\partial L}{\partial b}$.

$$\frac{\partial L}{\partial b} = -\sum_{j=1}^{m} y_j \alpha_j$$

The condition that the gradient is zero implies this entry is zero, i.e. $\sum_{j=1}^{m} y_j \alpha_j = 0$. In particular, and this will be a helpful reminder for next post, we could add this constraint to the dual problem formulation without changing the optimal solution, allowing us to remove the term $b\sum_{j=1}^{m} y_j \alpha_j$ from the Lagrangian since it's zero. We will use this reminder again when we implement the Sequential Minimal Optimization algorithm next time.

Next, the individual components w_i of w.

$$\frac{\partial L}{\partial w_i} = w_i - \sum_{j=1}^{m} \alpha_j y_j x_{j,i}$$

Note that $x_{i,j}$ is the i-th component of the j-th training point x_j, since this is the only part of the expression $w \cdot x_j$ that involves w_i.

Setting all these equal to zero means we require $w = \sum_{j=1}^{m} \alpha_j y_j x_j$. This is interesting! The optimality criterion, that the gradient of the Lagrangian must be zero, actually shows us how to write the optimal solution w in terms of the Lagrange multipliers α_j and the training data/labels. It also hints at the fact that, because of this complementary slackness condition, many of the α_i will turn out to be zero, and hence the optimal solution can be written as a *sparse* sum of the training examples.

And, now that we have written w in terms of the α_j, we can eliminate w in the formula for the Lagrangian and get a dual optimization objective only in terms of the α_j. Substituting (and combining the resulting two double sums whose coefficients are $\frac{1}{2}$ and -1), we get

$$L(\alpha) = \sum_{j=1}^{m} \alpha_j - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle$$

Again foreshadowing, the fact that this form only depends on the inner products of the training points will allow us to replace the standard (linear) inner product for a nonlinear "inner-product-like" function, called a *kernel*, that will allow us to introduce nonlinearity into the decision boundary.

Now back to differentiating the Lagrangian. For the remaining entries of the Lagrangian where the variable is a KKT multiplier, it coincides with the requirement that the constraints of the primal are satisfied:

$$\frac{\partial L}{\partial \alpha_j} = 1 - y_j(\langle w, x_j\rangle + b) \leq 0$$

Next, the KKT theorem says that one needs to have both feasibility of the dual:

$$\alpha_j \geq 0 \text{ for all } j$$

And finally the complementary slackness conditions,

$$\alpha_j(1 - y_j(\langle w, x_j \rangle + b)) = 0 \text{ for all } j = 1, \ldots, m$$

To be completely clear, the dual problem for the SVM is just the generalized Lagrangian:

$$\max_{\alpha}(\inf_{x} L(x, \alpha))$$

subject to the non-negativity constraints:

$$\alpha_i \geq 0$$

And the one (superfluous reminder) equality constraint

$$\sum_{j=1}^{m} y_j \alpha_j = 0$$

All of the equality constraints above (Lagrangian being zero, complementary slackness, and this reminder constraint) are consequences of the KKT theorem.

At this point, we're ready to derive and implement the Sequential Minimal Optimization Algorithm and run it on some data. We'll do that next time.

This entry was posted in General. Bookmark the permalink.

2 thoughts on "Duality for the SVM"

1. **Neil Paul**

   June 24, 2018 at 7:05 am • Reply

   Sir,
   You never got around to deriving the SMO. If its not in the works could you at least point me to some useful resources.
   Thank You.

   - **j2kun**

     June 24, 2018 at 2:43 pm • Reply

     This was a long tale for me, that eventually resulted in me forgoing further progress on this series for the sake of progress on my book. In the future I will come back to it.

     After this post I think you can read the original paper. https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/

     However, I now wonder if SMO is the best algorithm for SVM. I discovered this in a roundabout manner. I was playing with some visualizations for the next post in this series, and I discovered that a mostly random initialization of the learning parameters provided an already nearly optimal separation. This is essentially the so-called "dual perceptron" and it informs us that SVM is really just getting an extra percentage or two error improvement to get from a random solution to optimal. I stumbled upon this fantastic paper of Freund & Schapire (of boosting fame):
     http://cseweb.ucsd.edu/~yfreund/papers/LargeMarginsUsingPerceptron.pdf

In the paper they start from dual perceptron and discuss how to improve the margin so as to approach the optimum margin (though not solve for it analytically). It's much easier to implement, works nearly as well, and also works in a number of alternative computational settings (harsh memory constraints, streaming input, etc). I was a bit unsure how to proceed with the series.

Also, the SMO implementation is quite involved (and not precisely specified in the SMO paper), and my first attempt at an implementation resulted in a number of bugs I've yet to fix (source code in smo.py here: https://github.com/j2kun/svm-sequential-minimal-optimization/tree/master/smo). So with that all together, I put the series on hold while I'm working on higher priority projects.