Machine Learning

Q&A

Fabio Vandin

January 12, 2024

Question

From Nov. 20^{th} lecture: SVM

- slide 16-17: definition of support vectors, proposition.
- slide 19, what are "slack variables"?

Equivalent Formulation and Support Vectors

Equivalent formulation (homogeneous halfspaces): assume first component of $\mathbf{x} \in \mathcal{X}$ is 1, then

$$\mathbf{w}_0 = \arg\min_{\mathbf{w}} ||\mathbf{w}||^2 \text{ subject to } \forall i : y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$$

"Support Vectors" = vectors at minimum distance from $\mathbf{w}_0$

The support vectors are the only ones that matter for defining $\mathbf{w}_0$!

Proposition

Let $\mathbf{w}_0$ be as above. Let $I = \{i : |\langle \mathbf{w}_0, \mathbf{x}_i \rangle| = 1\}$. Then there exist coefficients $\alpha_1, \ldots, \alpha_m$ such that

$$\mathbf{w}_0 = \sum_{i \in I} \alpha_i \mathbf{x}_i$$

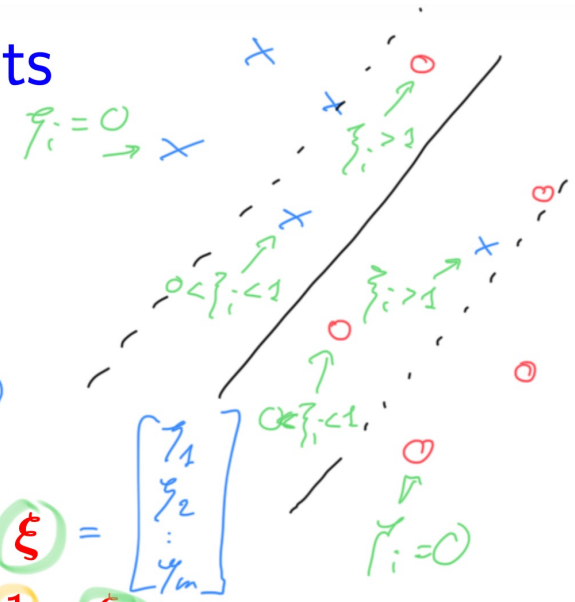
"Support vectors" = $\{\mathbf{x}_i : i \in I\}$

Note: Solving Hard-SVM is equivalent to find α_i for $i = 1, \ldots, m$, and $\alpha_i \neq 0$ only for support vectors

Soft-SVM Constraints

Hard-SVM constraints:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$$



Soft-SVM constraints:

- *slack variables*: $\xi_1, \ldots, \xi_m \geq 0 \Rightarrow$ vector $\boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{bmatrix}$
- for each $i = 1, \ldots, m$: $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$
- ξ_i: how much constraint $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ is violated

Soft-SVM minimizes combinations of

- norm of $\mathbf{w}$
- average of ξ_i

Tradeoff among two terms is controlled by a parameter
$\lambda \in \mathbb{R}, \lambda > 0$

From the ‘Material to study” document:

- definition of support vectors for hard-SVM: all details
- soft-SVM optimization problem, hinge loss: all details

Question

Brief review of quadratic programming formulation for hard SVM and brief review of the dual problem again for hard SVM.

Hard-SVM: Quadratic Programming Formulation

- **input**: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
- **solve**:

$$(\mathbf{w}_0, b_0) = \arg\min_{(\mathbf{w},b)} ||\mathbf{w}||^2$$

subject to $\forall i : y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$

linear constraints (in $\vec{w}, b$)

- **output**: $\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{||\mathbf{w}_0||}, \hat{b} = \frac{b_0}{||\mathbf{w}_0||}$

$\Downarrow$

$||\hat{w}|| = 1$

Proposition

The output of algorithm above is a solution to the *Equivalent Formulation* in the previous slide.

How do we get a solution? Quadratic optimization problem: objective is convex quadratic function, constraints are linear inequalities $\Rightarrow$ Quadratic Programming solvers!

8

Duality

We now present (Hard-)SVM in a different way which is very useful for *kernels*.

We want to solve

$$\mathbf{w}_0 = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 \text{ subject to } \forall i : y_i\langle \mathbf{w}, \mathbf{x}_i\rangle \geq 1$$

One can prove (details in the book!) that $\mathbf{w}$ that minimizes the function above is equivalent to find $\boldsymbol{\alpha}$ that solves the *dual problem*:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m:\boldsymbol{\alpha}\geq\mathbf{0}} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \langle \mathbf{x}_j, \mathbf{x}_i\rangle \right)$$

Note:

- solution is the vector $\boldsymbol{\alpha}$ which defines the support vectors $= \{\mathbf{x}_i : \alpha_i \neq 0\}$
- $\mathbf{w}_0$ can be derived from $\boldsymbol{\alpha}$ (see previous slides!)
- dual problem requires only to compute inner products $\langle \mathbf{x}_j, \mathbf{x}_i\rangle$, does not need to consider $\mathbf{x}_i$ by itself

From the ‘Material to study” document:

- Hard-SVM: equivalent formulation, and quadratic formulation: main idea
- Hard-SVM dual formulation: main idea

Question

I have a generic question about the final exam: how much detail do we have to go into in the theoretical questions about defining certain learning tasks? For instance, point 1 of exercise 4 of the 07/02/2020 exam asks to "briefly introduce the clustering problem". I answered the question by writing only the informal definition, but the posted solution adds the mathematical definitions of training set, clusters, and distance function. So how do we know the degree of detail to use in the answers? Do we always add the mathematical definition of every concept the learning task uses? How much score does someone lose if they do not go into as much detail as expected (e.g. writing only the informal definition of clustering in the question mentioned above)?

Question

How detailed should a task be answered, wrt. defining $\mathcal{X}, \mathcal{Y}, \mathcal{H}$, h etc. (e.g. let $h : \mathcal{X} \to \{0, 1\}$ with h in $\mathcal{H}$...) ? Is it weighted a lot, or is it just minor details?

Question

3. Consider the data in the figure below where each point $\mathbf{x} \in \mathbb{R}^2$ is represented by a cross. Draw (approximately) the output of Lloyd's algorithm for $k = 3$ when

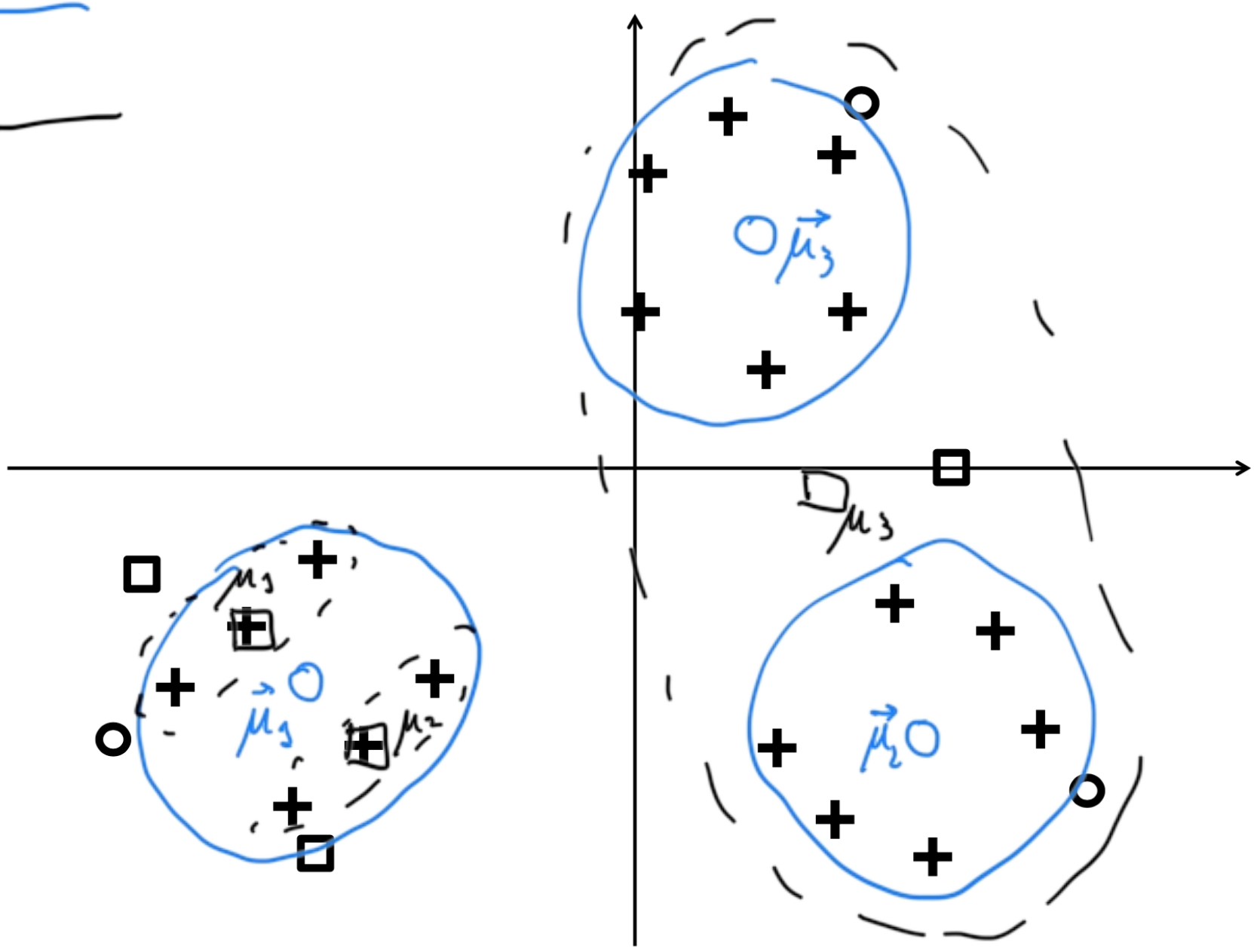
   (a) the initial centers for the algorithm are the circles;

   (b) the initial centers for the algorithm are the squares.

   Which one of the two resulting clusterings has a lower cost?

a) —

b) —



Cost: see solution to the exam posted online.

Question

"Bias-complexity trade off" topic:

- what is "prior knowledge"?
- how is it connected to the NFL theorem?

The No Free Lunch Theorem

The following answers the previous question for some specific settings.

Theorem (No-Free Lunch)

Let A be any learning algorithm for the task of binary classification with respect to the 0-1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:

- *there exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$*
- *with probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq 1/8$.*

Note: there are similar results for other learning tasks.

No Free Lunch and Prior Knowledge

Corollary

Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}$ be the set of all functions from $\mathcal{X}$ to $\{0,1\}$. Then, $\mathcal{H}$ is not PAC learnable.

What's the implication?

"what you know about the problem"

We need to use our prior knowledge about $\mathcal{D}$ to pick a *good* hypothesis set.

How do we choose $\mathcal{H}$?

- we would like $\mathcal{H}$ to be *large*, so that it may contain a function h with small $L_{\mathcal{D}}(h)$
- no free lunch $\Rightarrow$ $\mathcal{H}$ cannot be too large!

From the ‘Material to study” document:

- No Free Lunch (NFL) theorem, NFL and prior knowledge: only main idea

Question

Any exercise on NN

Question

Any exercise on NN

?

Exercise

Let

$$\mathcal{H}_d = \{h_{\mathbf{w}}(\mathbf{x}) : h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(< \mathbf{w}, \mathbf{x} >)\}$$

where $\mathcal{X} = \mathbb{R}^d$.
Prove that $VCdim(\mathcal{H}_d) = d$.

Solution We need to prove that $VC\dim(\mathcal{H}_d) \geq d$ and $VC\dim(\mathcal{H}_d) \leq d$.

i) $VC\dim(\mathcal{H}_d) \geq d$. We need to show a set of d vectors in $\mathbb{R}^d$ that is shattered by $\mathcal{H}_d$.

Consider $\{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_d\}$ with $\vec{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_i$, $1 \leq i \leq d$

Let's show that such a set is shattered by $\mathcal{H}_d$:

we need to show that for every labeling $y_1, y_2, \ldots, y_d$, where y_i is the label of $\vec{e}_i$ (with $y_i \in \{-1, 1\}$), there is an hypothesis in $\mathcal{H}_d$ that assigns such labels to the $\vec{e}_i$'s.

Consider an arbitrary labeling $y_1, y_2, \ldots, y_d$: consider the hypothesis $h_{\vec{w}}$ where $\vec{w} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix}$.

We have that for every i, $1 \leq i \leq d$:

$$h_{\vec{w}}(\vec{e}_i) = \text{sign}(\langle \vec{w}, e_i \rangle) = \text{sign}\left(\left\langle \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_d \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\rangle\right) = y_i$$

ii) $VCdim(\mathcal{H}_d) \leq d$: we need to show that no set of $d+1$ vectors in $\mathbb{R}^d$ can be shattered by $\mathcal{H}_d$.

Consider an arbitrary set $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_{d+1}\}$, $\vec{x}_i \in \mathbb{R}^d$ for $1 \leq i \leq d+1$.

They cannot be linearly independent $\Rightarrow \exists\ a_1, a_2, \dots, a_{d+1}$ with $a_i \in \mathbb{R}$ for $1 \leq i \leq d+1$, such that:

- not all a_i's are $=0$
- $\sum_{i=1}^{d+1} a_i \vec{x}_i = 0$ (★★)

Define: $I = \{i : a_i > 0\}$; $J = \{j : a_j < 0\}$.

Note that it cannot be that $I = \emptyset = J$

There are 3 cases: i) $I \neq \emptyset \neq J$; ii) $I \neq \emptyset = J$
iii) $I = \emptyset \neq J$

Case i) $[I \neq \emptyset \neq J]$

Then

$$\sum_{i \in I} a_i \vec{x}_i = \sum_{j \in J} |a_j| \vec{x}_j \quad (*)$$

Assume that $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{d+1}\}$ is shattered by $\mathcal{H}_d$: there must exist $\vec{w}$ such that

$$\langle \vec{w}, \vec{x}_i \rangle > 0 \quad \forall i \in I \quad (***)$$

$$\langle \vec{w}, \vec{x}_j \rangle < 0 \quad \forall j \in J \quad (**)$$

$$0 < \sum_{i \in I} a_i \langle \vec{w}, \vec{x}_i \rangle$$

a_i: < 0 for $(\star\star)$; $\langle \vec{w}, \vec{x}_i \rangle$: < 0

$$= \left\langle \sum_{i \in I} a_i \vec{x}_i , \vec{w} \right\rangle$$

see in $\star$

$$= \left\langle \sum_{j \in J} |a_j| \vec{x}_j , \vec{w} \right\rangle$$

$$= \sum_{j \in J} |a_j| \langle \vec{x}_j, \vec{w} \rangle \quad < 0$$

$|a_j|$: < 0; $\langle \vec{x}_j, \vec{w} \rangle$: > 0 (for $\star\star$)

$\Rightarrow$ contradiction

Case ii) $[I \neq \emptyset = J]$

same steps as case i) lead to

$0 < \ldots = 0 \Rightarrow$ contradiction

Case iii) $[I = \emptyset \neq J]$

same steps as case i) lead to

$0 = \ldots < 0 \Rightarrow$ contradiction.

□