

Support Vector Machines

Patricia Hoffman, PhD.

June 3, 2010

Contents

I	Support Vector Machine Concepts	3
1	Review of Generalized Learning Algorithms	3
2	Intuition for Support Vector Machines	5
3	Support Vector Machines: Mathematical Derivation	7
3.1	Exploration of the Geometry	7
3.2	Solving Optimization Problems with Lagrange Multipliers . . .	9
3.3	Lagrange Duality and the Karush-Kuhn-Tucker Conditions . .	13
3.4	Soft Margin	16
3.5	Kernels	20
II	Support Vector Machine Implementation	22
4	Insight into Kernels	22
5	The Sequential Minimal Optimization Algorithm	26
III	Classification using Support Vector Machines	27
6	Procedure for Classifying Data	27
6.1	Data Preprocessing	27
6.2	Complexity vs Error	27

7	Examples	29
7.1	Handwriting Example	29
7.2	Protein Example TBD	29
A	General Form of the Quadratic Optimization Problem	30
	Appendix	30
	References	32

List of Figures

1	Two Gaussian Distributions	4
2	Maximum Margin Found with Support Vector Machine	4
3	Other separations of the data	6
4	Data which is not Linearly Separable	6
5	Illustration of $\ \omega\ \gamma^{(i)}$	8
6	The Surface given by $f(x, y) = x^2 + 2y^2$	10
7	Contour plot: $f(x, y) = x^2 + 2y^2$ - Graph: $g(x, y) = x^2 + y^2 = 1$	11
8	Three non-colinear points in a plane shattered by a line	17
9	Decision Boundaries for four points	17
10	Soft Margin	17
11	Linear Kernel	23
12	Polynomial Kernel	24
13	Radial Basis Function Kernel	25
14	Complexity vs. Error	28
15	Example of Digits to Classify	29

Part I

Support Vector Machine Concepts

There are many Machine Learning techniques that can be used to classify data.¹ Four in particular are Generalized Learning Algorithms, Decision Trees, Neural Nets, and of course Support Vector Machines. Vladimir Vapnik and his colleagues developed Support Vector Machines while at AT & T. Support Vector Machines (SVM) have been shown to compete well with Neural Nets in classifying handwritten digits.

Professor Andrew Ng claims that SVMs are among the best "off-the-shelf" supervised learning algorithms. Professor Jerome Friedman prefers Decision Trees. He claims that SVM's work very well for homogeneous data (similar to pixels in an image), however if the features measure diverse quantities, Decision Trees shine.

1 Review of Generalized Learning Algorithms

For Generalized Linear Models the data is modeled by assuming its distribution is a member of the exponential family whose parameters are found by maximizing the log of the likelihood function. Gaussian Discriminant Analysis is a specific example.

Pictorially the Gaussian Discriminant (Fig 1)² calculates Gaussian distributions fit to the two classes of training sets. The figure then shows a straight line giving the decision boundary at which the probability of $y = 1$ given \mathbf{x} is one half ($\mathbf{p}(y = 1|\mathbf{x}) = 0.5$). On one side of the boundary $y = 1$ is predicted as the most likely outcome, while on the other side $y = 0$ is predicted.

Remember that Generalized Learning Algorithms depend on their assumptions. The data is assumed fit the distribution that is chosen. If the assumptions are not met, there will be classification errors.

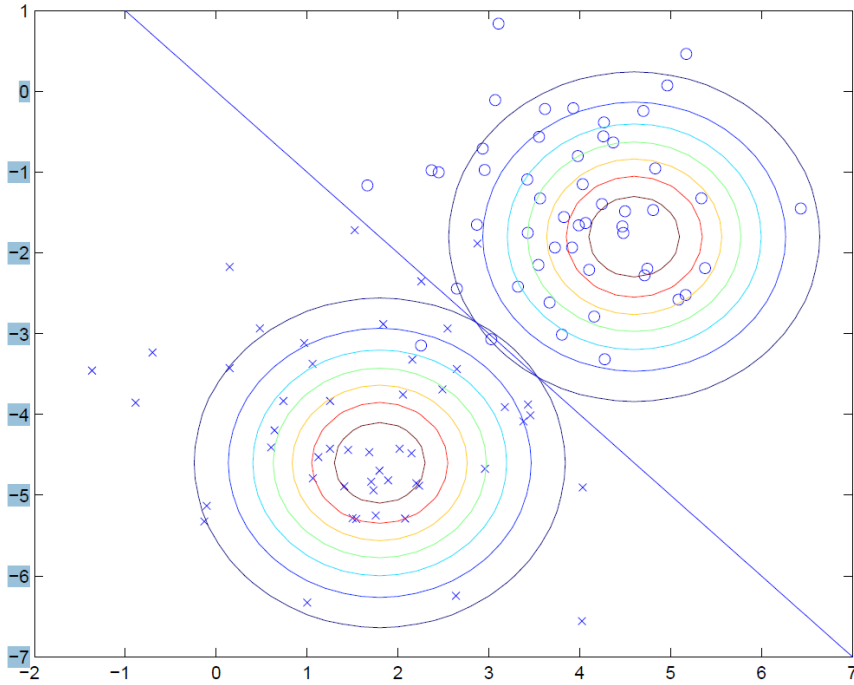


Figure 1: Two Gaussian Distributions

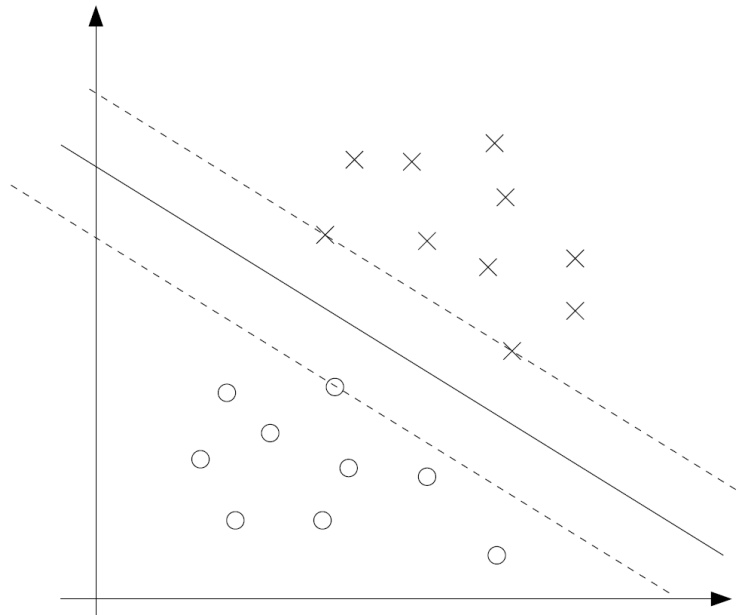


Figure 2: Maximum Margin Found with Support Vector Machine

2 Intuition for Support Vector Machines

Instead of modeling the data, the SVM models the boundary between the data (Fig 2). The idea of SVM is to find the boundary so that the distance to the nearest data point and this boundary is maximized. This makes the maximum sized margin. Although the figure illustrates a two dimensions problem in which this boundary is a line, the SVM algorithm does works in multiple dimensions. That is, the SVM algorithm finds the "optimal separating hyperplane" given the training data. If the data, as in this figure, can be separated by a hyperplane the problem is linearly separable. The points on the margin of the boundary are called the support vectors. In this figure there are two x 's and one o which are support vectors. The boundary is called the discriminant and the equation for it is given by

$$\omega^T \mathbf{x} + \mathbf{b} = 0 \quad (1)$$

Once the SVM determines ω and \mathbf{b} from the training data, an arbitrary data point \mathbf{x} can be categorized by determining which side of the line it is on.

In Figure 2 as the data is linearly separable it is possible to find the discriminant using the easiest formulation of the SVM. There are other possible separations for the data as shown in Figure 3, but the SVM algorithm will find the separation which maximizes the margin. If the data is not linearly separable (Fig 4) there are various versions of the SVM algorithm used to separate the categories. One standard method used in the non-linearly separable case is the Soft Margin technique in which weighting factors are included in the SVM calculation. By regularizing the SVM optimization problem, outlying training data points are allowed to be misclassified. Adding a kernel, can create non-linear boundaries. Kernels can also map the problem into a higher dimensional space in which case the problem may be linearly separable. The development of the SVM algorithm starts with the easy case in which the data is linearly separable. Each training observations for SVM is denoted as a pair, $\{\mathbf{y}^{(i)}, \mathbf{x}^{(i)}\}$ where the $\mathbf{y}^{(i)}$ is the category and the $\mathbf{x}^{(i)}$ are the factors or measurements. However, for SVMs the two categories are $+1$ and -1 . Thus each $\mathbf{y}^{(i)}$ is either equal to $+1$ or -1 . In (Fig 2), for the x 's: $\omega^T \mathbf{x} + \mathbf{b} \geq +1$ holds, while for the o 's: $\omega^T \mathbf{x} + \mathbf{b} \leq -1$ holds. Hence for every observation the following is true:

$$\mathbf{y}^{(i)}(\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq +1 \quad (2)$$

¹See <http://patriciahoffmanphd.com/machinelearning.php>

²Figure reproduced from Prof. Andrew Ng's Lecture notes on Generative Algorithms <http://www.stanford.edu/class/cs229/materials.html>

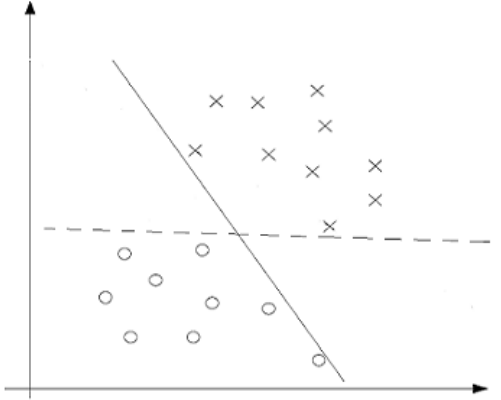


Figure 3: Other separations of the data

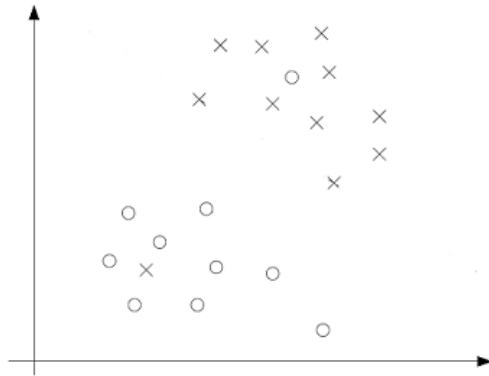


Figure 4: Data which is not Linearly Separable

For notation, there are m training samples and the number of features is equal to the dimension of the vector $\mathbf{x}^{(i)}$ which is n . Restated, there are m pairs, $\{\mathbf{y}^{(i)}, \mathbf{x}^{(i)}\}$, in which each vector $\mathbf{x}^{(i)} \in \mathbb{R}^n$ and each $\mathbf{y}^{(i)}$ is ± 1 .

The SVM algorithm finds the discriminant. The goal is to state the problem as a standard quadratic optimization, as there are many solutions to this type of problem. In addition, in this form the determination of the SVM model parameters corresponds to a convex optimization problem, hence any local solution is also automatically a global solution. In fact, as will be shown, the problem for the linearly separable case can be stated as finding

$$\begin{aligned} \min \left\{ \frac{1}{2} \|\omega\|^2 \right\} \\ \text{subject to } \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq +1 \text{ for all } i \end{aligned} \quad (3)$$

The SVM algorithm then expresses ω and \mathbf{b} in terms of the training data $\{\mathbf{y}^{(i)}, \mathbf{x}^{(i)}\}$.

3 Support Vector Machines: Mathematical Derivation

The geometry of the problem is used to derive the mathematical formula for the Support Vector Machine in the linearly separable case. The mathematical formula for the SVM is shown to be a special case of a standard quadratic optimization problem. By using the Lagrangian the dual problem is constructed. In the dual form, the problem is expressed in terms of a simple inner product. This can be expressed by using the kernel which is the identity matrix. Other kernels can replace the identity matrix to solve more complicated categorization problems.

3.1 Exploration of the Geometry

The first step is to calculate the distance between a specific training data point and the discriminant.

Lemma 1. *The formula for the distance between $\mathbf{x}^{(i)}$ and the discriminant $\omega^T \mathbf{x} + \mathbf{b} = 0$ is given by:*

$$\|\omega\| \gamma^{(i)} = |\omega^T \mathbf{x}^{(i)} + \mathbf{b}| = \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \quad (4)$$

where this distance is denoted as $\gamma^{(i)}$.

Proof. The geometry is illustrated in Figure 5. In the figure the distance between A and B is given as $\|\omega\| \gamma^{(i)}$ where A corresponds to an arbitrary $\mathbf{x}^{(i)}$. The first step is to show that ω is perpendicular to the discriminant. Let \mathbf{x}_1 and \mathbf{x}_2 be two points on the line $\omega^T \mathbf{x} + \mathbf{b} = 0$. This implies that $\omega^T \mathbf{x}_1 + \mathbf{b} = \omega^T \mathbf{x}_2 + \mathbf{b} = 0$. Hence, $\omega^T \mathbf{x}_1 = \omega^T \mathbf{x}_2$. To show that ω is perpendicular to the discriminant it is only necessary to show that ω is perpendicular to the line segment from \mathbf{x}_1 and \mathbf{x}_2 . Remember that if the dot product of two vectors is zero then the vectors are perpendicular. Here as,

$$\omega \bullet (\mathbf{x}_2 - \mathbf{x}_1) = \omega^T (\mathbf{x}_2 - \mathbf{x}_1) = \omega^T \mathbf{x}_2 - \omega^T \mathbf{x}_1 = 0 \quad (5)$$

For the second part notice that $\frac{\omega}{\|\omega\|}$ is a unit vector perpendicular to the discriminant and $\gamma^{(i)}$ is the distance between $\mathbf{x}^{(i)}$ and the line. Hence

$$\mathbf{x}^{(i)} - \mathbf{y}^{(i)} \gamma^{(i)} \left(\frac{\omega}{\|\omega\|} \right) \quad (6)$$

is a point on the discriminate line: $\omega^T \mathbf{x} + \mathbf{b} = 0$. As it is on the line it must satisfy the equation for the line, hence

$$\omega^T \left(\mathbf{x}^{(i)} - \mathbf{y}^{(i)} \gamma^{(i)} \left(\frac{\omega}{\|\omega\|} \right) \right) + \mathbf{b} = 0 \quad (7)$$

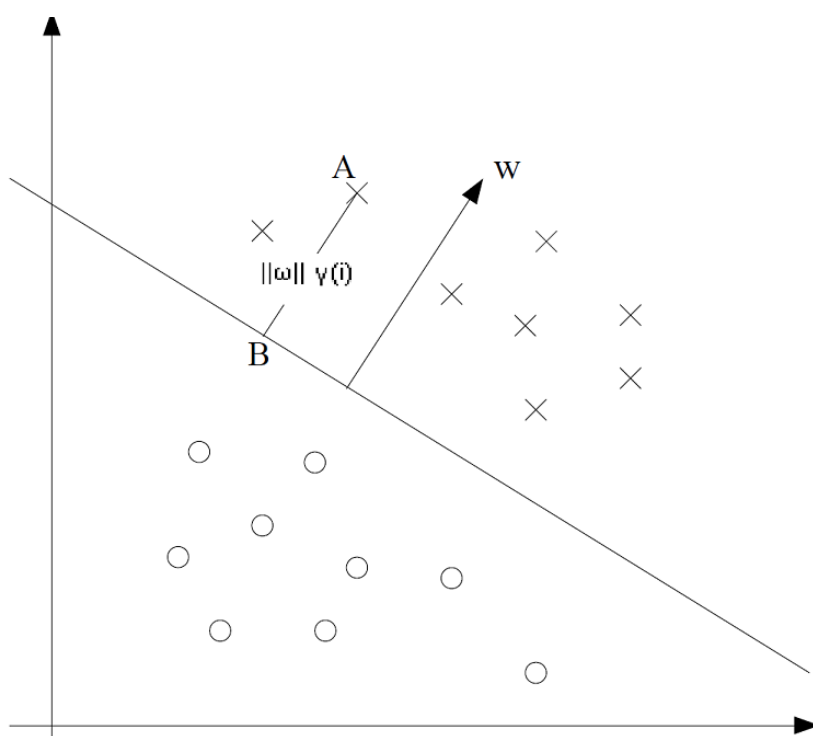


Figure 5: Illustration of $\|\omega\| \gamma^{(i)}$

The last step implies the desired result.

$$\begin{aligned}
 \omega^T \left((\mathbf{x}^{(i)} - \mathbf{y}^{(i)} \gamma^{(i)} \left(\frac{\omega}{\|\omega\|} \right)) \right) + \mathbf{b} &= 0 \\
 \Leftrightarrow \omega^T \mathbf{x}^{(i)} - \omega^T \mathbf{y}^{(i)} \gamma^{(i)} \left(\frac{\omega}{\|\omega\|} \right) + \mathbf{b} &= 0 \\
 \Leftrightarrow \mathbf{y}^{(i)} \gamma^{(i)} \omega^T \left(\frac{\omega}{\|\omega\|} \right) &= \omega^T \mathbf{x}^{(i)} + \mathbf{b} \\
 \Leftrightarrow \|\omega\| \gamma^{(i)} &= |\omega^T \mathbf{x}^{(i)} + \mathbf{b}| = \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \tag{8}
 \end{aligned}$$

First ω^T is distributed through and then terms are moved around the equal sign. Remember that $\omega^T \omega = \|\omega\|^2$ and $(\frac{\omega}{\|\omega\|})^T \frac{\omega}{\|\omega\|} = 1$. Equation (2) shows that the absolute value of $\omega^T \mathbf{x}^{(i)} + \mathbf{b}$ is equal to $\mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b})$. As equation (8) is the same as equation (4), the proof is complete. \square

The problem is to maximize γ subject to the condition that $\gamma \leq \gamma^{(i)}$. Notice that $\omega^T \mathbf{x} + \mathbf{b} = 0$ creates the same discriminate as $2\omega^T \mathbf{x} + 2\mathbf{b} = 0$. To create a unique solution the value of $\gamma \|\omega\|$ must be fixed. With $\gamma \|\omega\| = 1$ maximizing the margin is the same as minimizing $\|\omega\|$. The goal has been achieved as the problem is now expressed in the standard Quadratic Optimization (QP) form described in equation (3)

$$\begin{aligned}
 \min \left\{ \frac{1}{2} \|\omega\|^2 \right\} \\
 \text{subject to } \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq +1 \text{ for all } i
 \end{aligned}$$

The fact that this is in QP form is shown in the Appendix Section A.

3.2 Solving Optimization Problems with Lagrange Multipliers

The general optimization problem can be stated as given two functions $f : \mathbb{R}^m \rightarrow \mathbb{R}$ and $g : \mathbb{R}^m \rightarrow \mathbb{R}$, find the optimum of f given the constraint $g = k$, where k is some constant.

To motivate the discussion, start with a simple function of two variables. For an example, consider the function, $f(x, y) = x^2 + 2y^2$ which describes the surface pictured in Figure 6. Next consider a constraint of the form $g(x, y) = k$. In this example, $g(x, y) = x^2 + y^2 = 1$, which is plotted along with a contour plot of $f(x, y)$ in Figure 7. Specifically, the contours for $f(x, y) = c$ where $c \in \{.5, 1, 1.5, 2\}$ are plotted in the figure. In this example the problem can be stated as find the extreme values of the function

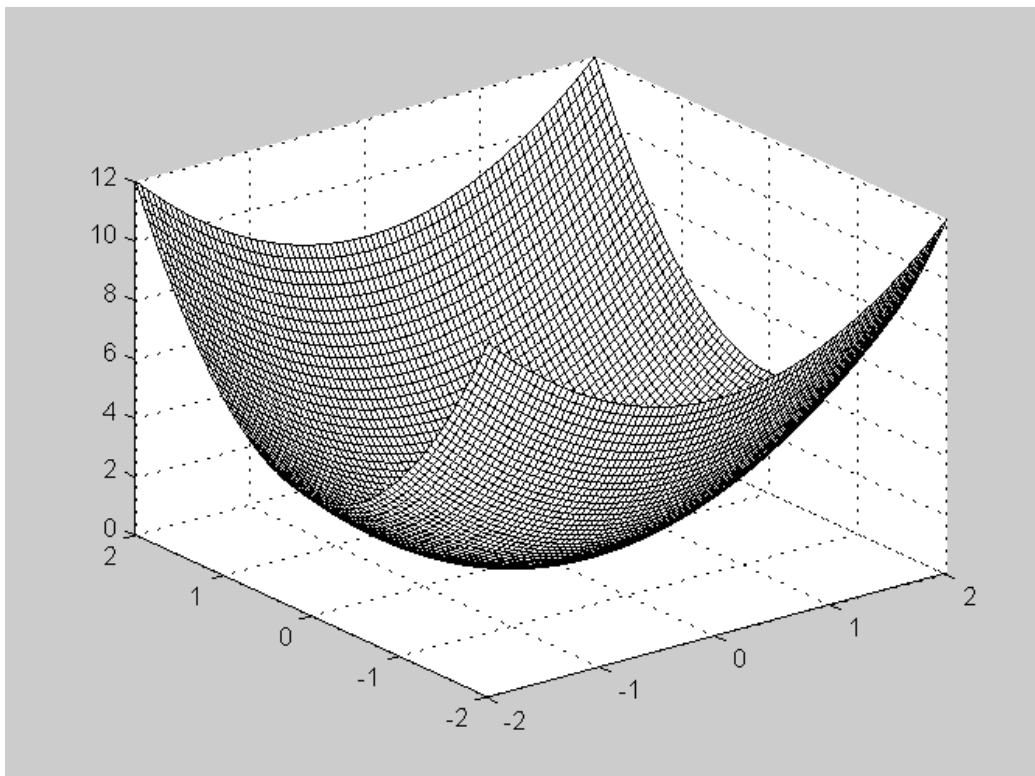


Figure 6: The Surface given by $f(x, y) = x^2 + 2y^2$

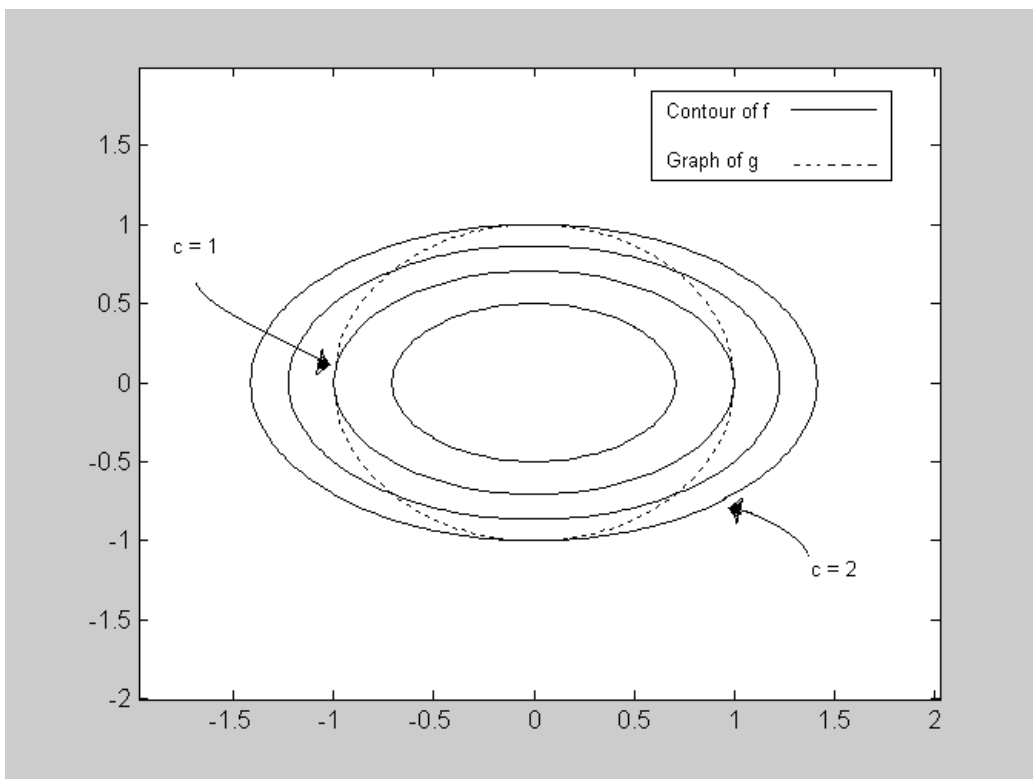


Figure 7: Contour plot: $f(x, y) = x^2 + 2y^2$ - Graph: $g(x, y) = x^2 + y^2 = 1$

$f(x, y) = x^2 + 2y^2$ given the constraint $g(x, y) = x^2 + y^2 = 1$. By looking at the contour plot, the problem is to find the largest value c such that the level curve $f(x, y) = c$ intersects with $g(x, y) = 1$ (and similarly for the smallest c).

As you can see from the figure, the c that is desired is on the curves which just touch each other. That is, when these curves have a common tangent line. Thus, the normal lines to these curves at the point (x_0, y_0) where they touch are identical. This can only happen if the gradient vectors are parallel. If two vectors are parallel then one is a multiple of the other. Hence there must exist some scalar λ such that $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$. This λ is called the Lagrange Multiplier.

Apply this to the specific example:

$$\nabla f(x_0, y_0) = (2x_0, 4y_0) = \lambda \nabla g(x_0, y_0) = (\lambda 2x_0, \lambda 2y_0) \quad (9)$$

Now there are three equations in three unknowns:

$$2x_0 = \lambda 2x_0 \quad (10)$$

$$4y_0 = \lambda 2y_0 \quad (11)$$

$$\text{and } g(x_0, y_0) = x_0^2 + y_0^2 = 1 \quad (12)$$

In equation (10) either $\lambda = 1$ or $x_0 = 0$. Similarly in equation (11) either $\lambda = 2$ or $y_0 = 0$. So there are two cases to explore: $\lambda = 1$ and $\lambda = 2$.

Case $\lambda = 1$ requires $y_0 = 0$ and by equation (12) this implies that $x_0 = +1$ or $x_0 = -1$. Evaluation of f at these points results in $f(1, 0) = 1$ and $f(-1, 0) = 1$

Case $\lambda = 2$ requires $x_0 = 0$ and by equation (12) this implies that $y_0 = +1$ or $y_0 = -1$. Evaluation of f at these points results in $f(0, 1) = 2$ and $f(0, -1) = 2$

By solving for λ the maximum value of f subject to the constraint $g = 1$ is 2 while the minimum value with the same constraint is 1.

The same argument is used in the more general case: Find the extreme values of $f : \mathbb{R}^m \rightarrow \mathbb{R}$ subject to the constraint $g : \mathbb{R}^m \rightarrow \mathbb{R}$. Instead of level curves, level surfaces are considered. The tangent vectors (and hence gradient vectors) must be parallel. Let $\mathbf{x} = (x_1, \dots, x_m) \in \mathbb{R}^m$ represent points in the domain of f . Suppose f has an extreme value at a specific point $P = \mathbf{x}_0$ on the surface S and let C be a curve with vector equation $\mathbf{r} = \mathbf{x}(t)$ that lies on S and passes through P . If t_0 is the parameter value corresponding to the point P , then $\mathbf{r}(t_0) = \mathbf{x}_0$. The composite function $h(t) = f(\mathbf{r}(t))$ represents the values that f takes on the curve C . Since f has an extreme value at \mathbf{x}_0 , it follows that h has an extreme value at t_0 , so $h'(t_0) = 0$. But if f is

differentiable, by use of the chain rule

$$0 = h'(t_0) = f_{x_1}(\mathbf{x}_0)x'_1(t_0) + \dots + f_{x_m}(\mathbf{x}_0)x'_m(t_0) = \nabla f(\mathbf{x}_0) \bullet r'(t_0) \quad (13)$$

This shows that the gradient vector $\nabla f(\mathbf{x}_0)$ is orthogonal to the tangent vector $r'(t_0)$ for every such curve C . Similarly the gradient vector of g , $\nabla g(\mathbf{x}_0)$ is also orthogonal to $r'(t_0)$, (the gradient vector is perpendicular to the tangent plane to the level surface and in particular it is perpendicular to the tangent vector of any curve lying on that surface). Hence the gradient vectors $\nabla f(\mathbf{x}_0)$ and $\nabla g(\mathbf{x}_0)$ must be parallel. This implies that if $\nabla g(\mathbf{x}_0) \neq 0$ there is a number λ such that

$$\nabla f(\mathbf{x}_0) + \lambda \nabla g(\mathbf{x}_0) = 0 \quad (14)$$

Before continuing, note that any constraint $h(\mathbf{x}) = k$ can be rewritten as $g(\mathbf{x}) = h(\mathbf{x}) - k = 0$, which is the general form.

The Lagrangian function \mathcal{L} is defined by

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x}) \quad (15)$$

Setting $\nabla_{\mathbf{x}} \mathcal{L} = 0$ provides the stationary constraint (equation (14)), and setting $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$ provides the constraint $g(\mathbf{x}) = 0$

3.3 Lagrange Duality and the Karush-Kuhn-Tucker Conditions

Recall that finding the boundary in the Support Vector Machine is a constrained optimization problem. So, consider a problem of the form

$$\begin{aligned} \min_{\omega} f(\omega) \\ \text{subject to } h_i(\omega) = 0 \text{ for all } i = 1, \dots, l \end{aligned}$$

In this case the Lagrangian is

$$\mathcal{L}(\omega, \beta) = f(\omega) + \sum_{i=1}^l \beta_i h_i(\omega) \quad (16)$$

The β_i 's are the Lagrange multipliers. From the previous section the first step is to find the partial derivatives of the Lagrangian, \mathcal{L} , set them equal to zero

$$\frac{\partial \mathcal{L}}{\partial \omega_i} = 0; \quad \frac{\partial \mathcal{L}}{\partial \beta_i} = 0 \quad (17)$$

and then solve for ω and β .

However in the problem of finding the boundary in the Support Vector Machine the constraint has not only an equality but also inequalities. Next consider the optimization problem which includes inequality constraints. This is called the primal optimization problem.

$$\begin{aligned} \min_{\omega} f(\omega) \\ \text{subject to } g_i(\omega) \leq 0 \text{ for all } i = 1, \dots, k \\ \text{and } h_i(\omega) = 0 \text{ for all } i = 1, \dots, l \end{aligned}$$

There are two cases for the solution which depend on whether the constrained stationary point lies in the region where $g_i(\omega) < 0$ (the inactive case) or whether the constrained stationary point lies on the boundary $g_i(\omega) = 0$ in which case it is called an active. In the inactive case, the function $g_i(\omega)$ plays no role and so the stationary condition can be satisfied with $\alpha_i = 0$. The active case is analogous to the equality constraint discussed previously and corresponds to a stationary point of the Lagrangian as discussed before with $\alpha_i \neq 0$. Now the sign of the Lagrange multiplier is crucial, because the function $f(\omega)$ will only be at a minimum if its gradient is oriented away from the region $g_i(\omega) < 0$. Hence, all the α_i 's must be greater than or equal to zero.

For either of these cases $\alpha_i g_i(\omega) = 0$. The problem of minimizing $f(\omega)$ subject to the constraints $g_i(\omega) \leq 0$ and $h_i(\omega) = 0$ is obtained by optimizing the Lagrange function with respect to ω , α , and β subject to these conditions. The Lagrangian in this case is

$$\mathcal{L}(\omega, \alpha, \beta) = f(\omega) + \sum_{i=1}^k \alpha_i g_i(\omega) + \sum_{i=1}^l \beta_i h_i(\omega) \quad (18)$$

Putting this all together, results in the Karush-Kuhn-Tucker (KKT) conditions which are given as

$$\text{(Lagrangian Stationarity)} \quad \frac{\partial}{\partial \omega_i} \mathcal{L}(\omega, \alpha, \beta) = 0, \quad i = 1, \dots, n \quad (19)$$

$$\frac{\partial}{\partial \beta_i} \mathcal{L}(\omega, \alpha, \beta) = 0, \quad i = 1, \dots, l \quad (20)$$

$$\text{(Complementary Slackness)} \quad \alpha_i g_i(\omega) = 0, \quad i = 1, \dots, k \quad (21)$$

$$\text{(Primal Feasibility)} \quad g_i(\omega) \leq 0, \quad i = 1, \dots, k \quad (22)$$

$$\text{(Dual Feasibility)} \quad \alpha_i \geq 0, \quad i = 1, \dots, k \quad (23)$$

If (ω, α, β) is a solution to these equations then it is also a solution to the primal and dual problems (under certain convexity assumptions ³). ω is referred to as the primal variable of the Lagrangian, while the α 's and β 's are known as the dual variables of the Lagrangian or also as the Lagrangian Multipliers. For a detailed discussion of Lagrange duality, see Chapter 5 of the text by Boyd and Vandenberghe [BV04] ⁴

The dual problem for the SVM is easier to solve and has many useful properties that can be used in developing an algorithm to find the solution.

As a summary, the strategy is to form the primal optimization problem, derive the dual optimization problem, and then solve the dual problem.

³These conditions include: 1) The convex constrained problem is of the OPT form, with differentiable convex functions f and g , and affine functions h . (ie. recall convex \Leftrightarrow Hessian is positive semi definite, affine implies linear) 2) the g_i 's are strictly feasible and 3) more

⁴This text is online at <http://www.stanford.edu/~boyd/cvxbook/>

3.4 Soft Margin

So far, only the linearly separable case has been discussed. The non-linearly separable case can be addressed by three options; The dimension of the data can be increased, a non-linear kernel can be applied, or the optimization equation can be regularized to create a soft margin.

Increasing the dimension of the data and changing the shape of the discriminant are discussed first. Vapnik-Chervonenkis (VC) dimension is the cardinality of the largest set of points that the algorithm can shatter. In other words the VC dimension is the maximum number of points that can be arranged so that the machine can still categorize them correctly, no matter how they are geometrically arranged. The concept of the VC dimension is illustrated in two dimensional pictures. For example (Fig 8), a single straight line can separate any three points in a plain that are not collinear in all possible ways. However there are examples of four points in a plane (Fig 9) which cannot be shattered by a simple line. The examples in Figure 8 are linearly separable, where as the example in Figure 9 is not. An SVM can shatter a set of m input data points if and only if every possible training set of m points can be classified exactly correctly with no training error. In the two dimensional case, the VC dimension of a linear SVM machine is three. In general the VC dimension of a linear classifier for the m dimensional case is $m+1$. The points in Figure 9 can be separated correctly by a curve. By changing the kernel of the SVM, the boundary can become a curve. Another method is to add more dimensions to the data which also can be accomplished by changing the kernel. For an unrealistic example, consider the right most portion of Figure 9. Let the o 's have the first dimension be -1 , while give the x 's a $+1$ in the first dimension. The data in this case can be separated by the plane $\{(x_1, x_2, x_3) \text{ such that } x_1 = 0\}$. This is an example of mapping the data into a higher dimensional feature space which increases the likelihood of the data to be linearly separable. In order to make this particular mapping, the classification of the point would have to known. Kernels can also be used to map the data into a higher dimensional space.

A better solution is given by regularization; A soft margin can be created which allows outlying data to be incorrectly classified. In Figure 10, if the highest o was not on the figure, the best discriminate would be the dashed line. A single outlier dramatically shifts the decision boundary, which may not be desired. Increasing the complexity of the boundary will result in an exact separation of the training data, but can lead to over fitting. By adding a penalty weight that increases with the distance from the boundary, data points are allowed to be misclassified. This leads to a better generalize boundary. Here, slack variables, $\xi_i \geq 0$ are introduced. For each observation,

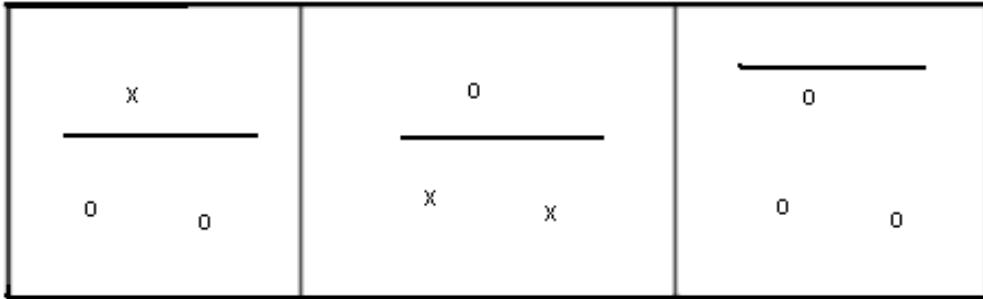


Figure 8: Three non-colinear points in a plane shattered by a line

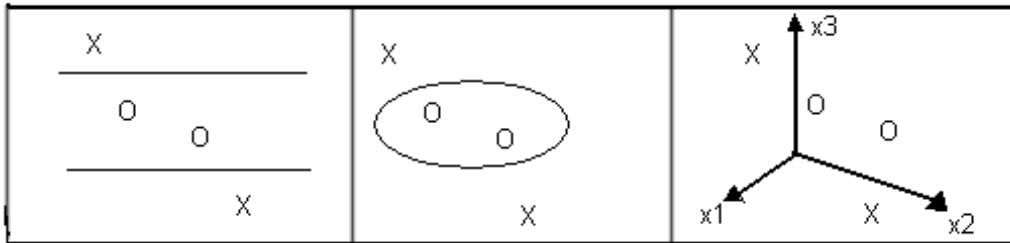


Figure 9: Decision Boundaries for four points

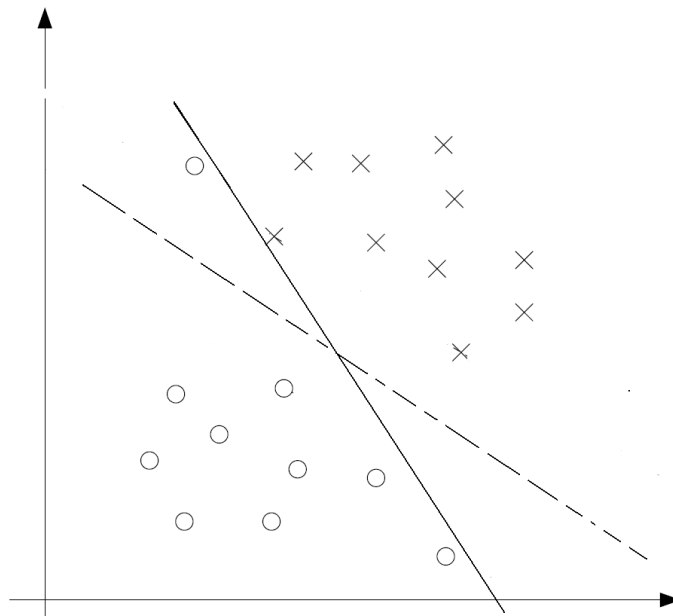


Figure 10: Soft Margin

$\{y^{(i)}, x^{(i)}\}$, the weight (ξ_i) is defined. If $x^{(i)}$ is correctly classified then $\xi_i = 0$, otherwise

$$\xi_i = |y^{(i)} - (\omega^T x^{(i)} + \mathbf{b})|. \quad (24)$$

Notice here that if $\{y^{(i)}, x^{(i)}\}$ is on the decision boundary, then $\xi_i = 1$. (Recall that if \mathbf{x} is on the boundary then $\omega^T \mathbf{x} + \mathbf{b} = 0$). If $\xi_i > 1$, this point is misclassified. The goal is to maximize the margin while "softly" penalizing points that lie on the wrong side of the margin boundary. Also, a parameter $C > 0$ is introduced to control the trade-off between the slack variable penalty and maximizing the margin. Recall that the original optimization problem was given as

$$\begin{aligned} & \min \left\{ \frac{1}{2} \|\omega\|^2 \right\} \\ & \text{subject to } \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq +1 \text{ for all } i \end{aligned} \quad (3)$$

By introduction of an l_1 norm penalty regularization (the norm is given by the absolute value), the optimization problem is expressed as

$$\begin{aligned} & \min \left\{ \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \xi_i \right\} \\ & \text{subject to } \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \text{ for all } i \end{aligned} \quad (25)$$

Notice that if $C = 0$, this equation is the same as the original equation for the linearly separable case. When C is small training errors are minimized. Once again form the Lagrangian by adding multipliers α_i and r_i to produce

$$\mathcal{L}(\omega, b, \xi, \alpha, r) = \frac{1}{2} \omega^T \omega + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i [\mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + b) - 1 + \xi_i] - \sum_{i=1}^m r_i \xi_i \quad (26)$$

Setting the derivatives with respect to ω , b , and the ξ_i s equal to zero results in

$$\frac{\partial \mathcal{L}}{\partial \omega} = 0 \Rightarrow \omega = \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} \mathbf{x}^{(i)} \quad (27)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} = 0 \quad (28)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow \alpha_i = C - r^{(i)} \quad (29)$$

Notice that by using the derivative equations (27 through 29), the third term of equation (26) can be written as

$$\omega^T \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} \mathbf{x}^{(i)} + b \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} - \sum_{i=1}^m \alpha_i + \sum_{i=1}^m \alpha_i \xi_i = \omega^T \omega + (b)(0) - \sum_{i=1}^m \alpha_i + \sum_{i=1}^m (C - r_i) \xi_i \quad (30)$$

The first term of (30) adds directly with the first term of (26). The last term of equation (30) cancels the second and last terms of equation (26).

Hence, by substituting the constraints given in the derivative equations back into equation (26) and simplifying the dual Lagrangian is produced in the form of

$$\begin{aligned} \tilde{\mathcal{L}}(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j \mathbf{y}^{(i)} \mathbf{y}^{(j)} \mathbf{x}^{(i)T} \mathbf{x}^{(j)} \\ &\text{with constraints } \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} = 0 \\ &\text{and } 0 \leq \alpha_i \leq C \text{ for } i = 1, \dots, m \end{aligned} \quad (31)$$

The constraint, $0 \leq \alpha_i \leq C$, is called the box constraint. All of the α_i 's and ξ_i 's must be positive as they are Lagrange multipliers. The box constraint occurs because equation (29) must also be true.

Recall that the Karush-Kuhn-Tucker conditions are the necessary conditions for the solution to be the optimum. In this case the KKT dual complementarity conditions are expressed as

$$\alpha_i = 0 \Rightarrow \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + b) \geq 1 \quad (32)$$

$$\alpha_i = C \Rightarrow \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + b) \leq 1 \quad (33)$$

$$0 < \alpha_i < C \Rightarrow \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + b) = 1 \quad (34)$$

These KKT conditions are important as they will be used in testing the convergence of the Sequential Minimal Optimization algorithm, which will be discussed later.

The dual form of the optimization is expressed as

$$\begin{aligned} \max_{\alpha} \mathbf{W}(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \mathbf{y}^{(i)} \mathbf{y}^{(j)} \alpha_i \alpha_j \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle \\ &\text{with constraints } \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} = 0 \\ &\text{and } 0 \leq \alpha_i \leq C \text{ for } i = 1, \dots, m \end{aligned} \quad (35)$$

The subset of the data points which have $\alpha_i = 0$, do not contribute to the predictive formula. The few data points in which $\alpha_i > 0$ are the support vectors. Given a point \mathbf{x} in the factor space, the formula to predict its category is given by

$$\omega^T \mathbf{x} + b \tag{36}$$

By substituting the value of ω given by equation (27), the predictions can be made by calculating

$$\sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} \mathbf{x}^{(i)T} \mathbf{x} + b = \sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} \langle \mathbf{x}^{(i)}, \mathbf{x} \rangle + b \tag{37}$$

The constant b can be calculated in a numerically stable manner by starting with the last KKT condition given by equation (34). Let \mathcal{M} = the set of indices of the data points having $0 < \alpha_i < C$. Let $N_{\mathcal{M}}$ denote the number of indices in this set. For each i in this set equation (34) holds. That is for each $i \in \mathcal{M}$ the following holds: $\omega^T \mathbf{x}^{(i)} + b = \mathbf{y}^{(i)} \Rightarrow b = \mathbf{y}^{(i)} - \omega^T \mathbf{x}^{(i)}$. Hence, b can be calculated by taking the average given by

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{i \in \mathcal{M}} \left(\mathbf{y}^{(i)} - \sum_{j \in \mathcal{M}} \alpha_j \mathbf{y}^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle \right) \tag{38}$$

Recall that as equation (35), equation (37), and equation (38) only depend on \mathbf{x} as the inner product ($\langle \mathbf{x}^{(i)}, \mathbf{x} \rangle$). By adding the feature mapping ($\phi(\mathbf{x})$) this inner product can be replaced by the corresponding Kernel, ($\mathbf{K}(\mathbf{x}^{(i)}, \mathbf{x})$) in each of these equations.

3.5 Kernels

For the sake of this section, the original variables \mathbf{x} input into the SVM are called attributes. A mapping from these attributes into a new set of quantities called features is denoted as ϕ . As an example

$$\phi(\mathbf{x}) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

Recall that equation (35) only depends on the inner product of the input attributes. This inner product can be replaced by a Kernel corresponding to the feature mapping ϕ defined as

$$K(x, z) = \phi(x)^T \phi(z) \quad (39)$$

A more complicated kernel is given by

$$K(x, z) = (x^T z)^2 \quad (40)$$

In this case, if $x = (x_1, x_2)$ then

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_2 x_1 \\ x_2 x_2 \end{bmatrix}$$

A necessary and sufficient condition for K to be a valid (Mercer) kernel is that K must be a symmetric positive semi-definite matrix. If K is a Mercer kernel then the mapping $K(x, z) = xKz$ where xKz is just the matrix multiplication can replace the inner product $\langle x, z \rangle$ in equation (35).

Part II

Support Vector Machine Implementation

the book is [BV04] this section describes the implementation details Complex Optimization Overview part 1 ... off-the-shelf software packages include CVX, Sedumi, CPLEX, MOSEK etc ...

4 Insight into Kernels

There are four basic kernels that are currently in use. The linear kernel in which K is just the identity matrix and the result is just the regular inner product. As a summary, the four most common kernels with parameters γ , r , and d are given as

- Linear Kernel: $K(x, z) = x^T z$
- Polynomial Kernel: $K(x, z) = (\gamma x^T z + r)^d, \gamma > 0$
- Radial Basis Function Kernel: $K(x, z) = \exp(-\gamma \|x - z\|^2), \gamma > 0$
- Sigmoid: $K(x, z) = \tanh(\gamma x^T z + r)$

The Gaussian Kernel is a special case of the Radial Basis Function (RBF) kernel. The Gaussian Kernel is given as

$$K(x, z) = \exp\left(\frac{-\|x - z\|^2}{2\sigma^2}\right)$$

For general rules to select the kernel parameters see Section 6.2 Complexity vs Error. Hsu, Chang, and Lin [HCL10] suggest considering the RBF kernel first. The exception to this rule of thumb is when the number of observations for training is much less than the number of features in an observation. In this case linear kernels work just as well.

Figures ⁵ 11, 12, and 13 show examples of the linear kernel, polynomial kernel, and radial basis function kernels. Notice how the shape of the bound-

⁵Figure created using an Applet developed by Hakan Serce <http://www.eee.metu.edu.tr/~alatan/Courses/Demo/AppletSVM.html>

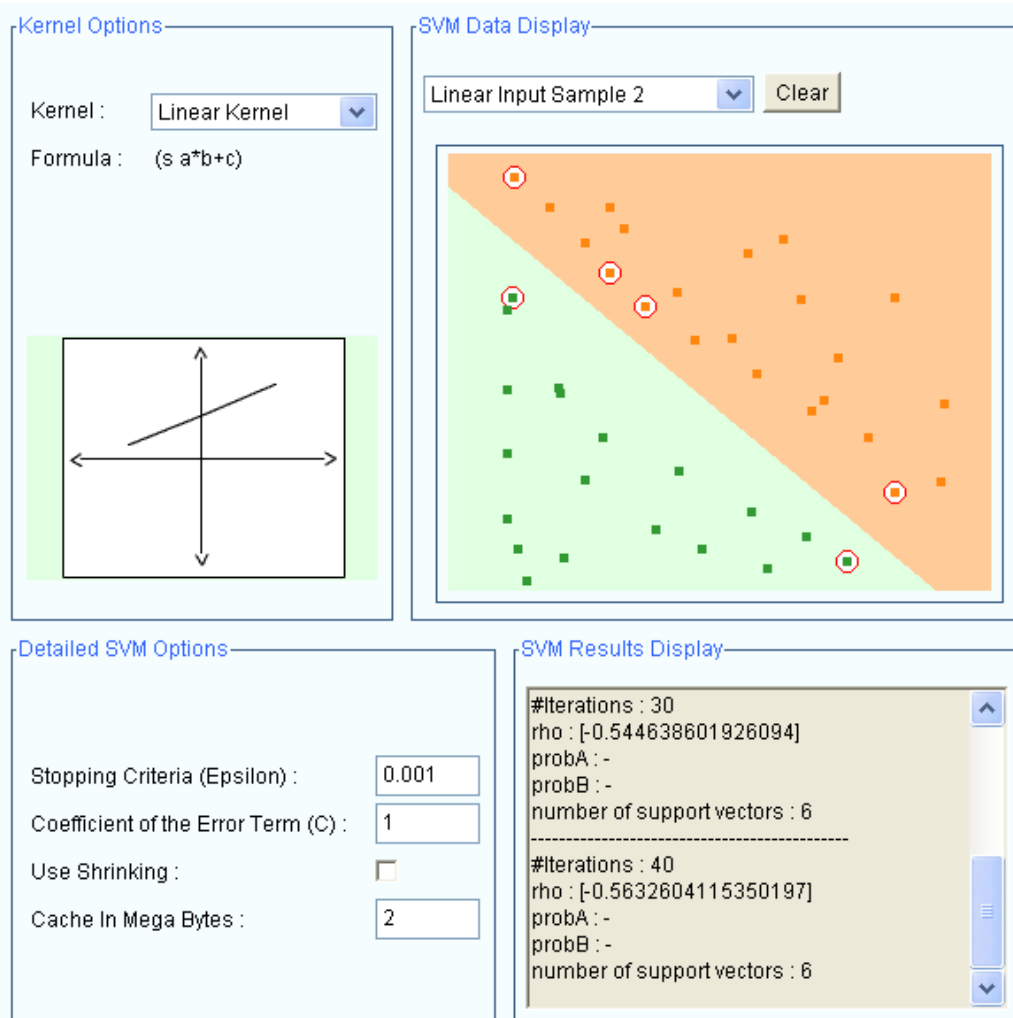


Figure 11: Linear Kernel



Figure 12: Polynomial Kernel

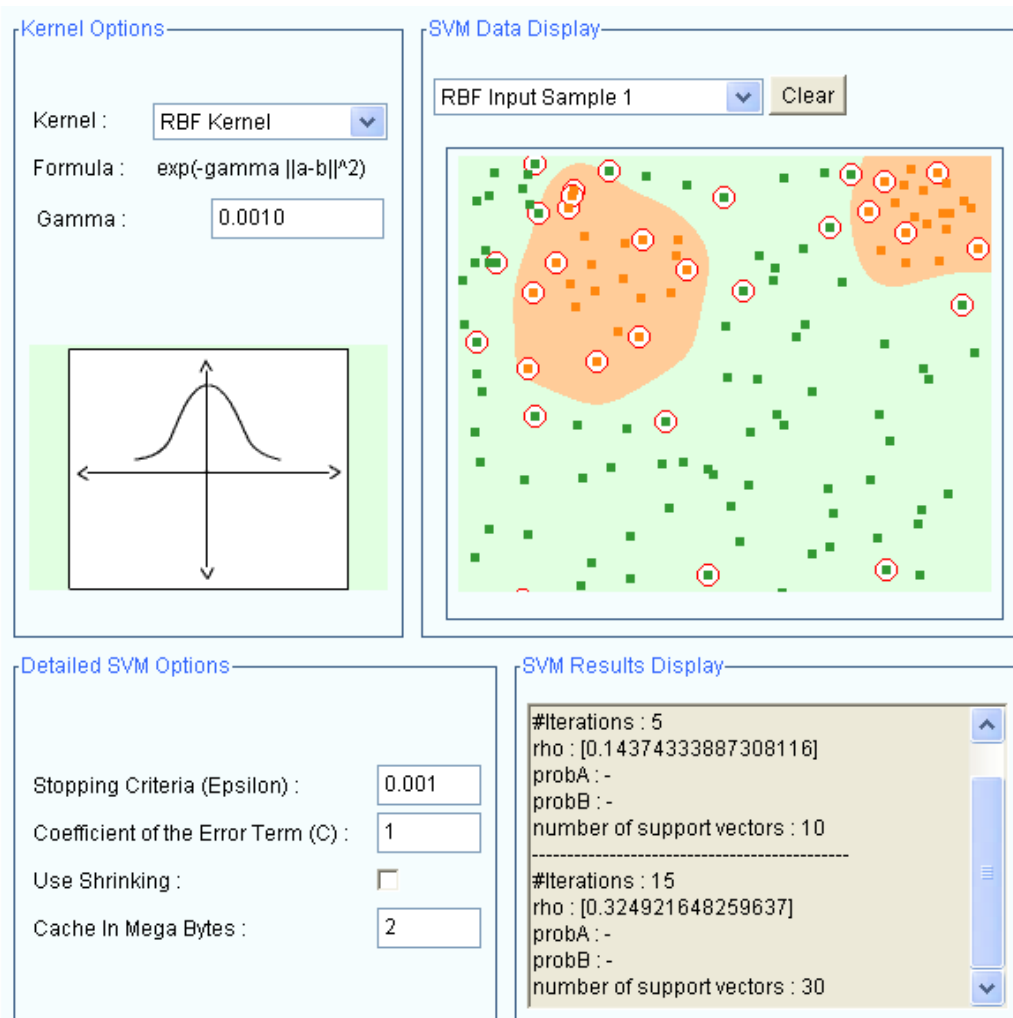


Figure 13: Radial Basis Function Kernel

ary changes with the various kernels. In each case the support vectors are circled in red.

5 The Sequential Minimal Optimization Algorithm

TBD This section is just notes so far ... It is true that there are many quadratic programming algorithms and software that can be used to solve the SVM problem. However a quick, efficient, and popular method to solve the SVM problem is called the Sequential Minimal Optimization (SMO) Algorithm proposed by Platt.[Pla07] ⁶

The SMO algorithm by Platt goes here

Talk about Coordinate Ascent TBD Coordinate Ascent goes here could use Figure 7 to talk about it

Talk about Pseudo Code TBD The boundary is defined as

$$f(\mathbf{x}) = \sum_{i=1}^m y_i \alpha_i k(x, x_i) + b \quad (41)$$

The dual form of the optimization is expressed as

$$\begin{aligned} \max_{\alpha} \mathbf{W}(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \mathbf{y}^{(i)} \mathbf{y}^{(j)} \alpha_i \alpha_j \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle \\ \text{with constraints} \quad &\sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} = 0 \\ &\text{and } 0 \leq \alpha_i \leq C \text{ for } i = 1, \dots, m \end{aligned} \quad (35)$$

The problem is to find the α 's. Because of the sum constraint

$$\sum_{i=1}^m \alpha_i \mathbf{y}^{(i)} = 0 \quad (42)$$

two α 's must be changed at the same time. The pseudo code uses the the box constraint, $0 \leq \alpha_i \leq C$, to find a pair to update. This pair α_p and α_q is also selected so that one is below its margin boundary while the other is above its margin boundary. The pseudo code finds η and adds it to one of the α and subtracts it from the other α . This keeps the sum constraint constant. η is selected to maximize \mathbf{W} .

⁶See <http://research.microsoft.com/apps/pubs/default.aspx?id=68391>

Placement of Pseudo Code

Part III

Classification using Support Vector Machines

6 Procedure for Classifying Data

6.1 Data Preprocessing

TBD this is just notes so far ... Preprocessing of the Data to encode categorical attributes for example represent colors as red, yellow, blue (0,0,1), (0,1,0), (1,0,0)

It is important to scale the data to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges.

6.2 Complexity vs Error

TBD

Talk about Bias vs. Variance

A simple model has high bias but low variance. Conversely, a complex model has low bias but high variance. To determine the optimal model a test set is required.

high complexity results in over fitting.

notes:

chalk board quadratic points estimate with line then high degree polynomial talk about Figure 14

use cross validation to find best parameter C and γ

Use cross-validation to find the best parameter C and γ Use the best parameter C and γ to train on the whole training set.

If using a polynomial kernel, the complexity increases as the degree of the polynomial increases.

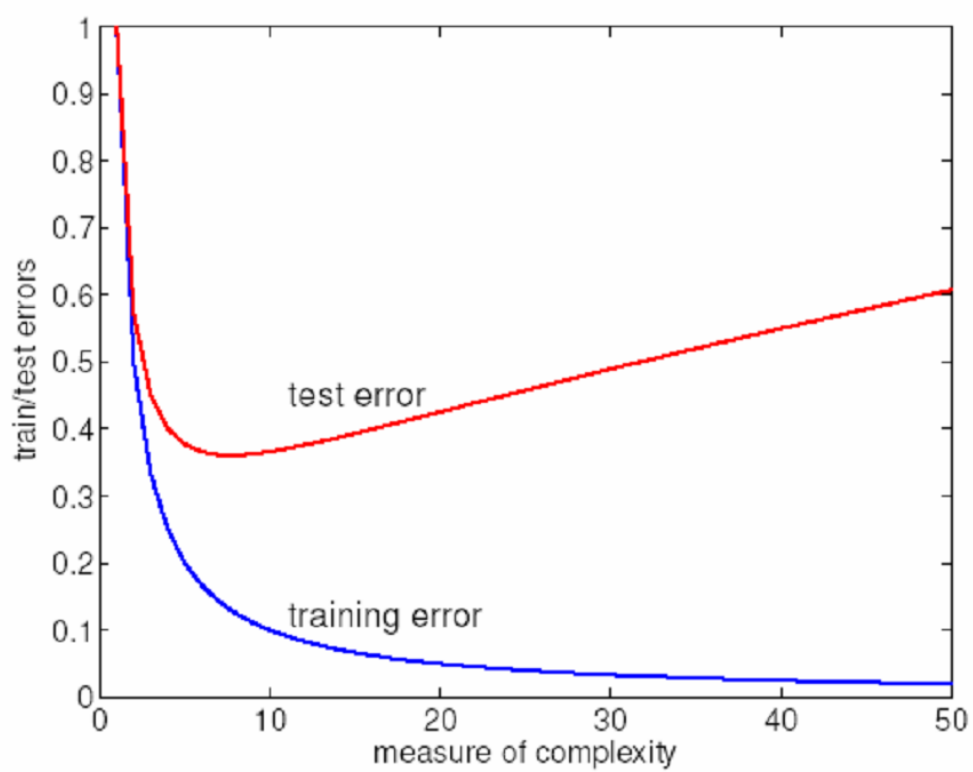


Figure 14: Complexity vs. Error

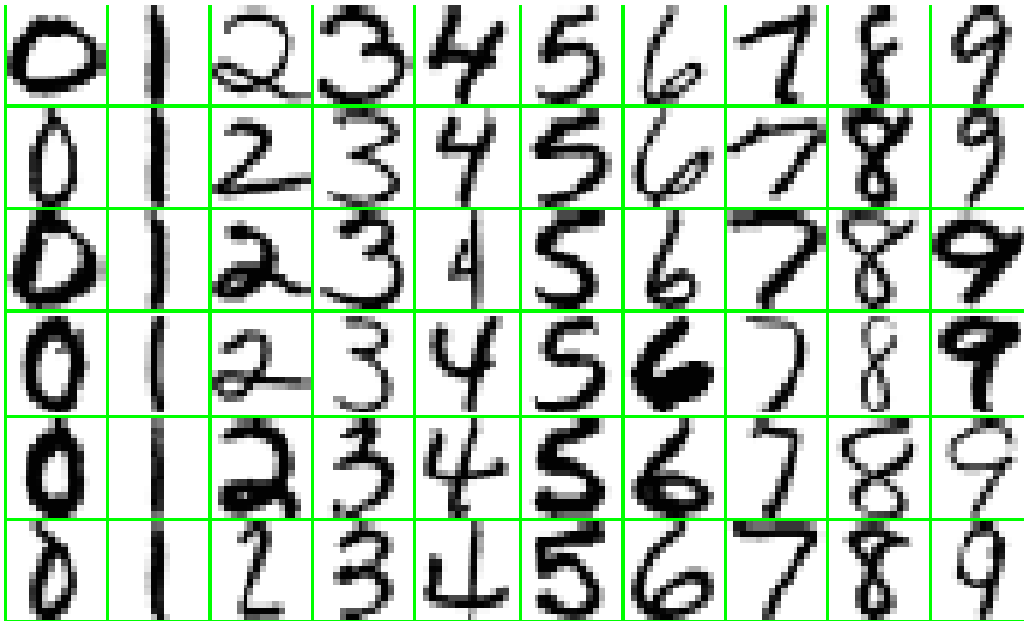


Figure 15: Example of Digits to Classify

7 Examples

7.1 Handwriting Example

The US postal mail system needed to recognize zip codes to automatically sort letters. Sixty handwritten digits are shown in Figure 15⁷. The images were deslanted and size normalized. The standard data set contained 60,000 examples. In the handwriting example each digit is encoded into a 16 x 16 pixel image. Hence, each $\mathbf{x}^{(i)}$ is simply a vector of length 256 which contains the intensity of the given pixel. The intensities varied from 0 to 255 in a grayscale. The geometry of the digit is not taken into account. It turns out that using an SVM with either a polynomial kernel or a Gaussian kernel produces great results. In fact the error rate using a 9th degree polynomial kernel is just 0.8%. [HTF09]

7.2 Protein Example TBD

Protein Example is TBD

⁷See [HTF09] <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>

A General Form of the Quadratic Optimization Problem

The General Form of a Quadratic Program (QP)[BV04]⁸ is

$$\begin{aligned} \min \{ & \frac{1}{2} \omega^T \mathbf{P} \omega + \mathbf{c}^T \omega + \mathbf{d} \} \\ \text{subject to } & \mathbf{G} \omega \preceq \mathbf{h} \\ & \mathbf{A} \omega = \mathbf{k} \end{aligned} \tag{43}$$

where $\omega \in \mathbb{R}^n$ is the optimization variable. The variables $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{d} \in \mathbb{R}$, $\mathbf{G} \in \mathbb{R}^{m \times n}$, $\mathbf{h} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{k} \in \mathbb{R}^p$, and $\mathbf{P} \in \mathbb{S}_+^n$, a symmetric positive semidefinite matrix, are defined by the problem. The symbol, \preceq , implies that every element of the vector, $\mathbf{G}\omega$, is less than the corresponding element of the vector \mathbf{h} .

Theorem 1. *The equation for the Optimal Margin Classifier, equation (3) is a QP.*

$$\begin{aligned} \min \{ & \frac{1}{2} \|\omega\|^2 \} \\ \text{subject to } & \mathbf{y}^{(i)} (\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq +1 \text{ for all } i \end{aligned} \tag{3}$$

Proof. Recall that there are m training samples and the number of features, n , is equal to the dimension of the vector $\mathbf{x}^{(i)}$.

First put the objective function into the correct form. Here $\mathbf{P} =$ the diagonal matrix with $1/2$ as entries. Hence, $\frac{1}{2} \|\omega\|^2 = \frac{1}{2} \omega^T \mathbf{P} \omega$. The rest of the terms to be minimized are all zero. That is $\mathbf{c} \equiv \mathbf{0}$ where $\mathbf{0} \in \mathbb{R}^n$ represents the 0 vector. Also, $\mathbf{d} = 0$. Similarly the equality constraint, $\mathbf{A} \omega = \mathbf{k}$ is also zeroed out.

Next work on the constraint. Define $\mathbf{G} = [-\text{diag}(y)\mathbf{X}]$ where

$$\mathbf{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix} \quad \text{and} \quad \mathbf{X} = \begin{bmatrix} x^{1T} \\ x^{2T} \\ \vdots \\ x^{mT} \end{bmatrix}$$

Note here that $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the matrix formed with the i th row equal to the transpose of the column vector x^i .

⁸Found online at <http://www.stanford.edu/~boyd/cvxbook/>

The constraint in (3) can be written as

$$\mathbf{y}^{(i)}(\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \geq +1 \Leftrightarrow -\mathbf{y}^{(i)}(\omega^T \mathbf{x}^{(i)} + \mathbf{b}) \leq -1 \Leftrightarrow -\mathbf{y}^{(i)}(\omega^T \mathbf{x}^{(i)}) \leq -1 + \mathbf{y}^{(i)} \mathbf{b} \quad (44)$$

Hence, $\mathbf{h} = \mathbf{y} \mathbf{b} - \mathbf{1}$ where $\mathbf{1} \in \mathbb{R}^m$ represents the vector with each entry equal to one.

Finally notice that

$$\mathbf{G} \omega = \begin{bmatrix} -y^1 & & & \\ & -y^2 & & \\ & & \ddots & \\ & & & y^m \end{bmatrix} \begin{bmatrix} x^{1T} \dots \\ x^{2T} \dots \\ \vdots \\ x^{mT} \dots \end{bmatrix} \begin{bmatrix} \omega \\ \vdots \end{bmatrix} = \begin{bmatrix} -y^1 x^{1T} \\ -y^2 x^{2T} \\ \vdots \\ -y^m x^{mT} \end{bmatrix} \begin{bmatrix} \omega \\ \vdots \end{bmatrix}$$

and the constraint in (3) can be written as

$$\mathbf{G} \omega \preceq \mathbf{h}$$

□

References

- [Bis06] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge UP, 2004.
- [GB08] M. Grant and S. Boyd. Cvx: Matlab software for disciplined convex programming. (*webpage and software*), 18:2564–2580, 2008.
- [HCL10] Chih-Wei Hsu, Chih-Chung Chang, and Chin-Jen Lin. A practical guide to support vector classification. *Tech Paper*, 18:1–16, 2010.
- [HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. Springer, 2009.
- [Pla07] John C. Platt. Fast training of support vector machines using sequential minimal optimization. (*MIT Press*), 12:41–65, 2007.