LECTURE 21: Support Vector Machines

- Empirical Risk Minimization
- The VC dimension
- Structural Risk Minimization
- Maximum margin hyperplane
- The Lagrangian dual problem

Introduction (1)

- **Consider the familiar problem of learning a binary classification problem from data**
  - Assume a given a dataset $(X,Y)=\{(x_1,y_1),(x_2,y_2),\ldots(x_N,y_N)\}$, where the goal is to learn a function $y=f(x)$ that will correctly classify unseen examples
- **How do we find such function?**
  - By optimizing some measure of performance of the learned model
- **What is a good measure of performance?**
  - As we saw in Lecture 4, a good measure is the ***expected risk***

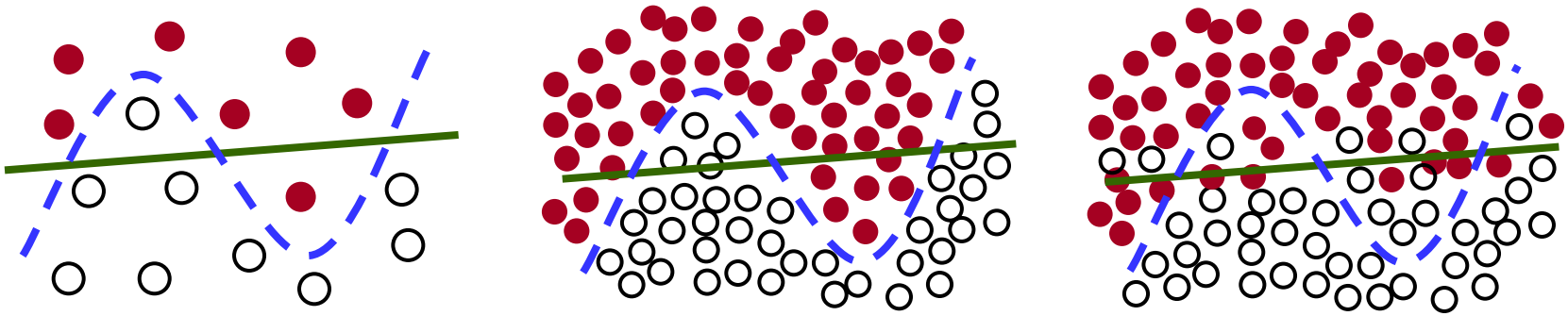
$$R[f]=\int C(f(x),y)dP(x,y)$$

    - where C(f,y) is a suitable cost function, such as the squared error $C(f,y)=(f(x)-y)^2$
  - Unfortunately, the risk cannot be measured directly since the underlying pdf is unknown. Instead, we typically use the risk over the training set, also known as the ***empirical risk***

$$R_{emp}[f]=\frac{1}{N}\sum_{i=1}^{N}C(f(x_i),y_i)$$

Introduction (2)

- **Empirical Risk Minimization**
  - A formal term for a simple concept: find the function f(x) that minimizes the average risk on the training set
  - Minimizing the empirical risk is not a bad thing to do, provided that sufficient training data is available, since the law of large numbers ensures that the empirical risk will asymptotically converge to the expected risk for $n \rightarrow \infty$
  - However, for small samples, one cannot guarantee that ERM will also minimize the expected risk. This is the all too familiar issue of generalization
- **How do we avoid overfitting?**
  - By controlling model complexity. Intuitively, we should prefer the simplest model that explains the data (Occam's razor)



From [Müller et al., 2001]

The VC dimension (1)

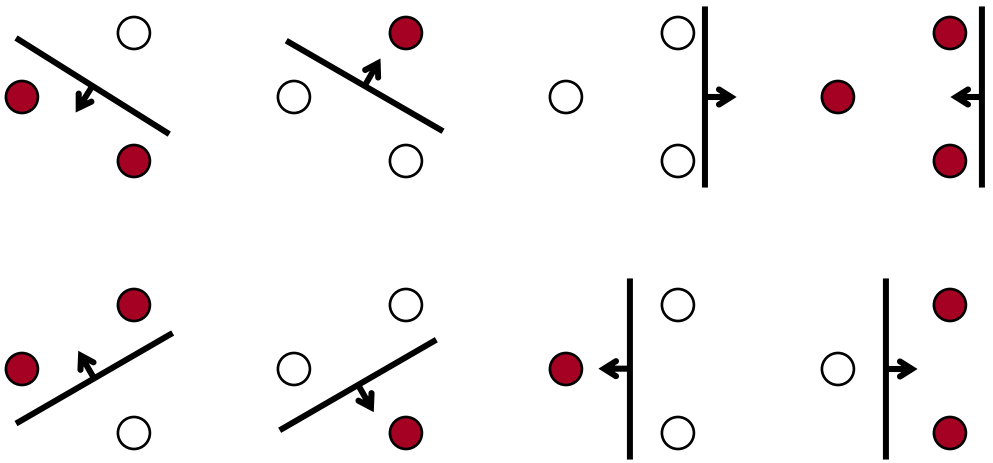
- **The Vapnik-Chervonenkis dimension is a measure of the complexity (or capacity) of a class of functions f(α)**
  - The VC dimension measures the largest number of examples that can be explained by the family f(α)
- **The basic argument is that high capacity and generalization properties are at odds**
  - If the family f(α) has enough capacity to explain every possible dataset, we should not expect these functions to generalize very well
  - On the other hand, if functions f(α) have small capacity but they are able to explain our particular dataset, we have stronger reasons to believe that they will also work well on unseen data

The VC dimension (2)

- **Shattering a set of examples**
  - Assume a binary classification problem with N examples in R^D and consider the set of $2^{|N|}$ possible dichotomies
    - For instance, with N=3 examples, the set of all possible dichotomies is {(000), (001), (010), (011), (100), (101), (110), (111)}
  - A class of functions f(α) is said to **shatter** the dataset if, for every possible dichotomy, there is a function in f(α) that models it
- **The VC dimension**
  - **The VC dimension VC(f) is the size of the largest dataset that can be shattered by the set of functions f(α)**
  - If the VC dimension of f(α) is h, then there exists at least one set of h points that can be shattered by f(α), but in general it will not be true that every set of h points can be shattered
    - One may even find a set of N<h points that cannot be shattered by this set of functions

The VC dimension (3)

- **Consider a binary classification problem in R^2, and let $f(\alpha)$ be the family of oriented hyperplanes (e.g., perceptrons)**
  - For N=3, one can perform a linear separation of all points for every possible class assignment (see examples below)
  - For N=4, a hyperplane cannot separate all possible class assignments (e.g., consider the XOR problem)
  - **Therefore, the VC dimension of the set of oriented lines in R^2 is three**
  - It can be shown that the VC dimension of the family of oriented separating hyperplanes in R^D is at least D+1



From [Burges, 1998]

The VC dimension (4)

- **The VC dimension and the number of free model parameters**
  - One may intuitively expect that models with a large number of free parameters would have high VC dimension, whereas models with few parameters would have low VC dimensions
  - Counter example
    - Consider the one-parameter function $f(x,\alpha)=\text{sign}(\sin(\alpha x))$, $\forall x,\alpha \in R$
    - You choose an arbitrary number h (as large as you want)
    - I choose the set of examples $x_i=10^{-i}$, i=1…h
    - You choose any labels you like $y_1, y_2,\ldots y_h$; $x_i \in \{-1,+1\}$
    - I choose α to be

$$\alpha = \pi\left(1+\sum_{i=1}^{h}\frac{(1-y_i)10^i}{2}\right)$$

      - Despite having only one parameter, the function $f(x,\alpha)$ shatters an arbitrarily large number of points chosen according to the outlined procedure
      - And, at the same time, one can find four points that cannot be shattered by this function!
- **So what do we make of this?**
  - The VC dimension is a more "sophisticated" measure of model complexity than dimensionality or number of free parameters [Pardo,2000]

Structural Risk Minimization (1)

- **Why is the VC dimension relevant?**
  - Because the VC dimension provides bounds on the expected risk as a function of the empirical risk and the number of available examples
  - It can be shown that, with probability 1-η, the following bound holds

$$R(f) \leq R_{emp}(f) + \underbrace{\sqrt{\frac{h(\ln(2N/h)+1)-\ln(\eta/4)}{N}}}_{\text{VC confidence}} \qquad \textbf{Eq. (1)}$$

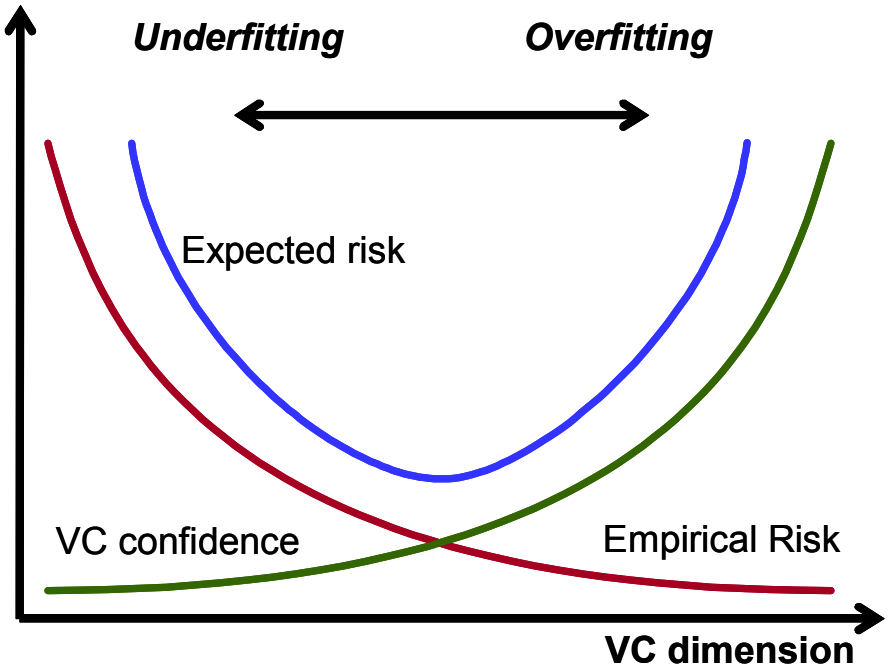
    - where h is the VC dimension of f(α), N is the number of training examples, and N>h
  - As the ratio N/h gets larger, the VC confidence becomes smaller and the actual risk becomes closer to the empirical risk
    - Therefore, this expression is consistent with the intuition that ERM is only suitable when sufficient data is available
  - This and other results are part of the field known as **Statistical Learning Theory** or Vapnik-Chervonenkis Theory, from which Support Vector Machines originated

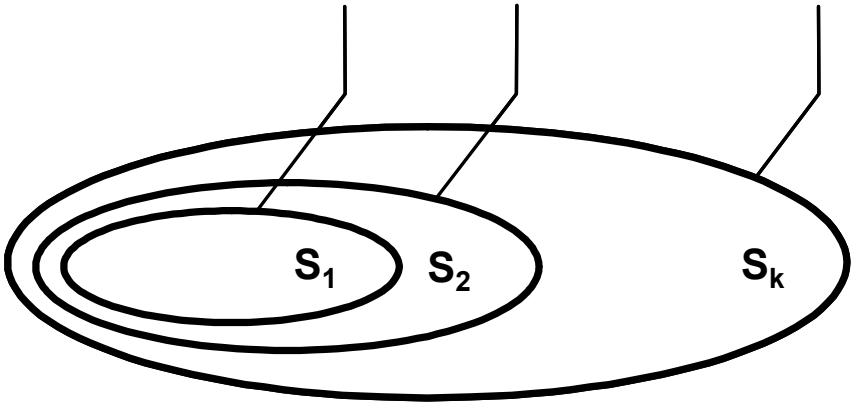
From [Burges, 1998]

Structural Risk Minimization (2)

- **Structural Risk Minimization**
  - Another formal term for an intuitive concept: the optimal model is found by striking a balance between the empirical risk and the VC dimension
- **The SRM principle proceeds as follows**
  - Construct a nested *structure* for family of function classes $F_1 \subset F_2 \subset \ldots \subset F_k$ with non-decreasing VC dimensions ($h_1 \leq h_2 \leq \ldots \leq h_k$)
  - For each class F_i, compute the solution f_i that minimizes the empirical risk
  - Choose the function class F_i, and the corresponding solution f_i, that minimizes the risk bound on the RHS of equation (1)
- **In other words**
  - Train a set of machines, one for each subset
  - For a given subset, train to minimize the empirical risk
  - Choose the machine whose sum of empirical risk and VC confidence is minimum

Structural Risk Minimization (3)





From [Cherkassky and Mulier, 1998]

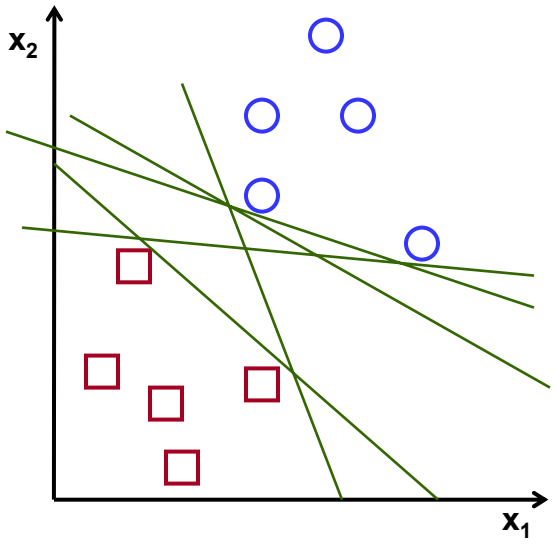
The VC dimension in practice

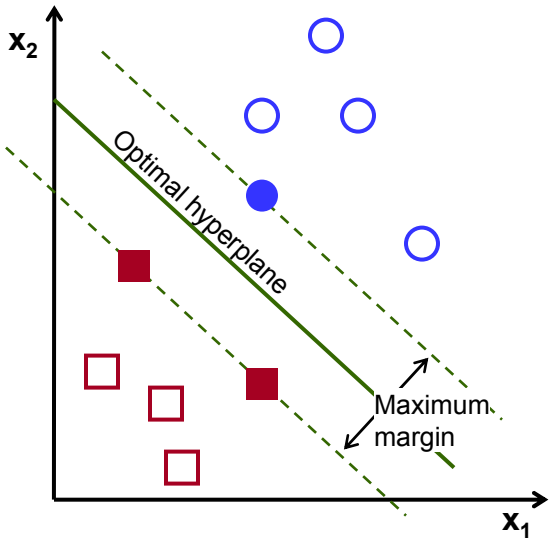
- **Unfortunately, computing an upper bound on the expected risk is not practical in various situations**
  - The VC dimension cannot be accurately estimated for non-linear models such as neural networks
  - Implementation of Structural Risk Minimization may lead to a non-linear optimization problem
  - The VC dimension may be infinite (e.g., k=1 nearest neighbor), requiring infinite amount of data or
  - The upper bound may sometimes be trivial (e.g., larger than one)
- **Fortunately, Statistical Learning Theory can be rigorously applied in the realm of linear models**

From [Cherkassky and Mulier, 1998; Müller et al., 2001]

Optimal separating hyperplanes (1)

- **Consider the problem of finding a separating hyperplane for a linearly separable dataset $\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$, $x \in R^D$, $y \in \{-1,+1\}$**
  - Which of the infinite hyperplanes should we choose?
    - Intuitively, a hyperplane that passes too close to the training examples will be sensitive to noise and, therefore, less likely to generalize well for data outside the training set
    - Instead, it seems reasonable to expect that a hyperplane that is farthest from all training examples will have better generalization capabilities
  - Therefore, the optimal separating hyperplane will be the one with the largest **margin**, *which is defined as the minimum distance of an example to the decision surface*





From [Cherkassky and Mulier, 1998]

Optimal separating hyperplanes (2)

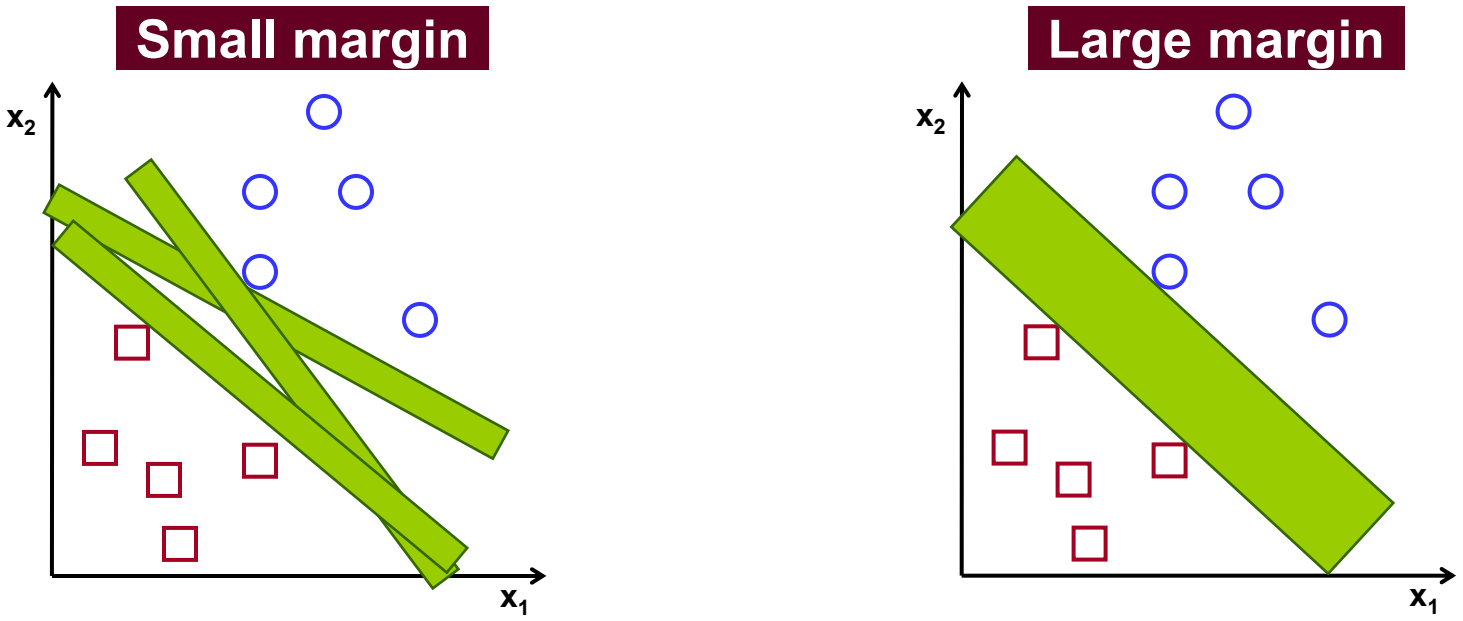
- **How does this intuitive result relate to the VC dimension?**
  - It can be shown [Vapnik, 1998] that the VC dimension of a separating hyperplane with a margin *m* is bounded as follows

$$h \leq \min\left(\left\lceil \frac{R^2}{m^2} \right\rceil, D\right) + 1$$

    - where D is the dimensionality of the input space, and R is the radius of the smallest sphere containing all the input vectors
  - Therefore, by maximizing the margin we are in fact minimizing the VC dimension
  - And, since the separating hyperplane has zero empirical error (it correctly separates all the training examples), maximizing the margin will also minimize the upper bound on the expected risk
- **Conclusion**
  - **The separating hyperplane with maximum margin will also minimize the structural risk**

Optimal separating hyperplanes (3)

- **To further understand the relationship between margin and capacity, consider the two separating hyperplanes depicted below**
  - A "skinny" one (small margin), which will be able to adopt many orientations
  - A "fat" one (large margin), which will have limited flexibility
- **A larger margin necessarily results in lower capacity**
  - We normally think of complexity as being a function of the number of parameters
    - Instead, Statistical Learning Theory tells us that if the margin is sufficiently large, the complexity of the function will be low even if the dimensionality is very high!



From [Bennett and Campbell, 2000]

Optimal separating hyperplanes (4)

- **Since we want to maximize the margin, let's express it as a function of the weight vector and bias of the separating hyperplane**
  - From basic trigonometry, the distance between a point x and a plane (w,b) is

$$\frac{\left|w^T x + b\right|}{\|w\|}$$

  - Noticing that the optimal hyperplane has infinite solutions by simply scaling the weight vector and bias, we choose the solution for which the discriminant function becomes one for the training examples closest to the boundary

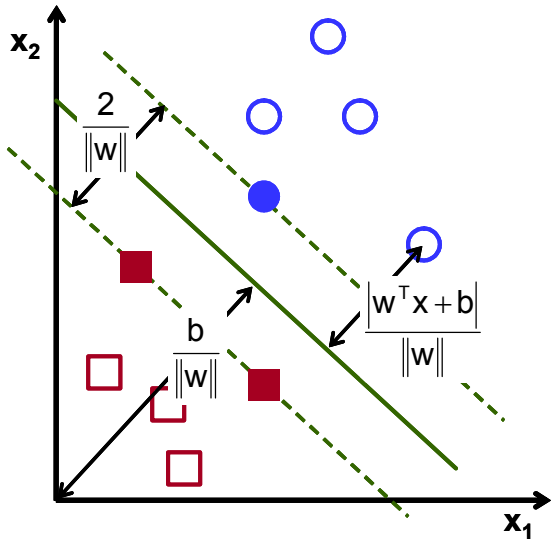
$$\left|w^T x_i + b\right| = 1$$

    - This is known as the *canonical* hyperplane
  - Therefore, the distance from the closest example to the boundary is

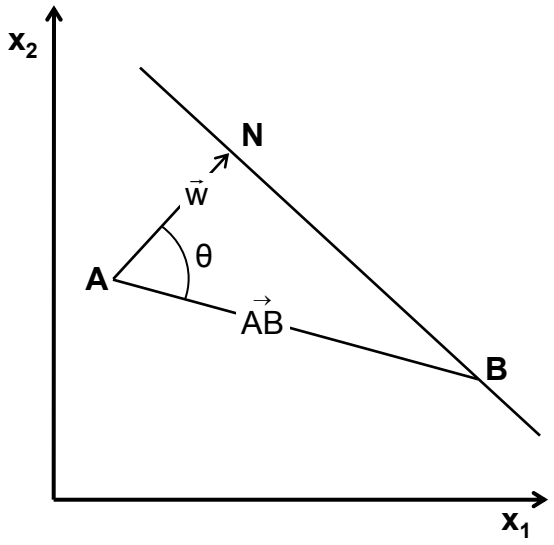
$$\frac{\left|w^T x + b\right|}{\|w\|} = \frac{1}{\|w\|}$$

  - And the margin becomes

$$m = \frac{2}{\|w\|}$$



(Distance between a plane and a point)



$$\|AN\| = \|AB\|\cos\theta = \|AB\| \frac{\vec{AB}\,\vec{w}}{\|AB\|\ \|w\|} = \frac{\vec{AB}\,\vec{w}}{\|w\|}$$

$$= \frac{(x_{1A} - x_{1B}, x_{2A} - x_{2B})^T (w_1, w_2)}{\|w\|} =$$

$$= \frac{w^T x_A - \overbrace{w^T x_B}^{=-b}}{\|w\|} = \frac{w^T x_A + b}{\|w\|}$$

Optimal separating hyperplanes (5)

- **Therefore, the problem of maximizing the margin is equivalent to**

$$\begin{aligned} &\text{minimize} && J(w) = \frac{1}{2}\|w\|^2 \\ &\text{subject to} && y_i\left(w^T x_i + b\right) \geq 1 \;\; \forall i \end{aligned}$$

  - Notice that J(w) is a quadratic function, which means that there exists a single global minimum and no local minima
- **To solve this problem, we will use classical Lagrangian optimization techniques**
  - We first present the Kuhn-Tucker Theorem, which provides an essential result for the interpretation of Support Vector Machines

(Kuhn-Tucker Theorem)

- **Given an optimization problem with convex domain $\Omega \subseteq R^N$**

$$\begin{aligned} &\text{minimize} && f(z) && z \in \Omega \\ &\text{subject to} && g_i(z) \le 0 && i = 1,\ldots,k \\ &&& h_i(z) = 0 && i = 1,\ldots,m \end{aligned}$$

  - with $f \in C^1$ convex and g_i, h_i affine, necessary and sufficient conditions for a normal point z^* to be an optimum are the existence of α^*, β^* such that

$$\begin{aligned} &\frac{\partial L(z^*,\alpha^*,\beta^*)}{\partial z} = 0 \\ &\frac{\partial L(z^*,\alpha^*,\beta^*)}{\partial \beta} = 0 \\ &\alpha_i^* g_i(z^*) = 0 && i = 1,\ldots,k \\ &g_i(z^*) \le 0 && i = 1,\ldots,k \\ &\alpha_i^* \ge 0 && i = 1,\ldots,k \end{aligned}$$

where $L(z,\alpha,\beta) = f(z) + \sum_{i=1}^{k} \alpha_i g_i(z) + \sum_{i=1}^{m} \beta_i h_i(z)$

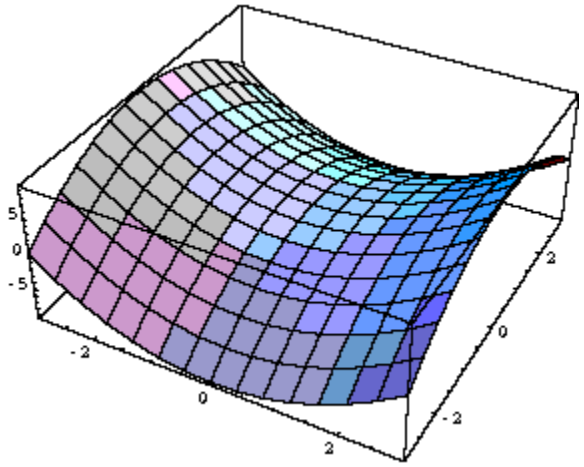
    - $L(z,\alpha,\beta)$ is known as a *generalized Lagrangian function*
    - The third condition is know as the Karush-Kuhn-Tucker (KKT) complementary condition. It implies that for active constraints $\alpha_i \ge 0$; and for inactive constraints $\alpha_i = 0$
      - As we will see in a minute, the KKT condition allows us to identify the training examples that define the largest margin hyperplane. These examples will be known as **Support Vectors**.

The Lagrangian dual problem (1)

- **Constrained minimization of $J(w)=1/2||w||^2$ is solved by introducing the Lagrangian**

$$L_P(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i \left(w^T x_i + b \right) - 1 \right]$$

  - which yields an unconstrained optimization problem that is solved by:
    - minimizing L_P with respect to the primal variables w and b, and
    - maximizing L_P with respect to the dual variables $\alpha_i \geq 0$ (the Lagrange multipliers)
  - Thus, the optimum is defined by a saddle point (see below for illustration)
- **This is known as the Lagrangian primal problem**



A saddle point

From [Haykin, 1999]

The Lagrangian dual problem (2)

- **To simplify the primal problem, we eliminate the primal variables (w,b) using the first Kuhn-Tucker condition $\partial J/\partial z=0$**
  - Differentiating $L_P(w,b,\alpha)$ with respect to w and b, and setting to zero yields

$$\frac{\partial L_P(w,b,\alpha)}{\partial w}=0 \quad \Rightarrow w=\sum_{i=1}^{N}\alpha_i y_i x_i$$
$$\frac{\partial L_P(w,b,\alpha)}{\partial b}=0 \quad \Rightarrow \sum_{i=1}^{N}\alpha_i y_i = 0$$

  - Expansion of L_P yields

$$L_P(w,b,\alpha)=\frac{1}{2}w^T w - \sum_{i=1}^{N}\alpha_i y_i w^T x_i - b\sum_{i=1}^{N}\alpha_i y_i + \sum_{i=1}^{N}\alpha_i$$

  - Using the optimality condition $\partial J/\partial w=0$, the first term in L_P can be expressed as

$$w^T w = w^T \sum_{i=1}^{N}\alpha_i y_i x_i = \sum_{i=1}^{N}\alpha_i y_i w^T x_i = \sum_{i=1}^{N}\alpha_i y_i \left(\sum_{j=1}^{N}\alpha_j y_j x_j\right)^T x_i = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

  - The second term in L_P can be expressed in the same way
  - The third term in L_P is zero by virtue of the optimality condition $\partial J/\partial b=0$

The Lagrangian dual problem (3)

- Merging these expressions together we obtain

$$L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

  - Subject to the (simpler) constraints $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$

- **This is known as the Lagrangian dual problem**

- **Comments**
  - We have transformed the problem of finding a saddle point for $L_P(w,b)$ into the easier one of maximizing $L_D(\alpha)$
    - Notice that $L_D(\alpha)$ depends on the Lagrange multipliers α, not on (w,b)
  - The primal problem scales with dimensionality (w has one coefficient for each dimension), whereas the dual problem scales with the amount of training data (there is one Lagrange multiplier per example)
  - Moreover, in $L_D(\alpha)$ the training data appears only as dot products $x_i^T x_j$
    - As we will see in the next lecture, this property can be cleverly exploited to perform the classification in a higher (e.g., infinite) dimensional space

Support Vectors

- **The KTT complementary condition states that, for every point in the training set, the following equality must hold**

$$\alpha_i\left[y_i\left(w^T x_i + b\right) - 1\right] = 0 \quad \forall i = 1 \ldots N$$

  - Therefore, for each example, either α_i=0 or $y_i(w^T x_i$+b-1)=0 must hold
  - Those points for which α_i>0 must then lie on one of the two hyperplanes that define the largest margin (only at these hyperplanes the term $y_i(w^T x_i$+b-1) becomes zero)
    - These points are known as the **Support Vectors**
  - All the other points must have α_i=0
  - Note that only the support vectors contribute to defining the optimal hyperplane

$$\frac{\partial J(w,b,\alpha)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

    - NOTE: the bias term b is found from the KKT complementary condition on the support vectors
  - Therefore, the complete dataset could be replaced by only the support vectors, and the separating hyperplane would be the same