# THE SCHLESINGER-KOZINEC ALGORITHM FOR OPTIMAL SEPARATING HYPERPLANES

The purpose of this note is to give a brief overview of the so-called Schlesinger-Kozinec algorithm for finding optimal separating hyperplanes. The exposition of this note follows mainly that of [FH03].

## 1. Background: the Kozinec algorithm

Recall that wo sets in $\mathbb{R}^n$ are *linearly separable* if there exists a hyperplane in $\mathbb{R}^n$ separating them. Any hyperplane in $\mathbb{R}^n$ can be implicitly described as the set of all points $x \in \mathbb{R}^n$ satisfying

$$\langle w, x \rangle + b = 0,$$

for some $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Given a set of labeled data points

$$\{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathbb{R}^n \times \{-1, 1\},$$

the hyperplane specified by $\langle w, x \rangle + b = 0$ separates those data points with label 1 from those with label $-1$ precisely if

$$\langle w, x_i \rangle + b \begin{cases} > 0, & \text{if } y_i = 1, \\ < 0, & \text{if } y_i = -1, \end{cases} \quad i = 1, \ldots, m.$$

These two conditions can be combined into the single requirement that

$$y_i(\langle w, x_i \rangle + b) > 0, \, i = 1, \ldots, m.$$

Defining $\omega = (w, b) \in \mathbb{R}^{n+1}$ and $\xi_i = y_i(x_i, 1) \in \mathbb{R}^{n+1}$, $i = 1, \ldots, m$, the above condition can be simplified further into

(1.1) $$\langle \omega, \xi_i \rangle > 0, \, i = 1, \ldots, m.$$

From now on, we will let $\omega$ and $\xi_i$ denote these "augmented" data points. Thus, in view of (1.1), the original task of separating the set of labeled data points $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ according to their labels by an affine hyperplane defined by $w$ and $b$ has been translated into the task of putting the "augmented" set $\{\xi_1 \ldots, \xi_m\}$ into the positive half-space corresponding to the (non-affine) hyperplane defined by $\omega$. This task can be solved using the Kozinec algorithm, which is defined as follows.

**Kozinec algorithm.**
(1) Set $\omega = \xi_1$.
(2) If $\langle \omega, \xi_i \rangle > 0$ for all $i = 1, \ldots, m$, stop and return $\omega$. Otherwise, pick $\xi_k$ such that $\langle \omega, \xi_k \rangle < 0$, for some $1 \leq k \leq m$.
(3) Set $\omega_{\text{new}} = (1 - t_*) \cdot \omega + t_* \cdot \xi_k$, where $t_* = \text{argmin}_{t \in (0,1]} ||(1 - t) \cdot \omega + t \cdot \xi_k||$ Continue with (2).

The above algorithm terminates iff the initial collection $\{(x_i, y_i)\}_{i=1,\ldots,m}$ is linearly separable according to the labels. If the algorithm terminates, its output is a vector $\omega \in \mathbb{R}^{n+1}$. Writing

$$\omega = (w, b) \in \mathbb{R}^n \times \mathbb{R},$$

the quantities $w$ and $b$ define a hyperplane in $\mathbb{R}^n$ that separates the original collection of data points $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ according to their labels. This can be seen by "reversing" the above steps that we used to transform the collection $\{(x_i, y_i)\}_{i=1,\ldots,m}$ into $\{\xi_i\}_{i=1,\ldots,m}$. For further details about this algorithm as well as illuminating illustrations, we refer the reader to [FH03].

*Remarks.*

- The similarity of the Kozinec algorithm and the Perceptron algorithm should be apparent. Indeed, setting $\omega_{\text{new}} = \omega + \xi_t$ in step (3) recovers the Perceptron algorithm.
- In step (1) of the above algorithm, $\omega$ can, in fact, be initialized to any vector belonging to the convex hull of $\{\xi_1, \ldots, \xi_n\}$. Different initializations may lead to different separating hyperplanes.
- In step (3), the algorithm really just replaces $\omega$ with the vector belonging to the line segment passing through $\omega$ and $\xi_t$ that is closest to the origin.
- An explicit formula for the value of $t_*$ in step (3) is given by

$$t_* = \min \left\{ 1, \frac{\langle \omega, \omega - \xi_k \rangle}{\langle \omega - \xi_k, \omega - \xi_k \rangle} \right\}.$$

  This can be verified e.g. by noting that the vector $(1 - t_*) \cdot \omega + t_* \cdot \xi_k$ must be perpendicular to the vector $\omega - \xi_k$, i.e.

$$\langle (1 - t_*) \cdot \omega + t_* \cdot \xi_k, \omega - \xi_k \rangle = 0.$$

  This equation allows one to find $t_*$ as a function of $\omega$ and $\xi_k$.

## 2. The Schlesinger-Kozinec algorithm

The Schlesinger-Kozinec algorithm (henceforth "SK algorithm") is an improvement of the Kozinec algorithm that does not seek *any* separating hyperplane, but rather the *optimal* separating hyperplane. To understand what an optimal separating hyperplane is, suppose that we are given a set of labeled data points

$$\{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathbb{R}^n \times \{-1, 1\},$$

which are linearly separable according to their labels. Given any separating hyperplane $H$, its *margin* is defined to be the distance from $H$ to the closest data point, i.e.

$$m(H) = \min_{i=1,\ldots,m} ||x_i - H||.$$

A separating hyperplane is optimal if it maximizes the margin among all separating hyperplanes. In other words, a hyperplane $H_*$ is optimal if

$$H_* = \underset{H \in \mathcal{H}}{\operatorname{argmax}} \, m(H),$$

where $\mathcal{H}$ denotes the set of all hyperplanes separating the data points according to their labels. The SK algorithm does not, in fact, necessarily find the optimal separating hyperplane $H_*$, but rather a so-called $\varepsilon$-*optimal* hyperplane $H_\varepsilon$, i.e. one that satisfies

$$m(H_*) - m(H_\varepsilon) \leq \varepsilon.$$

Here, $\varepsilon > 0$ is a parameter that is chosen before letting the algorithm run. The smaller the value of $\varepsilon$, the closer the hyperplane returned by the SK algorithm will be to the theoretical optimal separating hyperplane.

As in the previous section, we will work with the "augmented" data points $\{\xi_i\}_{i=1,\dots,m}$ stemming from a collection of labeled data points $\{(x_i, y_i)\}_{i=1,\dots,m}$. As before, a separating hyperplane corresponds to a vector $\omega \in \mathbb{R}^{n+1}$ that satisfies (1.1). As mentioned, the SK algorithm does not find $\omega_*$ (the vector describing the optimal separating hyperplane) itself, but rather a vector $\omega$ that describes an $\varepsilon$-optimal hyperplane. The algorithm looks almost the same as the Kozinec algorithm, with the only difference being the stopping condition.

**Schlesinger-Kozinec algorithm.**
(1) Set $\omega = \xi_1$.
(2) If
$$||\omega|| - \min_{i=1,\dots,m} \frac{\langle \omega, \xi_i \rangle}{||\omega||} \le \varepsilon,$$
stop and return $\omega$. Otherwise, set $\xi_k = \operatorname{argmin}_{i=1,\dots,m} \langle \omega, \xi_i \rangle$.
(3) Set $\omega_{\mathrm{new}} = (1 - t_*) \cdot \omega + t_* \cdot \xi_k$, where $t_* = \operatorname{argmin}_{t \in (0,1]} ||(1-t) \cdot \omega + t \cdot \xi_k||$
Continue with (2).

Similar remarks as those to the Kozinec algorithm apply here. The idea behind the stopping condition in step (2) above is that $||\omega||$ is an upper bound for the theoretically optimal margin $m(H_*)$, while $\min_{i=1,\dots,m} \frac{\langle \omega, \xi_i \rangle}{||\omega||}$ equals the margin of $H_\omega$ (the hyperplane specified by $\omega$). Thus, provided that the stopping condition is satisfied, we have

$$m(H_*) - m(H_\omega) \le ||\omega|| - \min_{i=1,\dots,m} \frac{\langle \omega, \xi_i \rangle}{||\omega||} \le \varepsilon.$$

Hence, once the algorithm terminates, it is guaranteed that the margin of $H_\omega$ is at most $\varepsilon$ away from the theoretically optimal one.

We close by pointing out that the vector describing the optimal separating hyperplane is given by
$$\omega_* = \operatorname*{argmin}_{\xi \in \mathrm{Conv}(\Xi)} ||\xi||,$$
where $\mathrm{Conv}(\Xi)$ denotes the convex hull of $\{\xi_1, \dots, \xi_m\} \subset \mathbb{R}^{n+1}$. See [SH13, Theorem 5.3] for a proof of this statement. This fact justifies the SK algorithm above further as follows. The initial value $\omega = \xi_1$ clearly is an element of $\mathrm{Conv}(\Xi)$. Each time the value of $\omega$ is updated according to the algorithm, the new value is again an element of $\mathrm{Conv}(\Xi)$ by construction and, moreover, has strictly smaller norm than the previous value of $\omega$. Thus, it is clear that the SK algorithm converges to the optimal value $\omega_*$ (at least asymptotically, when $\varepsilon = 0$). Again, we refer the reader to [FH03] for more details and illuminating illustrations.

## References

[FH03] Vojtěch Franc and Václav Hlaváč. An iterative algorithm learning the maximal margin classifier. *Pattern Recognition*, 36(9):1985–1996, 2003. Kernel and Subspace Methods for Computer Vision.

[SH13] Michail I. Schlesinger and Václav Hlaváč. *Ten lectures on statistical and structural pattern recognition*. Computational Imaging and Vision. Springer, Dordrecht, Netherlands, 2002 edition, March 2013.