# Lecture 23: Colorings with Low Discrepancy

Bansal's algorithm (2010). This computes the desired coloring through a sequence of semicolorings, where a semicoloring is an arbitrary mapping $c : V \to [-1,1]$.

The algorithm starts with the semicoloring $x_0 = 0$. Then it produces a sequence

$x_0, x_1, x_2, \ldots, x_\ell \in [-1,1]^n$ of semicolorings.

What $\ell$ should be will be determined later.

The t-th step. $x_t$ is obtained from $x_{t-1}$ as follows.

Let $A_t = \{ j \in V : (x_{t-1})_j \neq \pm 1 \}$ is the set of coordinates that are still active in the t-th step.

We will use semidefinite programming to compute a coloring of $A_t$ by unit vectors (rather than $\pm 1$).

More explicitly, we compute unit vectors $u_{t,j}$ where $j \in A_t$ so that

$$\Big\| \sum_{j \in A_t \cap S_i} u_{t,j} \Big\|^2 \leq D^2$$

for all $i$, with $D \geq 0$ as small as possible.

For convenience, we also set $u_{t,j} = 0$ for $j \notin A_t$.

Next, we generate a random vector $r_t$ from the n-dimensional standard normal distribution, i.e., the coordinates of $r_t$ are independent $N(0,1)$ random variables.

Then we set $(\Delta_t)_j = \sigma \cdot r_t^T u_{t,j}$ for $j = 1, \ldots, n$.

Here $\sigma$ is a small parameter.

We will update $x_{t-1}$ to $x_t$ as follows:

- A tentative value of $x_t$ is $\tilde{x}_t = x_{t-1} + \Delta_t$.
- But we need to truncate each coordinate to the interval $[-1, 1]$. So we update $\tilde{x}_t$ as:

$$(x_t)_j = \begin{cases} +1 & \text{if } (\tilde{x}_t)_j \geq 1 \\ -1 & \text{if } (\tilde{x}_t)_j \leq -1 \\ (\tilde{x}_t)_j & \text{otherwise,} \end{cases}$$

for $j = 1, \dots, n$.

The length $l$ of the sequence $x_0, x_1, \dots, x_l$ of semicolorings will be set to $\frac{C_1 \cdot \log n}{\sigma^2}$ for a suitable constant $C_1$.

As we will see in due time $\sigma = \frac{1}{C_0 n \sqrt{\log n}}$ for another suitable constant $C_0$.

This concludes the description of Bansal's algorithm.

---

Roughly speaking, why does the algorithm work? The algorithm can be regarded as a random walk in the cube $[-1, 1]^n$. The projection of the random walk to a given coordinate axis behaves like a 1-dimensional random walk with increments having the normal distribution $\sigma \cdot N(0, 1) = N(0, \sigma^2)$. Such a walk will typically cross the boundary of the interval $[-1, 1]$ in about $1/\sigma^2$ steps. After $\frac{\log n}{\sigma^2}$ steps, it is very likely to have crossed the boundary. This means that $x_l$ is very likely to be a $\pm 1$ vector.

Regarding the <u>imbalance</u> of any $S_i$, it starts out at 0 under the semicoloring $x_0$ and in the $t$-th step, it is changed by $\sum_{j \in S_i} \sigma \cdot r_t^T u_{t,j} = \sigma r_t^T v_{t,i}$ where $v_{t,i} = \sum_{j \in S_i} u_{t,j}$. But the $u_{t,j}$ were selected with the goal of making $\| v_{t,i} \|$ small. So the imbalance of each $S_i$ grows slowly during the algorithm.

We will now make these arguments formally.

Claim 1. The algorithm produces a coloring with probability $\geq 1 - 1/n$.

Proof. Let $x_0, x_1, \ldots, x_\ell$ be the sequence of semicolorings generated by the algorithm. Let $j \in \{1, \ldots, n\}$ be a fixed index. We will call the sequence $(x_0)_j, (x_1)_j, \ldots, (x_\ell)_j$ the j-th coordinate random walk.

We will say that the j-th coordinate random walk terminates if $(x_\ell)_j \in \{\pm 1\}$. To prove Claim 1, it is enough to show that, for every $j$, the probability that the j-th coordinate random walk does not terminate is $\leq 1/n^2$. Then w.p. $\geq 1 - \frac{1}{n}$, all of the coordinate random walks terminate (by the union bound).

To simplify notation, let us fix $j$ and write $X_t = (x_t)_j - (x_{t-1})_j$, for $t = 1, \ldots, \ell$.

We want to claim that $X_1, X_2, \ldots, X_\ell$ are independent random variables but this is not true.

This is because once $(x_t)_j \in \{\pm 1\}$, this value stays fixed. Let us define $t_0 \leq l$ to be the last step of the $j$-th coordinate walk for which $(x_{t_0})_j \in (-1,1)$.

For $t \leq t_0$, we have $X_t = (\Delta_t)_j = \sigma \cdot r_t^T u_{t,j}$ for some unit vector $u_{t,j}$. Recall that $r_t$ is $n$-dimensional Gaussian, independent of $u_{t,j}$. Recall that $r_t$ is spherically symmetric — so $r_t^T u_{t,j}$ has the 1-dimensional standard normal distribution $N(0,1)$ and $\boxed{\sigma \cdot r_t^T u_{t,j}}$ $\to$ this is $X_t$ has the 1-dimensional normal distribution $N(0,\sigma^2)$.

We want to claim that $X_1, \ldots, X_{t_0}$ are independent random variables. Again, one has to be careful since $t_0$ itself is not independent of $X_1, X_2, \ldots,$ so even the formulation of such a claim may not be clear.

So we will formulate our claim of independence in the following way. Let $Z'_1, \ldots, Z'_l$ be a new sequence of independent random variables, each with the $N(0,\sigma^2)$ distribution. We define another sequence of random variables $Z_1, Z_2, \ldots, Z_l$ as follows:

$$Z_t = \begin{cases} X_t & \text{for } t \leq t_0 \\ Z'_t & \text{for } t > t_0. \end{cases}$$

We claim that $Z_1, Z_2, \ldots, Z_l$ are independent.

The justification is that if we fix the values of $r_1, \ldots, r_t$ in the algorithm and also the values of the auxiliary variables $Z'_1, \ldots, Z'_{t-1}$ arbitrarily, the values of $Z_1, \ldots, Z_{t-1}$ are determined uniquely, while $Z_t$ has the $N(0,\sigma^2)$ distribution.

So if the $j$-th coordinate walk does not terminate, then all the partial sums $Z_1 + Z_2 + \ldots + Z_t$ belong to $(-1, 1)$, $t = 1, 2, \ldots, l$.

Lemma. Let $Z_1, \ldots, Z_l$ be independent random variables, each with the $N(0, \sigma^2)$ distribution. Then the probability that all of the partial sums $\sum_{i=1}^{t} Z_i$, $t = 1, 2, \ldots, l$ are in $(-1, 1)$ is at most $\boxed{e^{-k_1 \lfloor \sigma^2 l \rfloor}}$, for a suitable constant $k_1$.

So Claim 1 follows from the above lemma (since $l = \frac{C_1 \cdot \log n}{\sigma^2}$). Note that the above lemma implies that the probability the $j$-th coordinate random walk does not terminate $\leq e^{-k_1 \lfloor C_1 \log n \rfloor} \leq \frac{1}{n^2}$ for $C_1 = \frac{3}{k_1}$ and $n$ sufficiently large.

Proof of the lemma. Let $k = \sigma^{-2}$. Let us assume for convenience that $k \in \mathbb{Z}$. Let us partition the sequence $Z_1, Z_2, \ldots$ into contiguous blocks of length $k$ and let $s_j$ be the sum of the $j$-th block. Formally, $s_j = \sum_{i=(j-1)k+1}^{jk} Z_i$. The number of blocks is $\lfloor \frac{l}{k} \rfloor$.

Fact. If $X, Y$ are independent $N(0,1)$ random variables and $a, b \in \mathbb{R}$, then $aX + bY$ has the 1-dimensional normal distribution $N(0, a^2 + b^2)$.

So each $s_j$ has the standard normal distribution $N(0, k\sigma^2) = N(0,1)$. Thus $\Pr[|s_j| \geq 2] \geq c_0$ for a suitable $c_0$ (in fact, $c_0 \approx 0.0455$).

If $\sum_{i=1}^{t} Z_i \in (-1, 1)$ for all $t = 1, 2, \ldots, l$, then necessarily $|s_j| < 2$ for all $j$. The $s_j$ are independent, thus the probability that $|s_j| < 2$ for all $j$ is $\leq (1 - c_0)^{\lfloor l/k \rfloor} = e^{-k_1 \lfloor \sigma^2 l \rfloor}$. Thus the lemma is proved. ▨

---

We will next show the following claim.

Claim 2. The discrepancy of the (semi) coloring $x_l$ returned by the algo. is $O(\mathcal{H} . \log(mn))$, where $\mathcal{H}$ is the hereditary vector discrepancy of $\mathbb{F}$.

(Note that if $x_l$ is not a coloring (this happens w.p. $\leq 1/n$), we restart the algorithm from scratch. Thus $x_l$ as returned by the algo. is always a coloring.)

---

The vector discrepancy of $\mathbb{F}$ (denoted by vecdisc($\mathbb{F}$)) is $\min D \geq 0$ s.t. $\|\sum_{j \in S_i} u_j\|^2 \leq D^2$ for $i = 1, 2, \ldots, m$ and $\|u_j\|^2 = 1$ for $j = 1, \ldots, n$. The hereditary vector discrepancy of $\mathbb{F}$ is $\max_{A \subseteq V}$ vecdisc$(\mathbb{F}|_A)$.

---

In order to prove Claim 2, we will prove

$$\Pr[\text{disc}(\mathbb{F}, x_l) > D_{max}] \leq 1/n,$$

where $D_{max} = O(\mathcal{H} . \log(mn))$.

Let us fix a set $S_i \in \mathbb{F}$ and let $D_i = \sum_{j \in S_i} (x_l)_j$ be its imbalance in $x_l$. We will show $\Pr[|D_i| > D_{max}] \leq \frac{1}{mn}$ for every $i$. Then Claim 2 will follow from the union bound.