# Linear and Combinatorial Optimization

Sara Maad Sasane

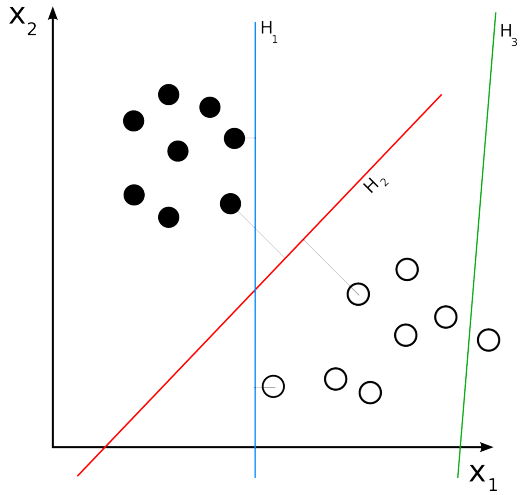
Center for Mathematical Sciences, Lund University

8 March 2019

# 1 Application of LP: Linear classification

# Linear classification

- In machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to.
- Each object (point) $\mathbf{x_i}$ in a training set has a binary label $s_i \in \{-1, 1\}$ (which identifies which of two groups the object belongs to).
- The aim is to find a hyperplane $\mathbf{a^T x} + b = 0$ which strictly separates the two groups.
- The purpose is to use the same classifier (hyperplane) on new points which are not in the training set, to determine which group they belong to.




The two types of dots can be correctly classified by many different linear classifiers. H1 (blue) and H2 (red) classifies them correctly, whereas H3 (green) fails.

# Linear classification

- By correctly classifying the points of the two groups, we mean that

$$\begin{cases} \mathbf{a}^{\mathbf{T}}\mathbf{x_i} + b > 0 & \text{if } s_i = 1, \\ \mathbf{a}^{\mathbf{T}}\mathbf{x_i} + b < 0 & \text{if } s_i = -1. \end{cases}$$

  for $i = 1, \ldots, N$.

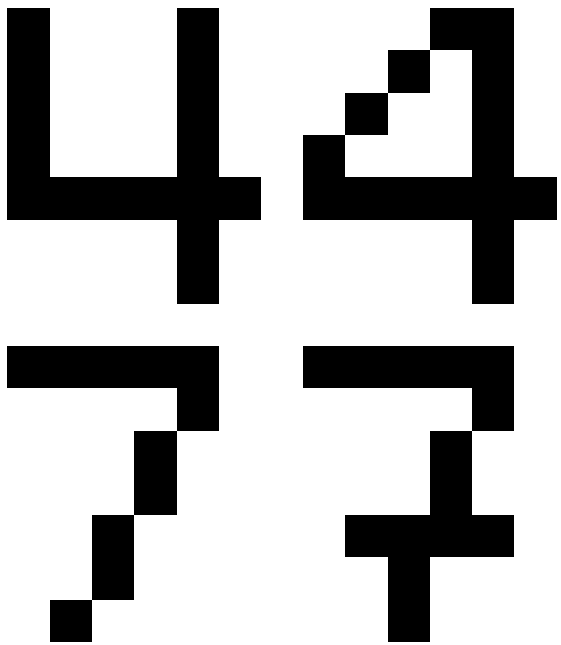
- The above can equivalently be expressed as

$$s_i(\mathbf{a}^{\mathbf{T}}\mathbf{x_i} + b) > 0$$

  for $i = 1, \ldots, N$.

# Linear classification

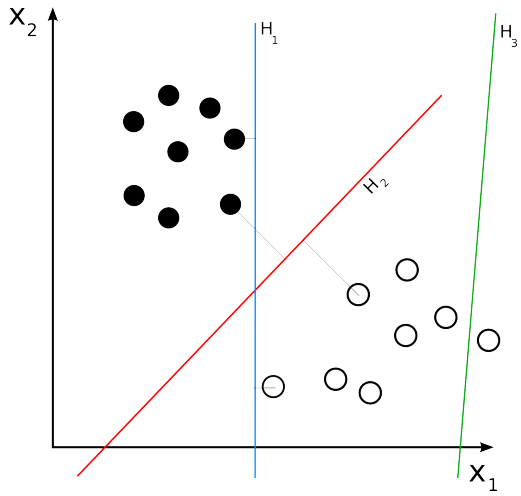
- The method will be explained with the help of a handwriting recognition example.
- Let us say that we would like to distinguish handwritten numbers, and that we only care about whether the numbers should be classified as a 7 or as a 4.
- The images of numbers are represented as points in an $N$-dimensional space, where $N = n^2$ (giving an image of size $n \times n$), where each of the $N$ variables can be 0 or 1.



Different versions of fours and sevens, in small images of size $8 \times 8$

# Linear classification

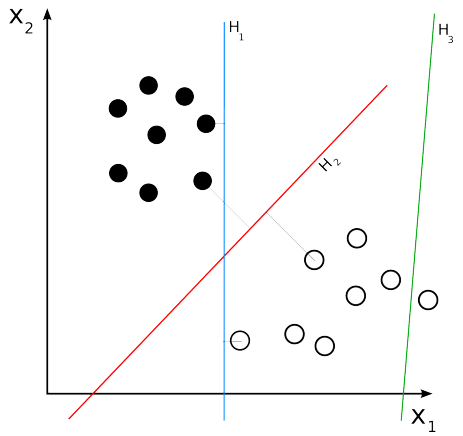
- Each pixel corresponds to one variable which can be 0 (white) or 1 (black). The linear space that we work in is then $N$-dimensional.
- Hence we would like to find a separating hyperplane in an $N$-dimensional space which separates the fours of the training set from the sevens.
- When such a hyperplane has been found, and we are given a new image, we can check on which side of the hyperplane it falls, to determine whether it should be classified as a 7 or as a 4.




Think of the sevens as being represented by the black dots and the fours as being represented by the white dots.

# Linear classification

- A hyperplane is described by a linear equation $\mathbf{a}^\mathbf{T}\mathbf{x} - b = 0$, where $\mathbf{a}, \mathbf{x} \in \mathbb{R}^N$ and $b \in \mathbb{R}$.
- We try to find a separating hyperplane like the red or blue one in the figure to the right.
- It turns out that if there exists one such hyperplane, then there will be a lot of them.
- We then try to find the "best" such hyperplane. The purpose is that there should be as few classification errors as possible, when new points are classified.




A situation where there are many possible hyperplanes that separates the two groups of points. The red one is considered "better" than the blue one, since the margin to the training points is larger.

# Linear classification

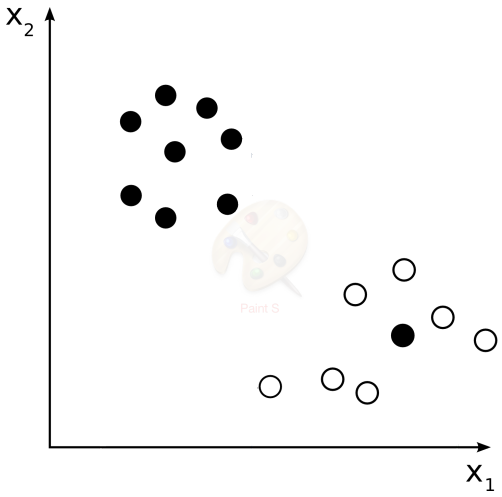
- If the problem of finding a separating hyperplane is feasible, it will be feasible with a margin. This follows from the fact that there will only be finitely many points in the training set.
- For this reason, we get the same thing if we replace the inequality
  - $\mathbf{a}^\mathsf{T}\mathbf{x} - b > 0$ by $\mathbf{a}^\mathsf{T}\mathbf{x} - b \geq \epsilon$ and
  - $\mathbf{a}^\mathsf{T}\mathbf{x} - b < 0$ by $\mathbf{a}^\mathsf{T}\mathbf{x} - b \leq -\epsilon$.

  where $\epsilon$ is some positive number.
- By rescaling $\mathbf{a}$ and $b$, we can take $\epsilon = 1$.
- Hence, we would now like to find $\mathbf{a}$ and $b$ such that
  - $\mathbf{a}^\mathsf{T}\mathbf{x} - b \geq 1$ for all $\mathbf{x}$ classified as 4s, and
  - $\mathbf{a}^\mathsf{T}\mathbf{x} - b \leq -1$ for all $\mathbf{x}$ classified as 7s.

# Linear classification

- Note that the variables in the problem are the components of $\mathbf{a}$ and $b$, and so there are $N + 1$ variables and $m + r$ inequality constraints, where
  - $m$ is the number of images in the training set which are classified as 4s, and
  - $r$ is the number of images in the training set which are classified as 7s.
- In a perfect world, we would be content with solving this feasibility problem (a system of linear inequalities), but only if the training set is perfect and without any errors or ambiguities.

# Linear classification




A situation where there is no separating hyperplane. This could occur if e.g. a 7 is wrongly classified as a 4, or if somebody just has unusually bad handwriting.

- What if there was a wrongly classified 7 in the cloud of 4s for example? There is no separating hyperplane, and the problem becomes infeasible.
- This is serious problem, since in real world applications, there will always be errors and ambiguities.
- The problem can be resolved by allowing for some errors (but trying to keep them small).

# Linear classification

Introduce slack variables, $u_i$, $v_j$ ($i = 1, \ldots, m$ and $j = 1, \ldots, r$), and solve instead

$$\text{minimize} \sum_{i=1}^{m} u_j + \sum_{j=1}^{r} v_j$$

$$\text{subject to} \begin{cases} \mathbf{a}^{\mathsf{T}}\mathbf{x_i} - b \geq 1 - u_i, & i = 1, \ldots, m, \\ \mathbf{a}^{\mathsf{T}}\mathbf{x_{m+j}} - b \leq -1 + v_j, & j = 1, \ldots, r, \\ \mathbf{a} \in \mathbb{R}^N, b \in \mathbb{R}, & \\ u_i \geq 0, v_j \geq 0, & i = 1, \ldots, m, j = 1, \ldots, r, \end{cases}$$

where $\mathbf{x_1}, \ldots, \mathbf{x_m}$ are the images that have been classified as 4s and $\mathbf{x_{m+1}}, \ldots, \mathbf{x_{m+r}}$ the ones that have been classified as 7s.

# Linear classification

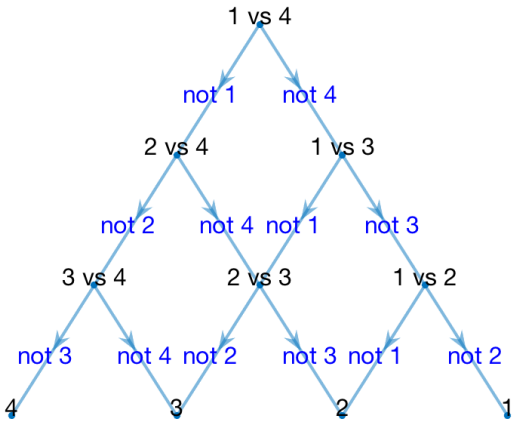
- Note that if a separating hyperplane exists, then all $u_j$ and $v_j$ will be 0 for the optimal solution. The problem can now be solved even with some wrongly classified numbers.
- An equivalent way of writing the problem is

$$\text{minimize } (\mathbb{1}^T \mathbf{u} + \mathbb{1}^T \mathbf{v})$$

$$\text{subject to } \begin{cases} \mathbf{X_4 a} + \mathbf{u} \geq (1+b)\mathbb{1}, \\ \mathbf{X_7 a} - \mathbf{v} \leq (-1+b)\mathbb{1}, \\ \mathbf{a} \in \mathbb{R}^N, b \in \mathbb{R}, \\ \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}, \end{cases}$$

where $\mathbf{u}$ and $\mathbf{v}$ are vectors in $\mathbb{R}^m$ and $\mathbb{R}^r$, respectively, and $\mathbf{X_4}$ is the $m \times N$-matrix whose rows are $\mathbf{x_i}$, $i = 1, \ldots, m$, and $\mathbf{X_7}$ is the $r \times N$-matrix whose rows are $\mathbf{x_{m+j}}$, $j = 1, \ldots, r$. $\mathbb{1}$ is a column vector of length $m$ or $r$, whose entries are all 1.

# Linear classification




The digits compete against each other pairwise. The image is classified as a certain digit when all other options are ruled out.

- The LP problem can now be solved, e.g. with the simplex algorithm.
- The method can be extended to a complete handwriting algorithm as the figure to the left indicates where we only consider the digits $1, \ldots, 4$.
- With four digits, we need to determine $6 = \frac{4 \cdot 3}{2}$ classifiers (hyperplanes).
- With all ten digits, we would need $45 = \frac{10 \cdot 9}{2}$ classifiers.

# Linear classification

- The above algorithm (idea) is taken from the paper Large Margin DAG's for Multiclass Classification by Platt, Cristianini and Shawe–Taylor.
- Other applications using separating hyperplanes include spam filters, etc. (Quadratic optimization is required).
- The handwriting recognition algorithm described above can also be improved with quadratic optimization.


The word spam (for junk mail) comes from a sketch by Monty Python.

- For some more applications, see this list.