10-725: Optimization **Fall 2012**

Lecture 22: SVMs and Introduction to Interior Point Methods

Lecturer: Geoff Gordon/Ryan Tibshirani *Scribes: Ben Eckart and Abhinav Shrivastava*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

22.1 Support Vector Machines

Support Vector Machines are one of the most popular optimization methods that people work with today. Being a supervised method, the general idea is to find a hyperplane that separates the negative and positive examples and maximize the distance from the examples to the hyperplane. This distance is called the margin (covered in last lecture). In the case when the data is not linearly separable, we want to maximize the margin plus a penalty term for points that lie on the wrong side of the margin. Put formally, our optimization problem is

$$\begin{aligned} \min_{v,d,s} \; & ||v||^2/2 + \sum_i s_i \\ \text{s.t.} \; & y_i(x_i^T v - d) \geq 1 - s_i \\ & s_i \geq 0 \end{aligned} \tag{22.1}$$

v is the normal vector to the classification surface. The norm of v is equivalent to one over the margin, so minimizing v is the same as maximizing the margin. d is the offset, y_i are the positive or negative labels for each example, and s_i are the slack variables, or penalty terms for examples not correctly classified by the margin.

We can put this equation is matrix-vector form and rewrite 22.1 as

$$\begin{aligned} \min_{v,d,s} \; & v^T v/2 + \mathbf{1}^T s \\ \text{s.t.} \; & Av - yd + s - \mathbf{1} \geq 0 \\ & s \geq 0 \end{aligned} \tag{22.2}$$

In this case, $\mathbf{1}$ is the ones vector and A is a matrix with columns equal to $y_i x_i^T$.

22.1.1 SVM Duality

In the case of SVM's, analyzing the dual let's us do a couple nice things. First, the dual helps our understanding of what it means to be a solution of the SVM problem. Second, looking at the dual gives us efficient ways to solve the problem. Thus, the dual has both a theoretical and practical advantage.

22.1.2 Taking the Dual

First, we introduce Lagrange multipliers α and t. These multipliers are vectors, with one for each constraint. We know that each constraint has to be bigger than zero, and so subtracting two positive numbers from the original objective given in eq:matrixSVM lower bounds the original objective. Likewise, the minimum of the Lagrangian provides a lower bound to the Langrangian,

$$\begin{aligned} \text{obj} &\geq v^t v/2 + \mathbf{1}^T s - \alpha^T(Av - yd + s - \mathbf{1}) - t^T s \quad (22.3)\\ &\geq \min_{v,d,s}\ v^t v/2 + \mathbf{1}^T s - \alpha^T(Av - yd + s - \mathbf{1}) - t^T s \quad (22.4)\end{aligned}$$

To obtain the dual, we carry out the minimum calculations of the Lagrangian by setting the gradient with respect to v,d,s to zero and solving.

$$\begin{aligned} \nabla_v &: 0 = v - A^T\alpha \quad (22.5)\\ \nabla_d &: 0 = \alpha^T y \quad (22.6)\\ \nabla_s &: 0 = \mathbf{1} - \alpha - t \quad (22.7)\end{aligned}$$

Equation 22.5 tells us that $v = A^T\alpha$, so we can substitute this back into the original objective,

$$\begin{aligned} \text{obj} &= \alpha^T AA^T\alpha/2 - \alpha^T AA^T\alpha + \mathbf{1}^T s + \alpha^T yd - \alpha^T s + \mathbf{1}^T\alpha - t^T s\\ &= -\alpha^T AA^T\alpha/2 + \alpha^T yd + s(\mathbf{1} - \alpha - t) + \mathbf{1}^T\alpha\end{aligned}$$

From 22.6, we know $\alpha^T y = 0$, so the second term must be zero. Similarly, we know from eq:grads that $\mathbf{1} - \alpha - t = 0$, so the third term must be zero. This leaves us with $-\alpha^T AA^T\alpha/2 + \mathbf{1}^T\alpha$. If we define K as AA^T, then the dual becomes

$$\begin{aligned} \max_{\alpha}\ & \mathbf{1}^T\alpha - \alpha^T K\alpha/2 \quad (22.8)\\ \text{s.t. }& \alpha^T y = 0\\ & 0 \leq \alpha \leq 1\end{aligned}$$

We get the final box constraint on α from the fact that $1 = \alpha + t$, from refeq:grads.

We can interpret the dual and see what it tells us about the solution. There is an α_i per example (one per constraint in the primal). Thus, we can look at them as example weights constrained to the range 0 to 1.

Complementary slackness says that if $\alpha_i > 0$, then the corresponding constraint in the primal is tight. This fact means that the example must lie right along along the margin. Similarly, if $\alpha = 1$, then it crosses the margin and has to be pushed back by the slack. In other words, if an example does not lie or cross the margin, we know that its α value is zero.

We get the name *Support Vector Machine* from the vectors that push, or support the hyperplane and have $\alpha > 0$.

From the complementary slackness condition, if $alpha < 1$, then the slack has to be zero. Thus, using the dual we can tell which points have nonzero slack and which points have nonzero weight. Furthermore, by looking at the fact that $y^T\alpha = 0$, we can see that the sum of the positive weights has to equal the sum of the negative wieghts.

22.1.3 From Dual to Primal

From our calculation of ∇_v in forming the dual, we know $v = A^T\alpha$. Rewriting this back in the original sum form and dividng through by the total sum of α_{tot},

$$\begin{aligned}\frac{v}{\alpha_{tot}} &= \frac{1}{\alpha_{tot}}\sum_i y_i x_i \alpha_i \\ &= \sum_{\text{positive}} \frac{x_i\alpha_i}{\alpha_{tot}} - \sum_{\text{negative}} \frac{x_i\alpha_i}{\alpha_{tot}}\end{aligned}$$

Looking at the above equation, we are taking a weighted mean over all positive support vectors and all negative support vectors and subtracting the two. So v (off by a scaling factor) is just the difference between the weighted means. This transformation gives us a nice idea as to what the primal and dual are finding.

22.1.4 Practical Applications of the Dual

Suppose we have n examples with m features. To make the following discussion more explicit, we shall introduce a function $\phi(u)$ that that creates features from directly observed features, u.

In the primal formulation of the SVM given by 22.2, we have $m+1$ variables in the vector v and intercept d, and n constraints (one for each example). In the dual, we have n α variables, with one equality constraint, and $2n$ box constraints.

If we want to use a large feature set while working with the primal, we may have computational problems. If, for example, we have 1 million features, it would be prohibitively expensive to invert the Hessian for a newton-based method. However, the dual doesn't depend on number of features at all. Of course, if we have a small number of features, solving the primal may be much faster.

Since we have no computational constraint on the number of features, we can use a very rich feature space with no penalty. Very large feature spaces correspond to very expressive classifiers, so by using the dual, we can sidestep the computational penalty for having a very expressive classifier.

22.1.5 The Kernel Trick

Taking the idea to the extreme, we could have a function $\phi(u)$ that makes m infinite. One small problem with this is that we need to calculate the K matrix from Equation 22.8. This matrix is composed of dot products, which would be linear in the number of features if we had to compute $\phi(u)$ directly. However, we do not need to explicitly compute $\phi(u)$ in order to compute $\phi(u_i)^T\phi(u_j)$. This is called the kernel trick, and it allows us to have potentially infinite m while still giving us computational tractability.

In other words, we don't need to know the features as long as we can compute the dot products. Re-writing K,

$$K_{ij} = x_i^T x_j = \phi(u_i)^T \phi(u_j) = k(u_i, u_j) \tag{22.9}$$

Thus, we only need a subroutine for $k(u_i, u_j)$, and so we don't need to care about ϕ. For many feature mappings ϕ, there are known subroutines for k. Any positive definite function (also known as a Mercer kernel) will work as a valid subroutine that corresponds with some potentially infinitely dimensional ϕ function.

22.1.6 Examples of Kernels

The following three kernels are commonly used:

$$K_1(u_i, u_j) = (1 + u_i^T u_j)^d \tag{22.10}$$
$$K_2(u_i, u_j) = (u_i^T u_j)^d \tag{22.11}$$
$$K_3(u_i, u_j) = e^{||u_i u_j||^2/2\sigma^2} \tag{22.12}$$

22.10 can represent any degree-d polynomial. Its decision surface is $p(u) = b$ for any degree-d polynomial p. Similarly, 22.11 represents polynomials where all terms have degree exactly d. When $d = 1$, this kernel reduces to the linear SVM. The third kernel shown in 22.12 is the Gaussian radial basis function with width σ. This kernel corresponds to an infinite feature space.

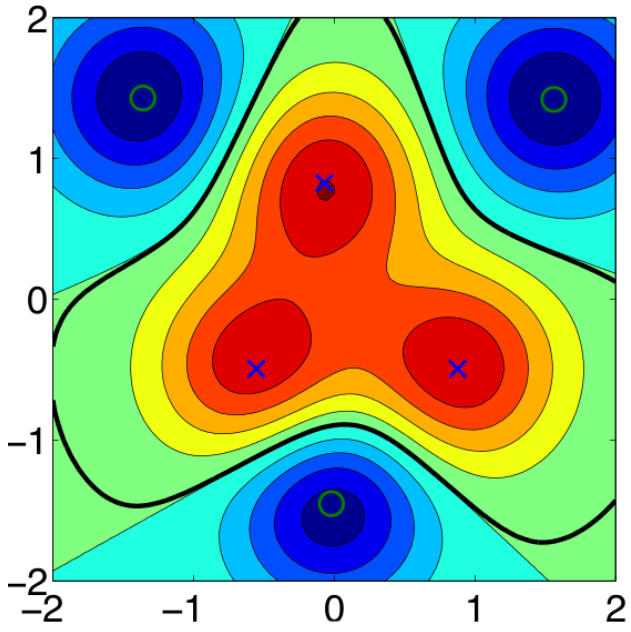


Figure 22.1: Gaussian kernel with $\sigma = 0.5$.

Figure 22.1 shows an example of a kernelized SVM using the Gaussian radial basis function. The decision surface is a linear combination of gaussian radial basis functions, one centered at each example. Every possible contour of that function is a possible decision boundary, with the optimal one (the one that maximizes the margin) in bold.

One can get a lot of very expressive decision surfaces from these kernels. In fact, the set of functions one can represent as a finite linear combinations of Gaussian radial basis functions is dense in all continuous functions. That is, any continuous function can be arbitrarily closely approximated by a sum of finitely many Gaussian radial basis functions. Any level set can be a possible decision boundary.

One may start to get worried about overfitting due to the high expressibility of the kernels. Fortunately, the maximization of the margin serves to reduce overfitting as it produces very smooth functions.

22.2 Interior-point methods

Interior-point methods are important because they are by far the most efficient methods to solve optimization problems with relatively small numbers of variables and potentially very large numbers of linear constraints (specially if we need very accurate solutions.) Also, they are based on geometric intuition and deriving the interior-point method for a problem can give interesting insights.

Historical Interest: (1) First practical polynomial time algorithm for solving an LP. (It is still an open question whether they are strongly polynomial time or not.) (2) First proposed method in literature: Ellipsoid Method. (Though it is not practical for a lot of problems, it is a great theoretical tool.)

Combinatorial or real analysis of LPs: We can either think of solving for Linear Program (LP) as being a combinatorial algorithm (for example, in case of the simplex algorithm, searching amongst discrete sets of vertices for solution) or a continuous problem (for example, maximizing a continuous function over a convex set). Interior-point methods can act as a bridge between both forms of reasoning (as we will see further in lectures.)

22.2.1 Ball Center (aka Chebyshev center)

Suppose for a problem, the feasible region (figure 22.3) is given by

$$X = \{\, x \mid Ax + b \geq 0\}. \tag{22.13}$$

In general, this is too big of a description (e.g. intersection of too many linear inequalities), and we would like to come up with a shorter description of this feasible region. One try might be to find the ball center, which is the center of the largest sphere that can be inscribed inside this feasible region (dotted blue line in figure 22.3.) As an optimization problem, we can write it as:

$$\max_{x} \min_{i} \texttt{dist}(x,\, a_i^T x + b_i = 0) \tag{22.14}$$

where $a_i^T x + b_i = 0$ is the constraint line for the i^{th} constraint, `dist` is euclidean distance of x to that constraint line, $\min_i$ finds closest constraint to the point x and $\max_x$ maximizes the distance of x to that closest constraint. Thus, (eq. 22.14) will maximize the radius of the ball that we can inscribe centered at x. We can find this ball center using a LP as follows:

1. If $\|a_i\| = 1$ (i.e. a_i is normalized), then $\texttt{dist}(x, a_i^T x + b_i = 0)$ is just $a_i^T x + b_i$ (a linear function of x). So we can re-write (eq. 22.14) as:

$$\max_{x,t} \; t \quad \text{subject to } Ax + b \geq t\,\mathbf{1} \tag{22.15}$$

$$\max_{x,t} \; t \quad \text{subject to } \mathbf{s} \geq t\,\mathbf{1} \text{ where } \mathbf{s} = Ax + b \tag{22.16}$$

2. In general, we can write the constraint as $\frac{s_i}{\|a_i\|} \geq t \; \forall \; i$. Since $\|a_i\|$ is constant, the constraint is linear in s_i and t, therefore we have a LP.

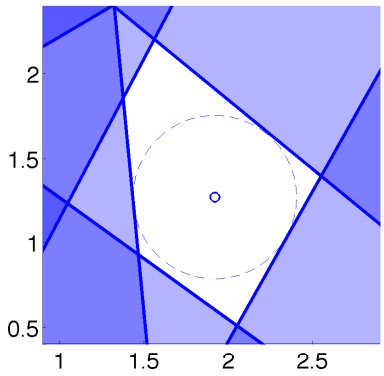
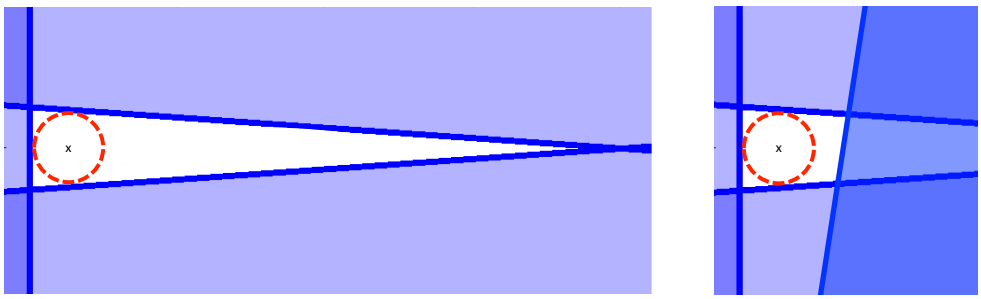


Figure 22.2: Ball Center (solid blue circle) for the feasible region (white polyhedron) (corresponding ball center in dotted blue)



(a) Feasible region with bad conditioning (b)

Figure 22.3: Ball center (dashed red circle) for a feasible region (white polyhedron) (corresponding center marked by "x".) For two very varied feasible regions given in (a) and (b), we have the same ball center.

Bad News: Consider the feasible region shown in figure 3(a) (long skinny triangle). The selected ball center is not a very good summary of the feasible region because we will have the same ball center if we had the feasible region from figure 3(b), which is indeed very different from figure 3(a). A symptom of this problem is that the ball may contain an arbitrarily small fraction of the volume of the feasible region.

22.2.2 Ellipsoid Center (aka max-volume inscribed ellipsoid)

It is the center of the inscribed ellipsoid with the largest possible volume. The ellipse centered at d is given by

$$E = \{Bu + d \mid \|u\|_2 \leq 1\} \tag{22.17}$$

i.e. a scaled and shifted version of the unit sphere. The volume of E is proportional to $\det B$, therefore 1/volume is proportional to $\det B^{-1}$. It turns out that we can find the ellipse centered at d as a solution of

the semi-definite program given by

$$\begin{aligned} &\min \ \log\det B^{-1} \\ &\text{subject to: } a_i^T(Bu+d)+b_i \geq 0 \quad \forall i \quad \forall u \text{ with } \|u\| \leq 1 \\ &\qquad B \succeq 0 \end{aligned}$$

The above constraints imply that the entire ellipse has to be contained in the feasible region. For each i and each u, we have a linear constraint in B and d (actually, there are infinitely many such constraints.) The constraints are linear, and therefore we have a convex optimization problem.

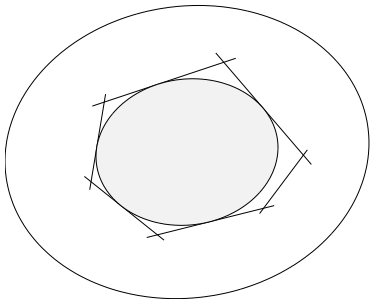


Figure 22.4: Maximum volume ellipsoid (shaded) inscribed in the feasible region (polyhedron)

As can be seen in example shown in figure 22.4, the resulting ellipsoid fills up the feasible region quite nicely. We can show that in general the $\texttt{Volume}(E) \geq \left(\frac{1}{n^n}\right)\texttt{Volume}(X)$ (E from eq. 22.17 and X from eq. 22.13) i.e. at least a polynomial fraction of the feasible region is covered by the ellipsoid.

In conclusion, we have a convex optimization problem to find a good summary of the feasible region. But the problem is very expensive, with infinitely many constraints in an SDP (but we can still do it.)

So, the next obvious question: Is there some summary of the feasible region which can be computed cheaply, and which can be said to be a good summary? (e.g. if it contains at least a polynomial fraction of the volume of the feasible region, like ellipsoid center discussed above.) Find the answer in the next section!

22.2.3 Analytic Center

Take the slack $\mathbf{s} = Ax + b \Leftrightarrow s_i = a_i^T x + b_i$ from eq. 22.16. We can define analytic center as:

$$\max_{x,s>=0} \prod_i \frac{s_i}{\|a_i\|} \tag{22.18}$$

This is similar to maximizing the minimum distance (from eq. 22.14, see general formulation) as if we have a small distance, then the above product is going to be small. On the other hand, the only way to maximize the product is to make all the slacks (s_i) approximately equal or at least some subset of slacks approximately equal and others bigger than that (can be seen as a soft version of the ball center.) Since

$-\log$ is anti-monotone function, therefore we can rewrite eq. 22.18 as

$$\max_{x,s>=0} \prod_i \frac{s_i}{\|a_i\|} \tag{22.19}$$

$$\Leftrightarrow \max_{x,s>=0} -\sum_i \log \frac{s_i}{\|a_i\|} \tag{22.20}$$

$$\Leftrightarrow \max_{x,s>=0} -\sum_i \log s_i + \sum_i log\|a_i\| \tag{22.21}$$

$$\Leftrightarrow \max_{x,s>=0} -\sum_i \log s_i \tag{22.22}$$

since $\|a_i\|$ is constant. Thus, minimizing the sum of negative logs of the slacks is same as maximizing the product in eq. 22.18. Therefore, the definition analytic center of a feasible region is given by eq. 22.22 (x in figure 22.5 is the analytic center.)

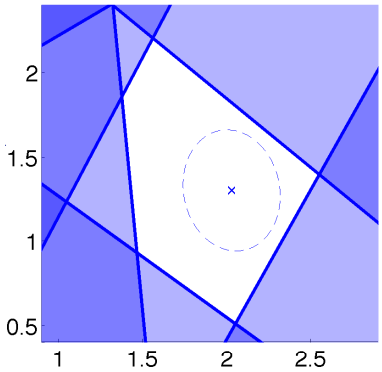


Figure 22.5: Analytic Center (blue "x") for the feasible region (white polyhedron) (corresponding Dikin Ellipsoid in dotted blue)

It turns out that we can define an ellipse, centered at the analytic center, that contains at least the polynomial fraction of the volume of the feasible region (for e.g. dikin ellipsoid, more on this in next lecture.). More concretely, for n dimensional problem with m constraints, we at least cover a factor of $(1/m^n)$ of the feasible region $\left(\texttt{Volume}(\text{Dikin Ellipsoid}) \geq \left(\frac{1}{m^n}\right) \texttt{Volume}(X)\right)$(the proof is in book). Therefore, we have a similar polynomial fraction guarantee as in ellipsoid center (though weaker). But how much weaker? In n dimensions, we need at least $m = n + 1$ constraints to make a closed polyhedron. So, $m > n \Leftrightarrow (1/m^n) < (1/n^n)$. Recall from previous section that $\texttt{Volume}$(Ellipsoid Center) $\geq \left(\frac{1}{n^n}\right) \texttt{Volume}(X)$, therefore we have $\texttt{Volume}$(Dikin Ellipsoid) $\leq \texttt{Volume}$(Ellipsoid Center) i.e. the ellipsoid center's volume has to be at least as large as analytic center's ellipsoid volume (but both are polynomial fraction of the total volume of the feasible region)

Invariance: The analytic center is invariant to scaling of the rows of A (scaling will only change the constant that we dropped.) i.e. it is invariant to individual scaling of the constraints (a_i).

Comparison to Ball Center Unlike the ball center, analytic center will be effected by every constraint, even the inactive ones, as the product/sum has every constraint (though far away constraints will have less effect.) That means that the analytic center is not the analytic center of the feasible region, it is just the center of this particular representation of the feasible region, as we could move an inactive constraint back and forth (a bit) and it will change the analytic center.

So why analytic centers? Some cool results in next lecture!

Feasible region with bad conditioning: Analytic relieves the problem of bad conditioning by stretching

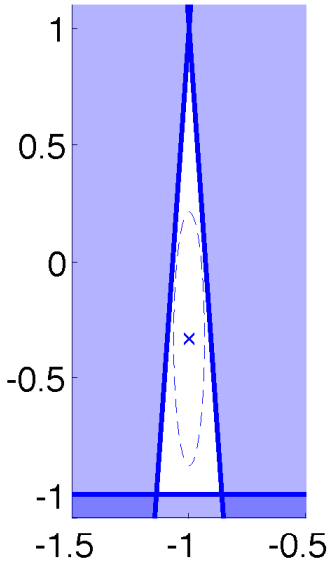


Figure 22.6: Example of badly conditioned feasible region (white polyhedron). Analytic Center (blue "x") for the feasible region (corresponding Dikin Ellipsoid in dotted blue)

out the ellipse a lot (as shown in figure 22.6) and we still get a decent volume of the feasible region. And in fact, we can show that not only analytic center is invariant to individual scaling of the constraints, but also any affine scaling of the feasible region. That is, we can shift and scale our feasible region, and the analytic center will shift and scale in exactly the same way, and the ellipsoid (dikin ellipsoid) will shift and scale the same way as well.

22.2.3.1 Newton's method for Analytic Center

The sum-of-logs in eq. 22.22 is very smooth and infinitely differentiable function, we know that Newton's method might be a good way to find the analytic center. From eq. 22.22 we have

Objective: $f(x) = -\sum_i \log(a_i^T x + b_i) = \sum_i \log(s_i)$ where $\mathbf{s} = Ax + b$

Gradient: $\frac{df(x)}{dx} = -\sum_i \frac{a_i}{a_i^T x + b_i} = -\sum_i a_i \frac{1}{s_i} = -A^T \left(\frac{1}{s}\right)$ where $(1/s)$ is component-wise.

Hessian: $\frac{d^2 f(x)}{d^2 x} = \sum_i \frac{a_i a_i^T}{(a_i^T x + b_i)^2} = A^T S^{-2} A$ where $S = \texttt{diag}(\mathbf{s})$

Using this gradient and hessian, therefore we can run Newton's method on $f(x)$. Because, our objective is strictly convex function, the hessian is strictly positive definite and hence the Newton's direction is actually a descent direction. This implies that the Newton's method converges from any feasible initializer (as long as we use line search).

But we still have to fine a strictly feasible initializer, where the objective is finite (see next lecture to get around this requirement). It turns out that the Newton's method will converge globally in a very small number of iterations.

Adding an Objective

Analytic center was for feasibility problem like finding x such that $Ax + b \geq 0$. Instead of this, now we want to find solution to $\min c^T x$ subject to $Ax + b \geq 0$.

We can use the same trick and solve

$$\min f_t(x) = c^T x - \left(\frac{1}{t}\right) \sum_i \log(a_i^T x + b_i)$$

where $t > 0$, is the trade-off between two terms. As $t \to 0$, $\left(\frac{1}{t}\right) \to \infty$ and the log part of $f_t(x)$ dominates, and we have the solution for analytic center. As $t \to \infty$, $\left(\frac{1}{t}\right) \to 0$ and the log part vanishes, thus we just get the LP optimal. The center path is smooth (infinitely differentiable for any finite t) and so the central path smoothly connects the analytic center to the LP optimal (more details on this in next lecture.)