Support Vector Machines

Greg Mori - CMPT 419/726

Bishop PRML Ch. 7

Outline

Maximum Margin Criterion

Math

Maximizing the Margin

Non-Separable Data

Outline

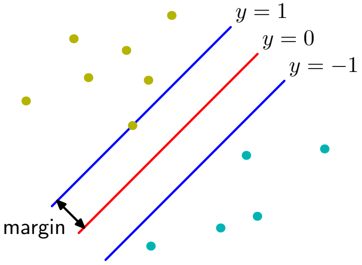
Linear Classification

- Consider a two class classification problem
- Use a linear model

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b$$

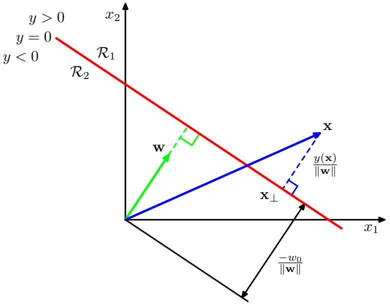
  followed by a threshold function
- For now, let's assume training data are linearly separable
  - Recall that the perceptron would converge to a perfect classifier for such data
  - But there are many such perfect classifiers

Max Margin



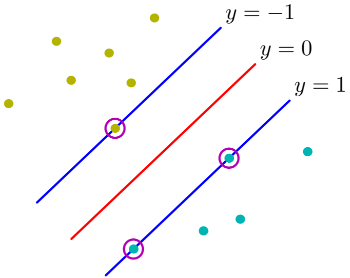
- We can define the margin of a classifier as the minimum distance to any example
- In support vector machines the decision boundary which maximizes the margin is chosen

Marginal Geometry



- Recall from Ch. 4
- Projection of $\boldsymbol{x}$ in $\boldsymbol{w}$ dir. is $\frac{\boldsymbol{w}^T\boldsymbol{x}}{||w||}$
- $y(\boldsymbol{x}) = 0$ when $\boldsymbol{w}^T\boldsymbol{x} = -b$, or $\frac{\boldsymbol{w}^T\boldsymbol{x}}{||\boldsymbol{w}||} = \frac{-b}{||\boldsymbol{w}||}$
- So $\frac{\boldsymbol{w}^T\boldsymbol{x}}{||\boldsymbol{w}||} - \frac{-b}{||\boldsymbol{w}||} = \frac{y(\boldsymbol{x})}{||\boldsymbol{w}||}$ is signed distance to decision boundary

Support Vectors



- Assuming data are separated by the hyperplane, distance to decision boundary is $\frac{t_n y(\boldsymbol{x}_n)}{||\boldsymbol{w}||}$
- The maximum margin criterion chooses $\boldsymbol{w}, b$ by:

$$\arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n [t_n (\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b)] \right\}$$

- Points with this min value are known as support vectors

Canonical Representation

- This optimization problem is complex:

$$\arg\max_{w,b}\left\{\frac{1}{||w||}\min_n[t_n(w^T\phi(x_n)+b)]\right\}$$

- Note that rescaling $w \to \kappa w$ and $b \to \kappa b$ does not change distance $\frac{t_n y(x_n)}{||w||}$ (many equiv. answers)
- So for x_* closest to surface, can set:

$$t_*(w^T\phi(x_*)+b)=1$$

- All other points are at least this far away:

$$\forall n\ ,\ t_n(w^T\phi(x_n)+b)\geq 1$$

- Under these constraints, the optimization becomes:

$$\arg\max_{w,b}\frac{1}{||w||}=\arg\min_{w,b}\frac{1}{2}||w||^2$$

Canonical Representation

- This optimization problem is complex:

$$\arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n [t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b)] \right\}$$

- Note that rescaling $\boldsymbol{w} \to \kappa \boldsymbol{w}$ and $b \to \kappa b$ does not change distance $\frac{t_n y(\boldsymbol{x}_n)}{||\boldsymbol{w}||}$ (many equiv. answers)
- So for x_* closest to surface, can set:

$$t_*(\boldsymbol{w}^T \phi(\boldsymbol{x}_*) + b) = 1$$

- All other points are at least this far away:

$$\forall n \ , \ t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1$$

- Under these constraints, the optimization becomes:

$$\arg\max_{\boldsymbol{w},b} \frac{1}{||\boldsymbol{w}||} = \arg\min_{\boldsymbol{w},b} \frac{1}{2} ||\boldsymbol{w}||^2$$

Canonical Representation

- This optimization problem is complex:

$$\arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n [t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b)] \right\}$$

- Note that rescaling $\boldsymbol{w} \rightarrow \kappa \boldsymbol{w}$ and $b \rightarrow \kappa b$ does not change distance $\frac{t_n y(\boldsymbol{x}_n)}{||\boldsymbol{w}||}$ (many equiv. answers)
- So for $\boldsymbol{x}_*$ closest to surface, can set:

$$t_*(\boldsymbol{w}^T \phi(\boldsymbol{x}_*) + b) = 1$$

- All other points are at least this far away:

$$\forall n \ , \ t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1$$

- Under these constraints, the optimization becomes:

$$\arg\max_{\boldsymbol{w},b} \frac{1}{||\boldsymbol{w}||} = \arg\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2$$

Canonical Representation

- This optimization problem is complex:

$$\arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n [t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b)] \right\}$$

- Note that rescaling $\boldsymbol{w} \rightarrow \kappa \boldsymbol{w}$ and $b \rightarrow \kappa b$ does not change distance $\frac{t_n y(\boldsymbol{x}_n)}{||\boldsymbol{w}||}$ (many equiv. answers)
- So for $\boldsymbol{x}_*$ closest to surface, can set:

$$t_*(\boldsymbol{w}^T \phi(\boldsymbol{x}_*) + b) = 1$$

- All other points are at least this far away:

$$\forall n \;,\; t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1$$

- Under these constraints, the optimization becomes:

$$\arg\max_{\boldsymbol{w},b} \frac{1}{||\boldsymbol{w}||} = \arg\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2$$

Canonical Representation

- So the optimization problem is now a constrained optimization problem:

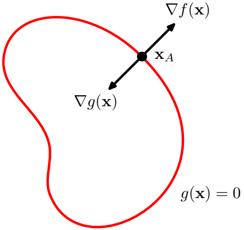
$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2$$

$$s.t. \qquad \forall n\ ,\ t_n(\boldsymbol{w}^T\phi(\boldsymbol{x}_n) + b) \geq 1$$

- To solve this, we need to take a detour into Lagrange multipliers

Outline

Lagrange Multipliers



Consider the problem:

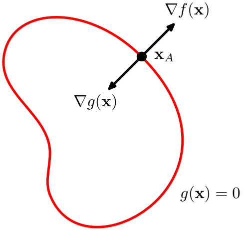
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) = 0$$

- Points on $g(\boldsymbol{x}) = 0$ must have $\nabla g(\boldsymbol{x})$ normal to surface
- A stationary point must have no change in f in the direction of the surface, so $\nabla f(\boldsymbol{x})$ must also be in this same direction
  - So there must be some λ such that $\nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$
- Define Lagrangian:

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

  - Stationary points of $L(\boldsymbol{x}, \lambda)$ have $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$ and $\nabla_{\lambda} L(\boldsymbol{x}, \lambda) = g(\boldsymbol{x}) = 0$
  - So are stationary points of constrained problem!

Lagrange Multipliers



Consider the problem:

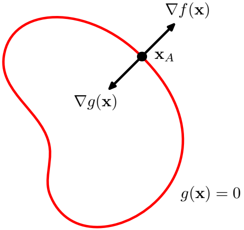
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) = 0$$

- Points on $g(\boldsymbol{x}) = 0$ must have $\nabla g(\boldsymbol{x})$ normal to surface
- A stationary point must have no change in f in the direction of the surface, so $\nabla f(\boldsymbol{x})$ must also be in this same direction
    - So there must be some λ such that $\nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$
- Define Lagrangian:

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

  - Stationary points of $L(\boldsymbol{x}, \lambda)$ have $\nabla_x L(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$ and $\nabla_\lambda L(\boldsymbol{x}, \lambda) = g(\boldsymbol{x}) = 0$
  - So are stationary points of constrained problem!

Lagrange Multipliers



Consider the problem:

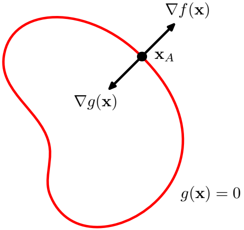
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) = 0$$

- Points on $g(\boldsymbol{x}) = 0$ must have $\nabla g(\boldsymbol{x})$ normal to surface
- A stationary point must have no change in f in the direction of the surface, so $\nabla f(\boldsymbol{x})$ must also be in this same direction
  - So there must be some λ such that $\nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$
- Define Lagrangian:

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

  - Stationary points of $L(\boldsymbol{x}, \lambda)$ have $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$ and $\nabla_{\lambda} L(\boldsymbol{x}, \lambda) = g(\boldsymbol{x}) = 0$
  - So are stationary points of constrained problem!

Lagrange Multipliers



Consider the problem:

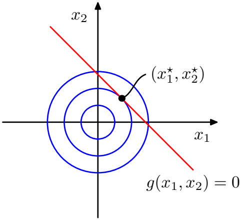
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) = 0$$

- Points on $g(\boldsymbol{x}) = 0$ must have $\nabla g(\boldsymbol{x})$ normal to surface
- A stationary point must have no change in f in the direction of the surface, so $\nabla f(\boldsymbol{x})$ must also be in this same direction
  - So there must be some λ such that $\nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$
- Define Lagrangian:

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

  - Stationary points of $L(\boldsymbol{x}, \lambda)$ have $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) + \lambda \nabla g(\boldsymbol{x}) = 0$ and $\nabla_{\lambda} L(\boldsymbol{x}, \lambda) = g(\boldsymbol{x}) = 0$
  - So are stationary points of constrained problem!

Lagrange Multipliers Example



- Consider the problem

$$\max_{\boldsymbol{x}} f(x_1, x_2) = 1 - x_1^2 - x_2^2$$
$$s.t. \qquad g(x_1, x_2) = x_1 + x_2 - 1 = 0$$

- Lagrangian:

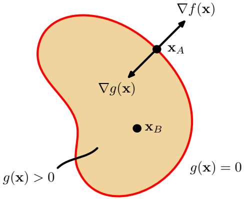
$$L(\boldsymbol{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$$

- Stationary points require:

$$\begin{aligned} \partial L/\partial x_1 &= -2x_1 + \lambda = 0 \\ \partial L/\partial x_2 &= -2x_2 + \lambda = 0 \\ \partial L/\partial \lambda &= x_1 + x_2 - 1 = 0 \end{aligned}$$

- So stationary point is $(x_1^*, x_2^*) = (\frac{1}{2}, \frac{1}{2})$, $\lambda = 1$

Lagrange Multipliers - Inequality Constraints



Consider the problem:

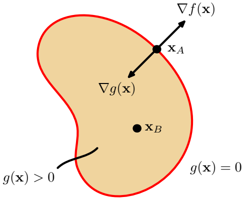
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) \geq 0$$

- Optimization over a region – solutions either at stationary points (gradients 0) in region or on boundary

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

- Solutions have either:
  - $\nabla f(\boldsymbol{x}) = 0$ and $\lambda = 0$ (in region), or
  - $\nabla f(\boldsymbol{x}) = -\lambda \nabla g(\boldsymbol{x})$ and $\lambda > 0$ (on boundary, $>$ for maximizing f).
  - For both, $\lambda g(\boldsymbol{x}) = 0$
- Solutions have $g(\boldsymbol{x}) \geq 0, \lambda \geq 0, \lambda g(\boldsymbol{x}) = 0$

Lagrange Multipliers - Inequality Constraints



Consider the problem:

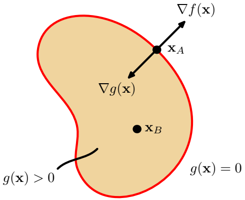
$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) \geq 0$$

- Optimization over a region – solutions either at stationary points (gradients 0) in region or on boundary

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

- Solutions have either:
  - $\nabla f(\boldsymbol{x}) = 0$ and $\lambda = 0$ (in region), or
  - $\nabla f(\boldsymbol{x}) = -\lambda \nabla g(\boldsymbol{x})$ and $\lambda > 0$ (on boundary, $>$ for maximizing f).
  - For both, $\lambda g(\boldsymbol{x}) = 0$
- Solutions have $g(\boldsymbol{x}) \geq 0, \lambda \geq 0, \lambda g(\boldsymbol{x}) = 0$

Lagrange Multipliers - Inequality Constraints



Consider the problem:

$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$s.t. \quad g(\boldsymbol{x}) \geq 0$$

- Exactly how does the Lagrangian relate to the optimization problem in this case?

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

- It turns out that the solution to optimization problem is:

$$\max_{\boldsymbol{x}} \min_{\lambda \geq 0} L(\boldsymbol{x}, \lambda)$$

Max-min

- Lagrangian

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

- Consider the following:

$$\min_{\lambda \geq 0} L(\boldsymbol{x}, \lambda)$$

  - If the constraint $g(\boldsymbol{x}) \geq 0$ is not satisfied, $g(\boldsymbol{x}) < 0$
    - Hence, λ can be made ∞, and $\min_{\lambda \geq 0} L(\boldsymbol{x}, \lambda) = -\infty$
  - Otherwise, $\min_{\lambda \geq 0} L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x})$, (with $\lambda = 0$)
- Hence,

$$\min_{\lambda \geq 0} L(\boldsymbol{x}, \lambda) = \begin{cases} -\infty & \text{constraint not satisfied} \\ f(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

Min-max (Dual form)

- So the solution to optimization problem is:

$$L_P(\boldsymbol{x}) = \max_{\boldsymbol{x}} \min_{\lambda \geq 0} L(\boldsymbol{x}, \lambda)$$

  which is called the primal problem
- The dual problem is when one switches the order of the max and min:

$$L_D(\lambda) = \min_{\lambda \geq 0} \max_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda)$$

- These are not the same, but it is always the case the dual is a bound for the primal (in the SVM case with minimization, $L_D(\lambda) \leq L_P(\boldsymbol{x})$)
- Slater's theorem gives conditions for these two problems to be equivalent, with $L_D(\lambda) = L_P(\boldsymbol{x})$.
- Slater's theorem apples for the SVM optimization problem, and solving the dual leads to kernelization and can be easier than solving the primal

Outline

Now Where Were We

- So the optimization problem is now a constrained optimization problem:

$$\arg\min_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2}$$
$$s.t. \qquad \forall n \ , \ t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1$$

- For this problem, the Lagrangian (with N multipliers a_n) is:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{||\boldsymbol{w}||^2}{2} - \sum_{n=1}^{N} a_n \left\{ t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) - 1 \right\}$$

- We can find the derivatives of L wrt $\boldsymbol{w}, b$ and set to 0:

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n)$$
$$0 = \sum_{n=1}^{N} a_n t_n$$

Dual Formulation

- Plugging those equations into L removes $\boldsymbol{w}$ and b results in a version of L where $\nabla_{\boldsymbol{w},b} L = 0$:

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_m)$$

  this new $\tilde{L}$ is the dual representation of the problem (maximize with constraints)
  - Note that it is kernelized
  - It is quadratic, convex in $\boldsymbol{a}$
  - Bounded above since $\boldsymbol{K}$ positive semi-definite
  - Optimal $\boldsymbol{a}$ can be found
    - With large datasets, descent strategies employed

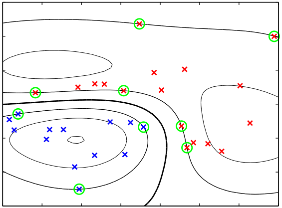
From $\boldsymbol{a}$ to a Classifier

- We found $\boldsymbol{a}$ optimizing something else
- This is related to classifier by

$$
\begin{aligned}
\boldsymbol{w} &= \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n) \\
y(\boldsymbol{x}) &= \boldsymbol{w}^T \phi(\boldsymbol{x}) + b = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b
\end{aligned}
$$

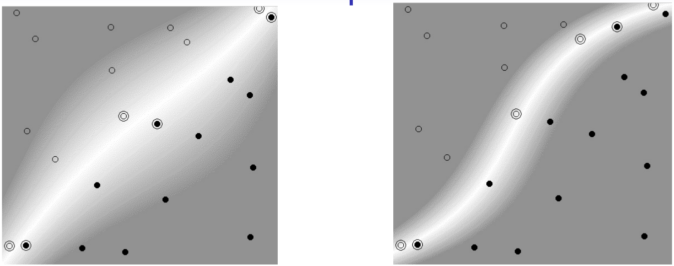
- Recall $a_n\{t_n y(\boldsymbol{x}_n) - 1\} = 0$ condition from Lagrange
  - Either $a_n = 0$ or $\boldsymbol{x}_n$ is a support vector
- $\boldsymbol{a}$ will be sparse - many zeros
  - Don't need to store $\boldsymbol{x}_n$ for which $a_n = 0$
- Another formula for finding b

Examples



- SVM trained using Gaussian kernel
- Support vectors circled
- Note non-linear decision boundary in x space

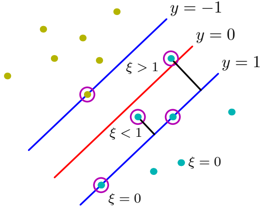
Examples



- From Burges, *A Tutorial on Support Vector Machines for Pattern Recognition* (1998)
- SVM trained using cubic polynomial kernel $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \boldsymbol{x}_2 + 1)^3$
- Left is linearly separable
  - Note decision boundary is almost linear, even using cubic polynomial kernel
- Right is not linearly separable
  - But is separable using polynomial kernel

Outline

Non-Separable Data

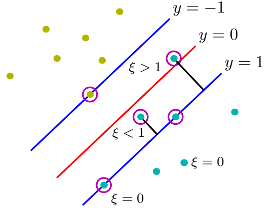


- For most problems, data will not be linearly separable (even in feature space ϕ)
- Can relax the constraints from

$$t_n y(\boldsymbol{x}_n) \geq 1 \quad \text{to} \quad t_n y(\boldsymbol{x}_n) \geq 1 - \xi_n$$

- The $\xi_n \geq 0$ are called slack variables
  - $\xi_n = 0$, satisfy original problem, so x_n is on margin or correct side of margin
  - $0 < \xi_n < 1$, inside margin, but still correctly classifed
  - $\xi_n > 1$, mis-classified

Loss Function For Non-separable Data



- Non-zero slack variables are bad, penalize while maximizing the margin:

$$\min C \sum_{n=1}^{N} \xi_n + \frac{1}{2}||\boldsymbol{w}||^2$$

- Constant $C > 0$ controls importance of large margin versus incorrect (non-zero slack)
  - Set using cross-validation
- Optimization is same quadratic, different constraints, convex

SVM Loss Function

- The SVM for the separable case solved the problem:

$$\arg\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2$$
$$s.t. \quad \forall n \ , \ t_n y_n \geq 1$$

- Can write this as:

$$\arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} E_\infty(t_n y_n - 1) + \lambda||\boldsymbol{w}||^2$$

where $E_\infty(z) = 0$ if $z \geq 0$, ∞ otherwise

- Non-separable case relaxes this to be:

$$\arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} E_{SV}(t_n y_n - 1) + \lambda||\boldsymbol{w}||^2$$

where $E_{SV}(t_n y_n - 1) = [1 - y_n t_n]_+$ hinge loss
  - $[u]_+ = u$ if $u \geq 0$, 0 otherwise

SVM Loss Function

- The SVM for the separable case solved the problem:

$$\arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2$$
$$s.t. \quad \forall n\ ,\ t_n y_n \geq 1$$

- Can write this as:

$$\arg\min_{\mathbf{w}} \sum_{n=1}^{N} E_\infty(t_n y_n - 1) + \lambda||\mathbf{w}||^2$$

where $E_\infty(z) = 0$ if $z \geq 0$, ∞ otherwise

- Non-separable case relaxes this to be:

$$\arg\min_{\mathbf{w}} \sum_{n=1}^{N} E_{SV}(t_n y_n - 1) + \lambda||\mathbf{w}||^2$$

where $E_{SV}(t_n y_n - 1) = [1 - y_n t_n]_+$ hinge loss

- $[u]_+ = u$ if $u \geq 0$, 0 otherwise

SVM Loss Function

- The SVM for the separable case solved the problem:

$$\arg\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2$$
$$s.t. \qquad \forall n\ ,\ t_n y_n \geq 1$$

- Can write this as:

$$\arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} E_{\infty}(t_n y_n - 1) + \lambda||\boldsymbol{w}||^2$$

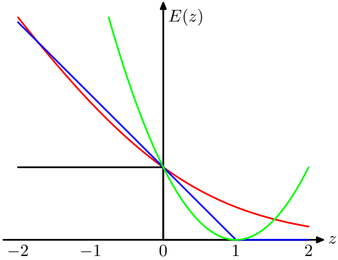
where $E_{\infty}(z) = 0$ if $z \geq 0$, ∞ otherwise

- Non-separable case relaxes this to be:

$$\arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} E_{SV}(t_n y_n - 1) + \lambda||\boldsymbol{w}||^2$$

where $E_{SV}(t_n y_n - 1) = [1 - y_n t_n]_+$ hinge loss
  - $[u]_+ = u$ if $u \geq 0$, 0 otherwise

Loss Functions



- Linear classifiers, compare loss function used for learning
  - Black is misclassification error
  - Simple linear classifier, squared error: $(y_n - t_n)^2$
  - Logistic regression, cross-entropy error: $t_n \ln y_n$
  - SVM, hinge loss: $\xi_n = [1 - y_n t_n]_+$

Conclusion

- Readings: Ch. 7 up to and including Ch. 7.1.2
- Maximum margin criterion for deciding on decision boundary
  - Linearly separable data
- Relax with slack variables for non-separable case
- Global optimization is possible in both cases
  - Convex problem (no local optima)
  - Descent methods converge to global optimum
- Kernelized