A survey of some Machine Learning models

Chris Cornwell

April 1, 2025

Outline

Support Vector Machines, continued

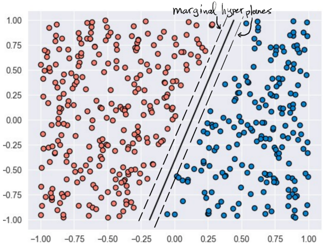
Kernels

Review - Goal of Maximum margin

The goal with a support vector machine, given sample data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^d$, is to find parameters $\omega = (\mathbf{w}, b)$, where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, so that $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ is satisfied for all i, and the norm of $\mathbf{w}$ is minimized (conventionally, you use half of the norm squared as a function to minimize).

Review - Goal of Maximum margin

The goal with a support vector machine, given sample data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^d$, is to find parameters $\omega = (\mathbf{w}, b)$, where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, so that $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ is satisfied for all i, and the norm of $\mathbf{w}$ is minimized (conventionally, you use half of the norm squared as a function to minimize). Minimizing the norm of $\mathbf{w}$ makes the *marginal hyperplanes*, where $\mathbf{w} \cdot \mathbf{x} + b = \pm 1$, be as far as possible from the hyperplane $\{\mathbf{x} \mid \mathbf{w} \cdot \mathbf{x} + b = 0\}$ as possible.



Using the method of Lagrange multipliers

Recall, can understand minimizing $\frac{1}{2}|\mathbf{w}|^2$ subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ through Lagrange multipliers.

Using the method of Lagrange multipliers

Recall, can understand minimizing $\frac{1}{2}|\mathbf{w}|^2$ subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ through Lagrange multipliers.
For $\underline{\alpha} = (\alpha_1, \ldots, \alpha_n)$, with $\alpha_i \in \mathbb{R}$, Lagrangian is

$$L(\mathbf{w}, b, \underline{\alpha}) = \frac{1}{2}|\mathbf{w}|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right).$$

It is minimized when

$$\begin{aligned}
\nabla_{\mathbf{w}} L = 0 \quad &\Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i; \\
\nabla_{b} L = 0 \quad &\Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0; \\
\alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right) = 0 \quad &\Rightarrow \quad \alpha_i = 0 \quad \text{OR} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1.
\end{aligned}$$

Using the method of Lagrange multipliers

Recall, can understand minimizing $\frac{1}{2}|\mathbf{w}|^2$ subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ through Lagrange multipliers.
For $\underline{\alpha} = (\alpha_1, \ldots, \alpha_n)$, with $\alpha_i \in \mathbb{R}$, Lagrangian is

$$L(\mathbf{w}, b, \underline{\alpha}) = \frac{1}{2}|\mathbf{w}|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right).$$

It is minimized when

$$\begin{aligned} \nabla_{\mathbf{w}} L = 0 \quad &\Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i; \\ \nabla_b L = 0 \quad &\Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0; \\ \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right) = 0 \quad &\Rightarrow \quad \alpha_i = 0 \quad \text{OR} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1. \end{aligned}$$

Support vectors are those $\mathbf{x}_i$ for which $\alpha_i \neq 0$, and so $\mathbf{w} \cdot \mathbf{x}_i + b = \pm 1$.

Slack Variables

The previous constrained minimization problem only has a solution if the ± 1-labeled data is linearly separable.

Slack Variables

The previous constrained minimization problem only has a solution if the ± 1-labeled data is linearly separable.
To accommodate for data that is not linearly separable, so-called **slack variables** ξ_i, with $1 \leq i \leq n$, are introduced in the constrained as follows.

Minimize: $\lambda |\mathbf{w}|^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i$

subject to: $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

Slack Variables

The previous constrained minimization problem only has a solution if the ± 1-labeled data is linearly separable.
To accommodate for data that is not linearly separable, so-called **slack variables** ξ_i, with $1 \leq i \leq n$, are introduced in the constrained as follows.

Minimize: $\lambda |\mathbf{w}|^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i$

subject to: $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

This minimization problem can likewise be approached through Lagrange multipliers.

Slack Variables

The previous constrained minimization problem only has a solution if the ± 1-labeled data is linearly separable.
To accommodate for data that is not linearly separable, so-called **slack variables** ξ_i, with $1 \leq i \leq n$, are introduced in the constrained as follows.

Minimize: $\lambda|\mathbf{w}|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i$

subject to: $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

This minimization problem can likewise be approached through Lagrange multipliers.
However, given $\mathbf{x}_k$ in the data, the corresponding ξ_k ought to be zero precisely if $\mathbf{x}_k$ is on the side of the hyperplane corresponding to its label y_k (and past the marginal hyperplane); that is, when $y_k(\mathbf{w} \cdot \mathbf{x}_k + b) \geq 1$. We'll use this observation to convert to the problem of minimizing a loss function.

SVMs, Minimizing a Regularized Loss Function

Fix some point $(\mathbf{x}, y) \in \mathbb{R}^d \times \{1, -1\}$. Given parameters $\mathbf{w}, b$, we define a function

$$\ell((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\mathbf{w} \cdot \mathbf{x} + b)\}.$$

SVMs, Minimizing a Regularized Loss Function

Fix some point $(\mathbf{x}, y) \in \mathbb{R}^d \times \{1, -1\}$. Given parameters $\mathbf{w}, b$, we define a function

$$\ell((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\mathbf{w} \cdot \mathbf{x} + b)\}.$$

Note that $\ell((\mathbf{w}, b), (\mathbf{x}, y)) \neq 0$ if and only if $y(\mathbf{w} \cdot \mathbf{x} + b) < 1$. We also noticed that, for some $(\mathbf{x}_k, y_k) \in \mathcal{S}$, we should only have $\xi_k \neq 0$ when $y_k(\mathbf{w} \cdot \mathbf{x}_k + b) < 1$.

SVMs, Minimizing a Regularized Loss Function

Fix some point $(\mathbf{x}, y) \in \mathbb{R}^d \times \{1, -1\}$. Given parameters $\mathbf{w}, b$, we define a function

$$\ell((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\mathbf{w} \cdot \mathbf{x} + b)\}.$$

Note that $\ell((\mathbf{w}, b), (\mathbf{x}, y)) \neq 0$ if and only if $y(\mathbf{w} \cdot \mathbf{x} + b) < 1$. We also noticed that, for some $(\mathbf{x}_k, y_k) \in \mathcal{S}$, we should only have $\xi_k \neq 0$ when $y_k(\mathbf{w} \cdot \mathbf{x}_k + b) < 1$.
And so, define $\mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b))$ to be the average

$$\mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b)) = \frac{1}{n} \sum_{i=1}^{n} \ell((\mathbf{w}, b), (\mathbf{x}_i, y_i)).$$

SVMs, Minimizing a Regularized Loss Function

Fix some point $(\mathbf{x}, y) \in \mathbb{R}^d \times \{1, -1\}$. Given parameters $\mathbf{w}, b$, we define a function

$$\ell((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\mathbf{w} \cdot \mathbf{x} + b)\}.$$

Note that $\ell((\mathbf{w}, b), (\mathbf{x}, y)) \neq 0$ if and only if $y(\mathbf{w} \cdot \mathbf{x} + b) < 1$. We also noticed that, for some $(\mathbf{x}_k, y_k) \in \mathcal{S}$, we should only have $\xi_k \neq 0$ when $y_k(\mathbf{w} \cdot \mathbf{x}_k + b) < 1$.
And so, define $\mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b))$ to be the average

$$\mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b)) = \frac{1}{n} \sum_{i=1}^{n} \ell((\mathbf{w}, b), (\mathbf{x}_i, y_i)).$$

Claim: The constrained minimization problem of the previous slide is equivalent to minimizing the function $\lambda|\mathbf{w}|^2 + \mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b))$.

SVMs, Minimizing a Regularized Loss Function

Fix some point $(\mathbf{x}, y) \in \mathbb{R}^d \times \{1, -1\}$. Given parameters $\mathbf{w}, b$, we define a function

$$\ell((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\mathbf{w} \cdot \mathbf{x} + b)\}.$$

Note that $\ell((\mathbf{w}, b), (\mathbf{x}, y)) \neq 0$ if and only if $y(\mathbf{w} \cdot \mathbf{x} + b) < 1$. We also noticed that, for some $(\mathbf{x}_k, y_k) \in \mathcal{S}$, we should only have $\xi_k \neq 0$ when $y_k(\mathbf{w} \cdot \mathbf{x}_k + b) < 1$.
And so, define $\mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b))$ to be the average

$$\mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b)) = \frac{1}{n} \sum_{i=1}^{n} \ell((\mathbf{w}, b), (\mathbf{x}_i, y_i)).$$

Claim: The constrained minimization problem of the previous slide is equivalent to minimizing the function $\lambda|\mathbf{w}|^2 + \mathcal{L}_{\mathcal{S}}^{hinge}((\mathbf{w}, b))$.

Proof.

Given $1 \leq k \leq n$, the best choice for ξ_k is 0 if $\ell((\mathbf{w}, b), (\mathbf{x}_k, y_k)) = 0$. Otherwise, by the constraint, we have $\xi_k \geq 1 - y_k(\mathbf{w} \cdot \mathbf{x}_k + b)$ and so the best choice is $\xi_k = \ell((\mathbf{w}, b), (\mathbf{x}_k, y_k))$. □

Gradient Descent for SVMs with Slack Variables

Say that $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, and let $x_{d+1} = 1$. The function $\ell((\mathbf{w}, b), (\mathbf{x}, y))$ has the following partial derivatives:

$$\frac{\partial \ell}{\partial w_j} = \begin{cases} 0, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) > 1 \\ -yx_j, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) < 1, \end{cases}$$

for all $1 \leq j \leq d + 1$.

Gradient Descent for SVMs with Slack Variables

Say that $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, and let $x_{d+1} = 1$. The function $\ell((\mathbf{w}, b), (\mathbf{x}, y))$ has the following partial derivatives:

$$\frac{\partial \ell}{\partial w_j} = \begin{cases} 0, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) > 1 \\ -yx_j, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) < 1, \end{cases}$$

for all $1 \leq j \leq d + 1$.

Note that the derivative is not defined if $y(\mathbf{w} \cdot \mathbf{x} + b) = 1$. However, this constitutes a set in $\mathbb{R}^d$ of volume zero – it will be encountered with probability zero.

Gradient Descent for SVMs with Slack Variables

Say that $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, and let $x_{d+1} = 1$. The function $\ell((\mathbf{w}, b), (\mathbf{x}, y))$ has the following partial derivatives:

$$\frac{\partial \ell}{\partial w_j} = \begin{cases} 0, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) > 1 \\ -yx_j, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) < 1, \end{cases}$$

for all $1 \leq j \leq d + 1$.

Note that the derivative is not defined if $y(\mathbf{w} \cdot \mathbf{x} + b) = 1$. However, this constitutes a set in $\mathbb{R}^d$ of volume zero – it will be encountered with probability zero.

If doing batch gradient descent, the above partial derivatives allow us to compute the gradient of the SVM regularized loss function, namely

$$2\lambda\mathbf{w} + \frac{1}{n}\sum_{i=1}^{n} \nabla\ell((\mathbf{w}, b), (\mathbf{x}_i, y_i)).$$

Gradient Descent for SVMs with Slack Variables

Say that $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, and let $x_{d+1} = 1$. The function $\ell((\mathbf{w}, b), (\mathbf{x}, y))$ has the following partial derivatives:

$$\frac{\partial \ell}{\partial w_j} = \begin{cases} 0, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) > 1 \\ -yx_j, \text{ if } y(\mathbf{w} \cdot \mathbf{x} + b) < 1, \end{cases}$$

for all $1 \leq j \leq d + 1$.

Note that the derivative is not defined if $y(\mathbf{w} \cdot \mathbf{x} + b) = 1$. However, this constitutes a set in $\mathbb{R}^d$ of volume zero – it will be encountered with probability zero.

If doing batch gradient descent, the above partial derivatives allow us to compute the gradient of the SVM regularized loss function, namely

$$2\lambda\mathbf{w} + \frac{1}{n}\sum_{i=1}^{n} \nabla\ell((\mathbf{w}, b), (\mathbf{x}_i, y_i)).$$

However, if doing stochastic gradient descent (SGD, which only the loss on a single point from $\mathcal{S}$), then for that selected point $(\mathbf{x}, y)$, we get

$$2\lambda\mathbf{w} + \nabla\ell((\mathbf{w}, b), (\mathbf{x}, y)).$$

A Procedure for SGD on SVM

The following is a procedure that will carry out Stochastic Gradient Descent for an SVM (with slack variables).

```
## lambda: the coeff of regularization; T: the number of iterations
input: x, y, lambda, T
theta[1] ← initial array of d+1 zeros
X ← (x,1) # 1's in last column
for (t = 1,...,T){
    W[t] ← theta[t]/(2*lambda*t)
    Choose i uniformly at random from 1,...,n
    if (y[i]*dot(W[t], X[i]) < 1)
        theta[t+1] ← theta[t] + y[i]*X[i]
    else
        theta[t+1] ← theta[t]
}
return average of W[1], ..., W[T]
```

Outline

Support Vector Machines, continued

Kernels

SVMs with Non-linear Decision Boundaries

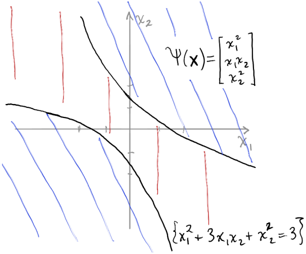
To use a hyperplane with normal vector $\mathbf{w}$, and shift b, but to get a predictive model that has non-linear decision boundary: first send the data through a map $\psi : \mathbb{R}^d \longrightarrow \mathbb{R}^D$, with $D > d$ (usually); then, use a hyperplane in $\mathbb{R}^D$.

SVMs with Non-linear Decision Boundaries

To use a hyperplane with normal vector $\mathbf{w}$, and shift b, but to get a predictive model that has non-linear decision boundary: first send the data through a map $\psi : \mathbb{R}^d \longrightarrow \mathbb{R}^D$, with $D > d$ (usually); then, use a hyperplane in $\mathbb{R}^D$.

Example. Define $\psi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ so that, for $\mathbf{x} = (x_1, x_2)$ we have

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}.$$



x_2
x_1
$\Psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$
$\{x_1^2 + 3x_1x_2 + x_2^2 = 3\}$

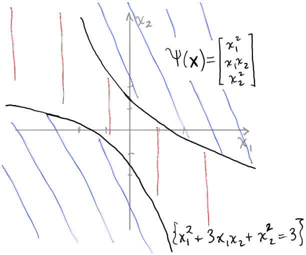
SVMs with Non-linear Decision Boundaries

To use a hyperplane with normal vector **w**, and shift b, but to get a predictive model that has non-linear decision boundary: first send the data through a map $\psi : \mathbb{R}^d \longrightarrow \mathbb{R}^D$, with $D > d$ (usually); then, use a hyperplane in $\mathbb{R}^D$.

Example. Define $\psi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ so that, for $\mathbf{x} = (x_1, x_2)$ we have

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}.$$

Letting $\mathbf{w} = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$ and $b = -3$, the set of points $\mathbf{x} \in \mathbb{R}^2$ such that $\mathbf{w} \cdot \psi(\mathbf{x}) + b = 0$ is union of two curves depicted to the right.


x_2
x_1
$\Psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}$
$\{x_1^2 + 3x_1x_2 + x_2^2 = 3\}$

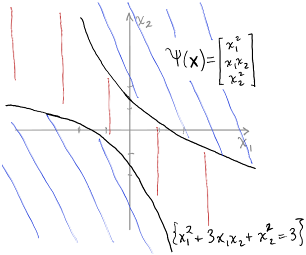
SVMs with Non-linear Decision Boundaries

To use a hyperplane with normal vector $\mathbf{w}$, and shift b, but to get a predictive model that has non-linear decision boundary: first send the data through a map $\psi : \mathbb{R}^d \to \mathbb{R}^D$, with $D > d$ (usually); then, use a hyperplane in $\mathbb{R}^D$.

Example. Define $\psi : \mathbb{R}^2 \to \mathbb{R}^3$ so that, for $\mathbf{x} = (x_1, x_2)$ we have

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}.$$

Letting $\mathbf{w} = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$ and $b = -3$, the set of points $\mathbf{x} \in \mathbb{R}^2$ such that $\mathbf{w} \cdot \psi(\mathbf{x}) + b = 0$ is union of two curves depicted to the right.


x_2
x_1
$\Psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$
$\{x_1^2 + 3x_1x_2 + x_2^2 = 3\}$

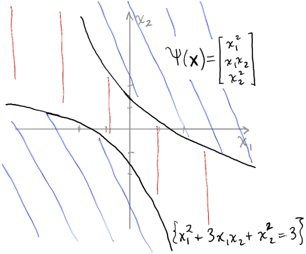
The set of $\mathbf{x} \in \mathbb{R}^2$ that this model would label positively are those such that $\mathbf{w} \cdot \psi(\mathbf{x}) + b > 0$, shaded in blue. (A hyperplane in $\mathbb{R}^3$ separates images, under ψ, of positively and negatively labeled points.)

SVMs with Non-linear Decision Boundaries

Example (cont'd). We have

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$$

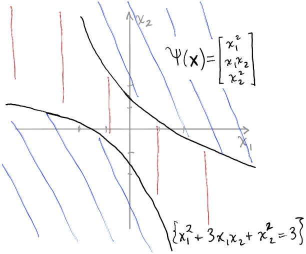
and $\mathbf{w} = (w_1, w_2, w_3)$. Say that the data is modeled well by this *type* of decision boundary (maybe not perfectly separated though).


x_2
$\Psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}$
x_1
$\{x_1^2 + 3x_1x_2 + x_2^2 = 3\}$

SVMs with Non-linear Decision Boundaries

Example (cont'd). We have

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$$


x_2
x_1
$\Psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}$
$\{x_1^2 + 3x_1x_2 + x_2^2 = 3\}$

and $\mathbf{w} = (w_1, w_2, w_3)$. Say that the data is modeled well by this *type* of decision boundary (maybe not perfectly separated though).

Then solving the minimization problem, over $\mathbf{w} \in \mathbb{R}^3$, $b \in \mathbb{R}$,

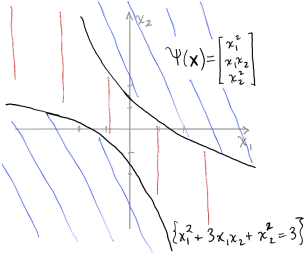
$$\min_{\mathbf{w},b} \qquad \lambda|\mathbf{w}|^2 + \frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i(\mathbf{w} \cdot \psi(\mathbf{x}_i) + b)\}$$

is a non-linear SVM (potentially with some points having a non-zero slack variable).

SVMs with Non-linear Decision Boundaries

Example (cont'd). We have

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$$


x_2
x_1
$\Psi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$
$\{x_1^2 + 3x_1x_2 + x_2^2 = 3\}$

and $\mathbf{w} = (w_1, w_2, w_3)$. Say that the data is modeled well by this *type* of decision boundary (maybe not perfectly separated though).

Then solving the minimization problem, over $\mathbf{w} \in \mathbb{R}^3$, $b \in \mathbb{R}$,

$$\min_{\mathbf{w},b} \qquad \lambda|\mathbf{w}|^2 + \frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i(\mathbf{w} \cdot \psi(\mathbf{x}_i) + b)\}$$

is a non-linear SVM (potentially with some points having a non-zero slack variable).

But, how can we do this? Especially if the map ψ is not known beforehand?

Lagrangian Dual Problem

Recall (from earlier SVM lecture), the Lagrange multiplier method leads to a "dual" maximization problem that is an equivalent one:[1]

$$\max_{\underline{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j).$$

[1] This is the version without slack variables. There is one with slack variables.

[2] The Lagrange multipliers then, in turn, determine both **w** and *b*.

Lagrangian Dual Problem

Recall (from earlier SVM lecture), the Lagrange multiplier method leads to a “dual” maximization problem that is an equivalent one:[1]

$$\max_{\underline{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j).$$

Note that this does not require knowing the actual points $\mathbf{x}_i \in \mathbb{R}^d$, only the dot products between pairs $\mathbf{x}_i$ and $\mathbf{x}_j$. It also does not require that we determine $(\mathbf{w}, b)$, but the multipliers $\alpha_1, \ldots, \alpha_n$ instead. [2]

[1] This is the version without slack variables. There is one with slack variables.

[2] The Lagrange multipliers then, in turn, determine both **w** and *b*.

Lagrangian Dual Problem

Recall (from earlier SVM lecture), the Lagrange multiplier method leads to a "dual" maximization problem that is an equivalent one:[1]

$$\max_{\underline{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j).$$

Note that this does not require knowing the actual points $\mathbf{x}_i \in \mathbb{R}^d$, only the dot products between pairs $\mathbf{x}_i$ and $\mathbf{x}_j$. It also does not require that we determine $(\mathbf{w}, b)$, but the multipliers $\alpha_1, \ldots, \alpha_n$ instead. [2]

These observations are part of a more general phenomenon.

[1] This is the version without slack variables. There is one with slack variables.

[2] The Lagrange multipliers then, in turn, determine both **w** and *b*.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$.

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$. We consider the minimization problem

$$\min_{\omega} \quad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$. We consider the minimization problem

$$\min_{\omega} \qquad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

- Our SVM optimization is an instance of this with[4] $f(a_1, \ldots, a_n) = \frac{1}{n} \sum \max\{0, 1 - y_i a_i\}$ and $R(a) = \lambda a^2$.

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$. We consider the minimization problem

$$\min_{\omega} \qquad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

- Our SVM optimization is an instance of this with[4] $f(a_1, \ldots, a_n) = \frac{1}{n}\sum \max\{0, 1 - y_i a_i\}$ and $R(a) = \lambda a^2$.

Often, $H = \mathbb{R}^D$ for some integer $D > 0$. However, this theorem works more generally, H being something called a *Hilbert space*.

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$.
We consider the minimization problem

$$\min_{\omega} \quad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

- Our SVM optimization is an instance of this with[4] $f(a_1, \ldots, a_n) = \frac{1}{n}\sum \max\{0, 1 - y_i a_i\}$ and $R(a) = \lambda a^2$.

Often, $H = \mathbb{R}^D$ for some integer $D > 0$. However, this theorem works more generally, H being something called a *Hilbert space*.

Theorem (Representer Theorem)

There exists vector $\underline{\alpha} \in \mathbb{R}^n$ *such that* $\omega = \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i)$ *is a minimizer of* $(\dagger)$*.*

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$.
We consider the minimization problem

$$\min_{\omega} \qquad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

- Our SVM optimization is an instance of this with[4] $f(a_1, \ldots, a_n) = \frac{1}{n} \sum \max\{0, 1 - y_i a_i\}$ and $R(a) = \lambda a^2$.

Often, $H = \mathbb{R}^D$ for some integer $D > 0$. However, this theorem works more generally, H being something called a *Hilbert space*.

Theorem (Representer Theorem)

There exists vector $\underline{\alpha} \in \mathbb{R}^n$ *such that* $\omega = \sum_{i=1}^n \alpha_i \psi(\mathbf{x}_i)$ *is a minimizer of* ($\dagger$).

Proof sketch.

Given ω from theorem statement, if ω^* is minimizer, can write $\omega^* = \omega + \upsilon$ with υ orthogonal to $\psi(\mathbf{x}_i)$ for all i, which implies $|\omega|^2 \leq |\omega^*|^2$.

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$. We consider the minimization problem

$$\min_{\omega} \quad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \dots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

- Our SVM optimization is an instance of this with[4] $f(a_1, \dots, a_n) = \frac{1}{n}\sum \max\{0, 1 - y_i a_i\}$ and $R(a) = \lambda a^2$.

Often, $H = \mathbb{R}^D$ for some integer $D > 0$. However, this theorem works more generally, H being something called a *Hilbert space*.

Theorem (Representer Theorem)

There exists vector $\underline{\alpha} \in \mathbb{R}^n$ *such that* $\omega = \sum_{i=1}^n \alpha_i \psi(\mathbf{x}_i)$ *is a minimizer of* ($\dagger$).

Proof sketch.

Given ω from theorem statement, if ω^* is minimizer, can write $\omega^* = \omega + \upsilon$ with υ orthogonal to $\psi(\mathbf{x}_i)$ for all i, which implies $|\omega|^2 \leq |\omega^*|^2$. This means $R(|\omega|) \leq R(|\omega^*|)$; can check that $y_i\langle \omega, \psi(\mathbf{x}_i)\rangle = y_i\langle \omega^*, \psi(\mathbf{x}_i)\rangle$ for all i.

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernels - The Representer Theorem

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an arbitrary function and let $R : \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ be an increasing[3] function. Further, say that we have a map $\psi : \mathbb{R}^d \longrightarrow H$.
We consider the minimization problem

$$\min_{\omega} \qquad f\left(\langle \omega, \psi(\mathbf{x}_1)\rangle, \ldots, \langle \omega, \psi(\mathbf{x}_n)\rangle\right) + R(|\omega|). \qquad (\dagger)$$

- Our SVM optimization is an instance of this with[4] $f(a_1, \ldots, a_n) = \frac{1}{n}\sum \max\{0, 1 - y_i a_i\}$ and $R(a) = \lambda a^2$.

Often, $H = \mathbb{R}^D$ for some integer $D > 0$. However, this theorem works more generally, H being something called a *Hilbert space*.

Theorem (Representer Theorem)

There exists vector $\underline{\alpha} \in \mathbb{R}^n$ *such that* $\omega = \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i)$ *is a minimizer of* $(\dagger)$*.*

Proof sketch.

Given ω from theorem statement, if ω^* is minimizer, can write $\omega^* = \omega + \upsilon$ with υ orthogonal to $\psi(\mathbf{x}_i)$ for all i, which implies $|\omega|^2 \leq |\omega^*|^2$. This means $R(|\omega|) \leq R(|\omega^*|)$; can check that $y_i\langle \omega, \psi(\mathbf{x}_i)\rangle = y_i\langle \omega^*, \psi(\mathbf{x}_i)\rangle$ for all i. Put together $\implies$ value of $(\dagger)$ on ω is not more than its value on ω^*. □

[3] Really only need *non-decreasing*: if $a_1 < a_2$ then $R(a_1) \leq R(a_2)$.

[4] Suppose last coordinate of $\psi(\mathbf{x})$ to be 1 and $\omega = (\mathbf{w}, b)$.

Kernel Trick

By the Representer Theorem, we can consider the optimal $\omega = \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i)$. Then we are able to rewrite the optimization problem (†).

$$\langle \omega, \psi(\mathbf{x}_j) \rangle = \sum_{i=1}^{n} \alpha_i \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$$

$$|\omega|^2 = \langle \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i), \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i) \rangle = \sum_{i,j=1}^{n} \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle.$$

We put these expressions into f and R in (†); they only depend on $\alpha_1, \ldots, \alpha_n$ and the dot products[5] between $\psi(\mathbf{x}_i)$ and $\psi(\mathbf{x}_j)$.

[5] *Inner products*, more generally.

Kernel Trick

By the Representer Theorem, we can consider the optimal $\omega = \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i)$. Then we are able to rewrite the optimization problem (†).

$$\langle \omega, \psi(\mathbf{x}_j) \rangle = \sum_{i=1}^{n} \alpha_i \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$$

$$|\omega|^2 = \langle \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i), \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i) \rangle = \sum_{i,j=1}^{n} \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle.$$

We put these expressions into f and R in (†); they only depend on $\alpha_1, \ldots, \alpha_n$ and the dot products[5] between $\psi(\mathbf{x}_i)$ and $\psi(\mathbf{x}_j)$.
A “trick”: don’t figure out ψ, but decide on a function $K(\mathbf{x}, \mathbf{x}')$ that will determine the dot products $\langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$.

[5] *Inner products*, more generally.

Kernel Trick

By the Representer Theorem, we can consider the optimal $\omega = \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i)$. Then we are able to rewrite the optimization problem (†).

$$\langle \omega, \psi(\mathbf{x}_j) \rangle = \sum_{i=1}^{n} \alpha_i \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$$

$$|\omega|^2 = \langle \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i), \sum_{i=1}^{n} \alpha_i \psi(\mathbf{x}_i) \rangle = \sum_{i,j=1}^{n} \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle.$$

We put these expressions into f and R in (†); they only depend on $\alpha_1, \ldots, \alpha_n$ and the dot products[5] between $\psi(\mathbf{x}_i)$ and $\psi(\mathbf{x}_j)$.
A "trick": don't figure out ψ, but decide on a function $K(\mathbf{x}, \mathbf{x}')$ that will determine the dot products $\langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$. Often the dimension D used by ψ needs to be fairly large and dot products in high dimension can be computationally expensive. However, if we choose K and get the **Gram matrix**, with (i, j) entry $K(\mathbf{x}_i, \mathbf{x}_j)$, such dot products not needed; optimize the parameters α_i only.

[5] *Inner products*, more generally.

Popular Kernel Functions

Two common choices for kernel function K are listed below. In general, the corresponding Gram matrix should be symmetric and *positive semi-definite*.

Popular Kernel Functions

Two common choices for kernel function K are listed below. In general, the corresponding Gram matrix should be symmetric and *positive semi-definite*.

1. **Polynomial kernels.** For constants $r \geq 0, \gamma > 0$, and positive integer d, set

$$K(\mathbf{x}, \mathbf{x}') = (r + \gamma(\mathbf{x} \cdot \mathbf{x}'))^d.$$

   In the definition, $\mathbf{x} \cdot \mathbf{x}'$ is usually the standard dot product, but could be another inner product.

Popular Kernel Functions

Two common choices for kernel function K are listed below. In general, the corresponding Gram matrix should be symmetric and *positive semi-definite*.

1. **Polynomial kernels.** For constants $r \geq 0, \gamma > 0$, and positive integer d, set

   $$K(\mathbf{x}, \mathbf{x}') = (r + \gamma(\mathbf{x} \cdot \mathbf{x}'))^d.$$

   In the definition, $\mathbf{x} \cdot \mathbf{x}'$ is usually the standard dot product, but could be another inner product.

2. **Gaussian kernels.** For constant $\gamma > 0$, set

   $$K(\mathbf{x}, \mathbf{x}') = e^{-\gamma|\mathbf{x}-\mathbf{x}'|^2}.$$

   These kernels are also called “Radial Basis Function” (RBF) kernels.

Popular Kernel Functions

Two common choices for kernel function K are listed below. In general, the corresponding Gram matrix should be symmetric and *positive semi-definite*.

1. **Polynomial kernels.** For constants $r \geq 0, \gamma > 0$, and positive integer d, set

$$K(\mathbf{x}, \mathbf{x}') = (r + \gamma(\mathbf{x} \cdot \mathbf{x}'))^d.$$

   In the definition, $\mathbf{x} \cdot \mathbf{x}'$ is usually the standard dot product, but could be another inner product.

2. **Gaussian kernels.** For constant $\gamma > 0$, set

$$K(\mathbf{x}, \mathbf{x}') = e^{-\gamma|\mathbf{x}-\mathbf{x}'|^2}.$$

   These kernels are also called "Radial Basis Function" (RBF) kernels.

- Think about a polynomial kernel with $r = \gamma = 1$ and $d = 2$. Further, say that $\mathbf{x}$ and $\mathbf{x}'$ are in $\mathbb{R}^2$. Check that $\psi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1) \in \mathbb{R}^6$ will give the equation

$$K(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x}) \cdot \psi(\mathbf{x}').$$

SGD Procedure for SVM with Kernels (and slack variables)

Here is the SGD procedure with a kernel (using the Gram matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$).

SGD Procedure for SVM with Kernels (and slack variables)

Here is the SGD procedure with a kernel (using the Gram matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$).

The algorithm uses gradient descent to iteratively find $\alpha_1^{(t)}, \ldots, \alpha_n^{(t)}$ at step t.

During it, scalars $\beta_1^{(t)}, \ldots, \beta_n^{(t)}$ are used. Writing `beta[i][t]` for $\beta_i^{(t)}$ and `psi` for ψ, the relation to our previous SGD algorithm is

`theta[t] = sum( beta[i][t]*psi(X[i]), i=1,...,n )`.

SGD Procedure for SVM with Kernels (and slack variables)

Here is the SGD procedure with a kernel (using the Gram matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$).

The algorithm uses gradient descent to iteratively find $\alpha_1^{(t)}, \ldots, \alpha_n^{(t)}$ at step t.

During it, scalars $\beta_1^{(t)}, \ldots, \beta_n^{(t)}$ are used. Writing `beta[i][t]` for $\beta_i^{(t)}$ and `psi` for ψ, the relation to our previous SGD algorithm is

`theta[t] = sum( beta[i][t]*psi(X[i]), i=1,...,n ).`

```
## lambda: the coeff of regularization; T: the number of iterations
## K: n by n Gram matrix
input: K, y, lambda, T
beta[i][1] ← initial zero for i from 1,...,n
for (t = 1,...,T){
    alpha[i][t] ← beta[i][t]/(2*lambda*t)  # for i from i,...,n
    Choose j uniformly at random from 1,...,n
    beta[i][t+1] ← beta[i][t] # for i not equal j
    if (y[j]*sum( alpha[i][t]*K[i,j], i from 1,...,n ) < 1)
        beta[j][t+1] ← beta[j][t] + y[j]
    else
        beta[j][t+1] ← beta[j][t]
}
return average (alpha[i][1], ..., alpha[i][T]) # for i from 1,...,n
```

SGD Procedure for SVM with Kernels (and slack variables)

Here is the SGD procedure with a kernel (using the Gram matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$).

The algorithm uses gradient descent to iteratively find $\alpha_1^{(t)}, \ldots, \alpha_n^{(t)}$ at step t.

During it, scalars $\beta_1^{(t)}, \ldots, \beta_n^{(t)}$ are used. Writing `beta[i][t]` for $\beta_i^{(t)}$ and `psi` for ψ, the relation to our previous SGD algorithm is

```
theta[t] = sum( beta[i][t]*psi(X[i]), i=1,...,n ).
```

```
## lambda: the coeff of regularization; T: the number of iterations
## K: n by n Gram matrix
input: K, y, lambda, T
beta[i][1] ← initial zero for i from 1,...,n
for (t = 1,...,T){
    alpha[i][t] ← beta[i][t]/(2*lambda*t)  # for i from i,...,n
    Choose j uniformly at random from 1,...,n
    beta[i][t+1] ← beta[i][t] # for i not equal j
    if (y[j]*sum( alpha[i][t]*K[i,j], i from 1,...,n ) < 1)
        beta[j][t+1] ← beta[j][t] + y[j]
    else
        beta[j][t+1] ← beta[j][t]
}
return average (alpha[i][1], ..., alpha[i][T]) # for i from 1,...,n
```

Note: The vectors $W^{(t)}$ of previous algorithm are $\sum_{i=1}^{n} \alpha_i^{(t)} \psi(\mathbf{x}_i)$. But, writing $\overline{W}$ for average of the $W^{(t)}$, to get prediction on unseen $\mathbf{x} \in \mathbb{R}^d$ we just need

$$\langle \overline{W}, \psi(\mathbf{x}) \rangle = \sum_i \overline{\alpha}_i K(\mathbf{x}_i, \mathbf{x}).$$