“And so ended Svejk’s Budejovice anabasis. It is certain that if Svejk had been granted liberty of movement he would have got to Budejovice on his own. However much the authorities may boast that it was they who brought Svejk to his place of duty, this is nothing but a mistake. With Svejk energy and irresistible desire to fight, the authorities action was like throwing a spanner into the works.”

– The good soldier Svejk, Jaroslav Hasek

1 Covering problems, expansion and shell sets

Consider a set $P$ of $n$ points in $\mathbb{R}^d$, that we are interested in covering by the best shape in a family of shapes $\mathcal{F}$. For example, $\mathcal{F}$ might be the set of all balls in $\mathbb{R}^d$, and we are looking for the minimum enclosing ball of $P$. A $\varepsilon$-coreset $S \subseteq P$ would guarantee that any ball that covers $S$ will cover the whole point set if we expand it by $(1 + \varepsilon)$.

However, sometimes, computing the coreset is computationally expensive, the coreset does not exist at all, or its size is prohibitively large. It is still natural to look for a small subset $S$ of the points, such that finding the optimal solution for $S$ generates (after appropriate expansion) an approximate solution to the original problem.

Definition 1.1 (Shell sets) Given a set $P$ of points (or geometric objects) in $\mathbb{R}^d$, and $\mathcal{F}$ be a family of shapes in $\mathbb{R}^d$. Let $f : \mathcal{F} \rightarrow \mathbb{R}$ be a target optimization function, and assume that there is a natural expansion operation defined over $\mathcal{F}$. Namely, given a set $r \in \mathcal{F}$, one can compute a set $(1 + \varepsilon)r$ which is the expansion of $r$ by a factor of $1 + \varepsilon$. In particular, we would require that $f((1 + \varepsilon)r) \leq (1 + \varepsilon)f(r)$.

Let $f_{\text{opt}}(P) = \min_{r \in \mathcal{F}, P \subseteq r} f(r)$ be the shape in $\mathcal{F}$ that bests fits $P$.

Furthermore, assume that $f_{\text{opt}}(\cdot)$ is a monotone function, that is for $A \subseteq B \subseteq P$ we have $f_{\text{opt}}(A) \leq f_{\text{opt}}(B)$.

A subset $\mathcal{S} \subseteq P$ is a $\varepsilon$-shell set for $P$, if $\text{SlowAlg}$ on a set $B$ that contains $\mathcal{S}$, if the range $r$ returned by $\text{SlowAlg}(\mathcal{S})$ covers $\mathcal{S}$, $(1 + \varepsilon)r$ covers $P$, and $f(r) \leq (1 + \varepsilon)f_{\text{opt}}(\mathcal{S})$. Namely, the range $(1 + \varepsilon)r$ is an $(1 + \varepsilon)$-approximation to the optimal range of $\mathcal{F}$ covering $P$.

A shell set $\mathcal{S}$ is a monotone $\varepsilon$-shell set if for any subset $B$ containing $\mathcal{S}$, if we apply $\text{SlowAlg}(B)$ and get the range $r$, then $P$ is contained inside $(1 + \varepsilon)r$ and $r$ covers $B$.

Note, that $\varepsilon$-shell sets are considerably more restricted and weaker than coresets. Of course, a $\varepsilon$-coreset is automatically a (monotone) $\varepsilon$-shell set. Note also, that if a problem has a monotone shell set, then to approximate it efficiently, all we need to do is to find some set, hopefully small, that contains the shell set.
We initialize all the points of $P$ to have weight 1, and we repeatedly do the following:

- Pick a random sample $R$ from $P$ of size $r = O((\alpha/\delta) \log(\alpha/\delta))$, where $\delta = 1/(4k_{opt})$.
  With constant probability $R$ is a $\delta$-net for $P$ by Theorem 7.1.
- Compute, using $\text{SlowAlg}(R)$ the range $r$ in $F$, such that $(1 + \varepsilon)r$ covers $R$ and realizes (maybe approximately) $f_{opt}(R)$.
- Compute the set $S$ of all the points of $P$ outside $(1 + \varepsilon)r$. If the total weight of those points exceeds $\delta w(P)$ then the random sample is bad, and return to the first step.
- If the set $S$ is empty then return $R$ as the required shell set, and $r$ as the approximation.
- Otherwise, double the weight of the points of $S$.

When done, return $r$ and the set $R$.

Figure 1: The algorithm for approximating optimal cover and computing a small shell set.

2 The Setting

Let $P$ be a set of $n$ points in $\mathbb{R}^d$, and let $\mathcal{F}$ be a family of shapes in $\mathbb{R}^d$. Furthermore, let us assume that the range space $X = (\mathbb{R}^d, \mathcal{F})$ has low VC dimension $\alpha$. Finally, assume that we want to compute the best shape in $\mathcal{F}$ that covers $P$ under a target function $f(\cdot)$. Namely, we would like to compute $f_{opt}(P) = \min_{r \in \mathcal{F}, P \subseteq B} f(r)$.

Assume that $f_{opt}(\cdot)$ is a monotone target function. Namely, if $S \subseteq T \subseteq P$ then $f_{opt}(S) \leq f_{opt}(T)$. This monotonicity property holds (almost) always for problems with small coresets.

Next, assume that we only have a slow algorithm $\text{SlowAlg}$ that can solve (maybe approximately) the given optimization problem. Namely, given a subset $S \subseteq P$, it computes a range $r \in \mathcal{F}$ such that $f(r) \leq (1 + \varepsilon)f_{opt}(S)$, and the running time of $\text{SlowAlg}$ is $T_{\text{SlowAlg}}(|S|)$.

Finally, assume that we know that a small monotone shell set of size $k_{opt}$ exists for $P$, but unfortunately we have no way of computing it explicitly (because, for example, we only have a constructive proof of the existence of such a shell set).

A natural question is how to compute this small shell set quickly, or alternatively compute an approximate shell set which is not much bigger. Clearly, once we have such a small shell set, we can approximate the optimal cover for $P$ in $\mathcal{F}$.

Example. We start with a toy example, a more interesting example is given below. Let $\mathcal{F}$ be the set of all balls in $\mathbb{R}^d$, and let $f(r)$ be the radius of the ball $r \in \mathcal{F}$. It is known that there is a $\varepsilon$-shell set for the minimum radius ball of size $O(1/\varepsilon)$ (we will prove this fact later in the course). The expansion here is the natural enlargement of a ball radius.

3 The Algorithm for Computing the Shell Set

Assume, that a kind oracle, told us that the there exist a monotone $\varepsilon$-shell set for $P$ of size $k_{opt}$, and that $\mathcal{F}$ is of VC dimension $\alpha$. The algorithm to approximate the optimal cover of $P$ and extract a small shell set is depicted in Figure 1. Note, that if we do not have a kind oracle at our possession, we can just perform a binary search for the right value of $k_{opt}$.
There are several non-trivial technicalities in implementing this algorithm. The first one is that Theorem 7.1 is for unweighted sets, but by replicating a point \( p \) of \( w_p \) times (conceptually), where \( w_p \) is the weight of \( p \), it follows that it still holds in this weighted settings.

**Random sampling from a weighted set.** Another technicality is that the weights might be quite large. To overcome this, we will store the weight of an element by storing an index \( i \), such that the weight of the element is \( 2^i \). We still need to do \( m \) independent draws from this weighted set. The easiest way to do that, is to compute the element \( e \) in of \( P \) in maximum weight, and observing that all elements of weight \( \leq w_e/n^{10} \) have weight which is so tiny, so that it can be ignored, where \( w_p \) is the weight of \( e \). Thus, normalize all the weights of by dividing them by \( 2^{\lfloor k w_e/n^{10} \rfloor} \), and remove all elements with weights smaller than 1. For a point \( p \), let \( \widehat{w}(p) \) denote its normalized weight. Clearly, all the normalized weights are integers in the range \( 1, \ldots, 2n^{10} \). Thus, we now have to pick points for a set with (small) integer weights. Place the elements in an array, and compute the prefix sum array of their weights. That is \( a_i \) have to pick points for a set with weights smaller than 1. For a point \( p \), let \( \widehat{w}(p) \) denote its normalized weight. Clearly, all the normalized weights are integers in the range \( 1, \ldots, 2n^{10} \). Thus, we now have to pick points for a set with (small) integer weights. Place the elements in an array, and compute the prefix sum array of their weights. That is \( a_i = \sum_{i=1}^{k} \widehat{w}(p_i) \), for \( i = 1, \ldots, n \). Next, pick a random number \( \gamma \) uniformly in the range \( [0, a_n] \), and using a binary search, find the \( j \), such that \( a_{j-1} \leq \gamma < a_j \). This picks the points \( p_j \) to be in the random sample. This requires \( O(n) \) preprocessing, but a single random sample can now be done in \( O(\log n) \) time. We need to perform \( r \) independent samples. Thus, this takes \( O(n + r \log n) \).

### 3.1 Correctness

**Lemma 3.1** The algorithm described above computes a \( \varepsilon \)-shell set for \( P \) of size \( O(r) = O(k_{opt} \log (k_{opt} \alpha)) \). The algorithm performs \( O(4k_{opt} \log n) \) iterations.

**Proof:** We only need to prove that the algorithm terminates in the claimed number of iterations. Observe, that with constant probability (say \( \geq 0.9 \)), the sample \( R_i \), in the \( i \) th iteration, is an \( \delta \)-net for \( P_{i-1} \) (the weighted version of \( P \) in the end of the \( (i-1) \)th iteration), in relation to the ranges of \( \mathcal{F} \). Observe, that this also implies that \( R_i \) is a \( \delta \)-net for the complement family \( \overline{\mathcal{F}} = \{ \mathbb{R}^d \setminus r \mid r \in \mathcal{F} \} \), with constant probability (since \( (\mathbb{R}^d, \mathcal{F}) \) and \( (\mathbb{R}^d, \overline{\mathcal{F}}) \) have the same VC dimension).

If \( R_i \) is such a \( \delta \)-net, then we know that range \( r \) we compute completely covers the set \( R_i \), and as such, for any range \( r' \in \overline{\mathcal{F}} \) that avoids \( R_i \), we have \( w(r') \leq \delta w(P_{i-1}) \). In particular, this implies that \( \omega(S_i) \leq \delta w(P_{i-1}) \). If not, than \( R_i \) is not a \( \delta \)-net, and we resample. The probability for that is \( \leq 0.1 \). As such, we expect to repeat this \( O(1) \) times in each iteration, till we have \( w(S_i) \leq \delta w(P_{i-1}) \).

Thus, in each iteration, the algorithm doubles the weight of at most a \( \delta \)-fraction of the total point set. Thus \( w(P_i) \leq (1 + \delta)w(P_{i-1}) = n(1 + \delta)\).

On the other hand, consider the smallest shell \( \mathcal{S} \) of \( P \), which is of size \( k_{opt} \). If all the elements of \( \mathcal{S} \) are in \( R_i \), then the algorithm would have terminated, since \( \mathcal{S} \) is a monotone shell set. Thus, if we continue to the next iteration, it must be that \( |\mathcal{S} \cap S_i| \geq 1 \). In particular, we are doubling the weight of at least one element of the shell set. We conclude that the weight of \( P_i \) in the \( i \) th iteration, is at least

\[
k_{opt} 2^{i/k_{opt}},
\]

since in every iteration at least one element of \( \mathcal{S} \) gets its weight redoubled. Thus, we have

\[
\exp\left( \frac{i}{2k_{opt}} \right) \leq 2^{i/k_{opt}} \leq k_{opt} 2^{i/k_{opt}} \leq (1 + \delta)^{i} n \leq n \cdot \exp(\delta i) = n \cdot \exp\left( \frac{i}{4k_{opt}} \right).
\]
Namely, \(\exp\left(\frac{i}{k_{\text{opt}}}\right) \leq n\). Implying that \(i \leq 4k_{\text{opt}} \ln n\). Namely, after \(4k_{\text{opt}} \ln n\) iterations the algorithm terminates, and thus returns the required shell set and approximation.

**Theorem 3.2** Under the settings of Section 2, one can compute a monotone \(\varepsilon\)-shell set for \(P\) of size \(O(k_{\text{opt}} \alpha \log(k_{\text{opt}} \alpha))\). The running time of the resulting algorithm is \(O((n + T(k_{\text{opt}} \alpha \log(k_{\text{opt}} \alpha)))k_{\text{opt}} \ln n)\), with high probability, for \(k_{\text{opt}} \leq n/\log^3 n\). Furthermore, one can compute an \(\varepsilon\)-approximation to \(f_{\text{opt}}(P)\) in the same time bounds.

**Proof:** The algorithm is described above. The bounds on the running time follows from the bounds on the number of iterations from Lemma 3.1. The only problem we need to address, is that the resampling would repeatedly fail, and the algorithm would spend exuberant amount of time on resampling. However, the probability of failure in sampling is \(\leq 0.1\). Furthermore, we need at most \(4k_{\text{opt}} \log n\) good samples before the algorithm succeeds. It is now straightforward to show using Chernoff inequality, that with high probability, we will perform at most \(8k_{\text{opt}} \log n\) samplings before achieving the required number of good samples.

### 3.2 Set Covering in Geometric Settings

Interestingly, the algorithm we discussed, can be used to get an improved approximation algorithm for the set covering problem in geometric settings. We remind the reader that set covering is the following problem.

**Problem:** \textbf{Set Covering}

\[
\begin{array}{l}
\text{Instance: } (S, \mathcal{F}) \\
S \text{ - a set of } n \text{ elements} \\
\mathcal{F} \text{ - a family of subsets of } S, \text{ s.t. } \bigcup_{X \in \mathcal{F}} X = S. \\
\text{Question: What is the set } X \subseteq \mathcal{F} \text{ such that } X \text{ contains as few sets as possible, and } X \text{ covers } S? \\
\end{array}
\]

The natural algorithm for this problem is the greedy algorithm that repeatedly pick the set in the family \(\mathcal{F}\) that covers the largest number of uncovered elements in \(S\). It is not hard to show that this provides a \(O(|S|)\) approximation. In fact, it is known that set covering can be better approximated unless \(P = NP\).

Assume, however, that we know that the VC dimension of the set system \((S, \mathcal{F})\) has VC dimension \(\alpha\). In fact, we need a stronger fact, that the dual family

\[
S = \left(\mathcal{F}, \left\{U(s, \mathcal{F}) \mid s \in S\right\}\right),
\]

is of low VC dimension \(\alpha\), where \(U(s, \mathcal{F}) = \left\{X \mid s \in X, X \in \mathcal{F}\right\}\).

It turns out that the algorithm of Figure 1 also works in this setting. Indeed, we set the weight of the sets to 1, we pick a random sample of sets. If they cover the universe \(S\), we are done. Otherwise, there must be a point \(p\) which is not covered. Arguing as above, we know that the random sample is a \(\delta\)-net of (the weighted) \(S\), and as such all the sets containing \(p\) have total weight \(\leq \delta(S)\). As such, double the weight of all the sets covering \(p\), and repeat. Arguing as above, one can show that the algorithm terminates after \(O(k_{\text{opt}} \log m)\) iterations, where \(m\) is the number of sets, where \(k_{\text{opt}}\) is the number of sets in the optimal cover of \(S\). Furthermore, the size of the cover generated is \(O(k_{\text{opt}} \alpha \log(k_{\text{opt}} \alpha))\).
Theorem 3.3 Let \((S, \mathcal{F})\) be a range space, such that the dual range space \(S\) has VC dimension \(\alpha\). Then, one can compute a set covering for \(S\) using sets of \(\mathcal{F}\) using \(O(k_{\text{opt}} \alpha \log(k_{\text{opt}} \alpha))\) sets. This requires \(O(k_{\text{opt}} \log n)\) iterations, and takes polynomial time.

Note, that we did not provide in Theorem 3.3 exact running time bounds. Usually in geometric settings, one can get improved running time using the underlying geometry. Interestingly, the property that the dual system has low VC dimension “buys” one a lot, as it implies that one can do \(O(\log k_{\text{opt}})\) approximation, instead of \(O(\log n)\) in the general case.

4 Application - Covering Points by Cylinders

5 Clustering and Coresets

We would like to cover a set \(P\) of \(n\) points in \(\mathbb{R}^d\) by \(k\) balls, such that the radius of maximum radius ball is minimized. This is known as the \(k\)-center clustering problem (or just \(k\)-center). The price function, in this case, \(r_d(k_{\text{opt}}(P))\) is the radius of the maximum radius ball in the optimal solution.

Definition 5.1 Let \(P\) be a point set in \(\mathbb{R}^d\), \(1/2 > \varepsilon > 0\) a parameter.

For a cluster \(c\), let \(c(\delta)\) denote the cluster resulting form expanding \(c\) by \(\delta\). Thus, if \(c\) is a ball of radius \(r\), then \(c(\delta)\) is a ball of radius \(r + \delta\). For a set \(C\) of clusters, let

\[ C(\delta) = \left\{ c(\delta) \mid c \in C \right\}, \]

be the additive expansion operator; that is, \(C(\delta)\) is a set of clusters resulting form expanding each cluster of \(C\) by \(\delta\).

Similarly,

\[ (1 + \varepsilon)C = \left\{ (1 + \varepsilon) c \mid c \in C \right\}, \]

is the multiplicative expansion operator, where \((1 + \varepsilon)c\) is the cluster resulting from expanding \(c\) by a factor of \((1 + \varepsilon)\). Namely, if \(C\) is a set of balls, then \((1 + \varepsilon)C\) is a set of balls, where a ball \(c \in C\), corresponds to a ball radius \((1 + \varepsilon)\) radius \((c)\) in \((1 + \varepsilon)C\).

A set \(S \subseteq P\) is an (additive) \(\varepsilon\)-coreset of \(P\), in relation to a price function radius, if for any clustering \(C\) of \(S\), we have that \(P\) is covered by \(C(\varepsilon \text{ radius}(C))\), where \(\text{radius}(C) = \max_{c \in C} \text{radius}(c)\). Namely, we expand every cluster in the clustering by an \(\varepsilon\)-fraction of the size of the largest cluster in the clustering. Thus, if \(C\) is a set of \(k\) balls, then \(C(\varepsilon f(C))\) is just the set of balls resulting from expanding each ball by \(\varepsilon r\), where \(r\) is the radius of the largest ball.

A set \(S \subseteq P\) is a multiplicative \(\varepsilon\)-coreset of \(P\), if for any clustering \(C\) of \(S\), we have that \(P\) is covered by \((1 + \varepsilon)C\).

Lemma 5.2 Let \(P\) be a set of \(n\) points in \(\mathbb{R}^d\), and \(\varepsilon > 0\) a parameter. There exists an additive \(\varepsilon\)-coreset for the \(k\)-center problem, and this coreset has \(O(k/\varepsilon^d)\) points.

Proof: Let \(C\) denote the optimal clustering of \(P\). Cover each ball of \(C\) by a grid of side length \(\varepsilon r_{\text{opt}}/d\), where \(r_{\text{opt}}\) is the radius of the optimal \(k\)-center clustering of \(P\). From each such grid cell, pick one points of \(P\). Clearly, the resulting point set \(S\) is of size \(O(k/\varepsilon^d)\) and it is an additive coreset of \(P\).

The following is a minor extension of an argument used in [APV02],

\[ \square \]
Lemma 5.3 Let $P$ be a set of $n$ points in $\mathbb{R}^d$, and $\varepsilon > 0$ a parameter. There exists a multiplicative $\varepsilon$-coreset for the $k$-center problem, and this coreset has $O(k!/\varepsilon^{dk})$ points.

Proof: For $k = 1$, the additive coreset of $P$ is also a multiplicative coreset, and it is of size $O(1/\varepsilon^d)$.

As in the proof of Lemma 5.2 we cover the point set by a grid of radius $\varepsilon r_{\text{opt}}/(5d)$, let $SQ$ the set of cells (i.e., cubes) of this grid which contains points of $P$. Clearly, $|SQ| = O(k/\varepsilon^d)$.

Let $S$ be the additive $\varepsilon$-coreset of $P$. Let $C$ be any $k$-center clustering of $S$, and let $\Delta$ be any cell of $SQ$.

If $\Delta$ intersects all the $k$ balls of $C$, then one of them must be of radius at least $(1 - \varepsilon/2) \text{rd}(P, k)$. Let $c$ be this ball. Clearly, when we expand $c$ by a factor of $(1 + \varepsilon)$ it would completely cover $\Delta$, and as such it would also cover all the points of $\Delta \cap P$.

Thus, we can assume that $\Delta$ intersects at most $k - 1$ balls of $C$. As such, we can inductively compute an $\varepsilon$-multiplicative coreset of $P \cap \Delta$, for $k - 1$ balls. Let $Q_\Delta$ be this set, and let $Q = S \cup \bigcup_{Q \in SQ} Q_\Delta$.

Note that $|Q| = T(k, \varepsilon) = O(k/\varepsilon^d)T(k-1, \varepsilon) + O(k/\varepsilon^d) = O(k!/\varepsilon^{dk})$. The set $Q$ is the required multiplicative coreset by the above argumentation. ■

6 Union of Cylinders

Let assume we want to cover $P$ by $k$ cylinders of minimum maximum radius (i.e., fit the points to $k$ lines). Formally, consider $\mathcal{S}$ to be the set of all cylinders in $\mathbb{R}^d$, and let $\mathcal{F} = \left\{ c_1 \cup c_2 \cup \ldots \cup c_k \mid c_1, \ldots, c_k \in \mathcal{F} \right\}$ be the set, which its members are union of $k$ cylinders. For $C \in \mathcal{F}$, let $f(C) = \max_{c \in C} \text{radius}(c)$.

Let $f_{\text{opt}}(P) = \min_{C \in \mathcal{F}, P \subseteq C} f(C)$.

One can compute the optimal cover of $P$ by $k$ cylinders in $O(n^{2d-1}k+1)$ time, see below for details. Furthermore, $(\mathbb{R}^d, \mathcal{F})$ has VC dimension $\alpha = O(dk \log(dk))$. Finally, one can show that this set of cylinders has $\varepsilon$-coreset of small size $\cdots$. Thus, we would like to compute a small $\varepsilon$-coreset, and compute an approximation quickly.

6.0.1 Covering by Cylinders - A Slow Algorithm

It is easy to verify (but tedios) that the VC dimension of $(\mathbb{R}^d, \mathcal{F}_k)$ is bounded by $\alpha = O(dk \log(dk))$. Furthermore, it has a small coreset (see Section 2). Furthermore, given a set $P$ of $n$ points, and consider its minimum radius enclosing cylinder $c$. The cylinder $c$ has (at most) $2d - 2$ points of $P$ on its boundary which if we compute their minimum enclosing cylinder, it is $c$. Note, that $c$ might contain even more points on its boundary, we are only claiming that there is a defining subset of size $2d - 1$. This is one of those “easy to see” but very tedious to verify facts. Let us quicly outline an intuitive explanation (but not a proof!) of this fact. Consider the set of lines $\mathcal{L}^d$ of lines in $\mathbb{R}^d$. Every member of $\ell \in \mathcal{L}^d$ can be parameterized by the closest point $p$ on $\ell$ to the origin, and consider the hyperplane that passes through $p$ and is orthogonal to $op$, where $o$ is the origin. The line $\ell$ now can be parameterized by its orientation in $h$. This requires specifying a point on the $d - 2$ dimensional unit hypersphere $S^{d-2}$. Thus, we can specify $\ell$ using $2d - 2$ real numbers.

Next, define for each point $p_i \in P$, its distance $g_i(\ell)$ from $\ell \in \mathcal{L}^d$. This is a messy but a nice algebraic function defined over $2d - 2$ variables. In particular, $g_i(\ell)$ induces a surface in $2d - 1$ dimensions (i.e., $\cup_{\ell}(p_i, g_i(\ell))$). Consider the arrangement $\mathcal{A}$ of those surfaces. Clearly, the minimum volume cylinder lies on a feature of this arrangement, thus to specify the minimum radius cylinder, we just need to specify the feature (i.e., vertex, edge, etc) of the arrangement that contains this point. However, every feature in an arrangement of well behaved surfaces in $2d - 1$ dimensions,
can be specified by $2d - 1$ surfaces. (This is intuitively clear but requires a proof - an intersection of $k$ surfaces, is going to be $d - k$ dimensional, where $d$ is the dimension of the surfaces. If we add a surface to the intersection and it does not reduce the dimension of the intersection, we can reject it, and take the next surface passing through the feature we care about.) Every such surface corresponds to a original point.

Thus, if we want to specify a minimum multiplicative radius cylinder induced by a subset of $P$, all we need to specify are $2d - 1$ points.

To specify $k$ such cylinders, we need to specify $M = (2d - 1)k$ points. This immediately implies that we can find the optimal cover of $P$ by $k$ cylinders in $O(n^{(2d - 1)k + 1})$ time, but just enumerating all such subsets of $M$ points, and computing for each subset its optimal cover (note, that the $O$ notation hides a constant that depends on $k$ and $d$).

Thus, we have a slow algorithm that can compute the optimal cover of $P$ by $k$ cylinders.

### 6.1 Existence of a Small Coreset

Since the coreset in this case is either multiplicative or additive, it is first important to define the expansion operation carefully. In particular, if $C$ is a set of $k$ cylinders, the $(1 + \varepsilon)$-expanded set of cylinders would be $C(\varepsilon \cdot \text{radius}(C))$, where radius$(C)$ is the radius of the largest cylinder in $C$.

Let $P$ be the given set of $n$ points in $\mathbb{R}^d$. Let $C_{\text{opt}}$ be the optimal cover of $P$ by $k$ cylinders. For each cylinder of $C_{\text{opt}}$ place $O(1/\varepsilon^{d-1})$ parallel lines inside it, so that for any point inside the union of the cylinders, there is a line in this family in distance $\leq (\varepsilon/10)r_{\text{opt}}$ from it. Let $\mathcal{L}$ denote this set of lines.

Let $P'$ be the point set resulting from snapping each point of $P$ to its closest point on $\mathcal{L}$. We claim that $P'$ is a $(\varepsilon/10)$-coreset for $P$, as can be easily verified. Indeed, if a set $C$ of $k$ cylinders cover $P'$, then the largest cylinder must be of radius $r \geq (1 - \varepsilon/10)\text{rd}(P, k)$, where $\text{rd}(P, k)$ is the radius of the optimal coverage of $P$ by $k$ cylinders. Otherwise, $\text{rd}(P, k) \leq r + (\varepsilon/10)\text{rd}(P, k) < \text{rd}(P, k)$.

The set $P'$ lies on $O(1/\varepsilon^{d-1})$-lines. Let $\ell \in \mathcal{L}$ be such a line, and consider the point-set $P'_\ell$. Assume for a second that there was a multiplicative $\varepsilon$-coreset $\mathcal{T}_\ell$ on this line. If $\mathcal{T}_\ell$ is covered by $k$ cylinders, each cylinder intersect $\ell$ along an interval. Expanding each such cylinder by a factor of $1 + \varepsilon$ is equivalent to expanding each such intersecting interval by a factor of $1 + \varepsilon$. However, by Lemma 5.3, we know that such a multiplicative $(\varepsilon/10)$-coreset exists, of size $O(1/\varepsilon^k)$. Thus, let $\mathcal{T}_\ell$ be the multiplicative $(\varepsilon/10)$-coreset for $P'_\ell$ for $k$ intervals on the line. Let $\mathcal{T} = \bigcup_{\ell \in \mathcal{L}} \mathcal{T}_\ell$. We claim that $\mathcal{T}$ is a (additive) $(\varepsilon/10)$-coreset for $P'$. This is trivial, since being a multiplicative coreset for each line implies that the union is a multiplicative coreset, and a $\delta$-multiplicative coreset is also a $\delta$-additive coreset. Thus, $\mathcal{T}$ is a $((1 + \varepsilon/10)^2 - 1)$-coreset for $P$. The only problem is that the points of $\mathcal{T}$ are not points in $P$. However, they correspond to points in $P$ which are in distance at most $(\varepsilon/10)\text{rd}(P, k)$ from them. Let $S$ be the corresponding set of points of $P$. It is now easy to verify that $S$ is indeed a $\varepsilon$-coreset for $P$, since $((1 + \varepsilon/10)^2 - 1) + \varepsilon/10 \leq \varepsilon$. We summarize:

**Theorem 6.1** Let $P$ be a set of $n$ points in $\mathbb{R}^d$. There exists a (additive) $\varepsilon$-coreset for $P$ of size $O(k/\varepsilon^{d-1+k})$ for covering the points by $k$-cylinders of minimum radius.

### 7 From previous lectures

**Theorem 7.1** Let $(X, R)$ be a range space of VC-dimension $d$, let $A$ be a finite subset of $X$ and suppose $0 < \varepsilon, \delta < 1$. Let $N$ be a set obtained by $m$ random independent draws from $A$, where

$$m \geq \max \left( \frac{4}{\varepsilon \log \frac{2}{\delta}}, \frac{8d}{\varepsilon \log \frac{8d}{\varepsilon}} \right).$$
Then $N$ is an $\varepsilon$-net for $A$ with probability at least $1 - \delta$.

8 Bibliographical notes

Section 3.2 is due to Clarkson [Cla93]. This technique was used to approximate terrains [?], and covering polytopes [Cla93].

The observation that this argument can be used to speedup approximation algorithms is due to Agarwal et al. [APV02]. The discussion of shell sets is implicit in the work of Bădoiu et al. [BHI02].

References

