On the Worst-Case Complexity of the k-Means Method

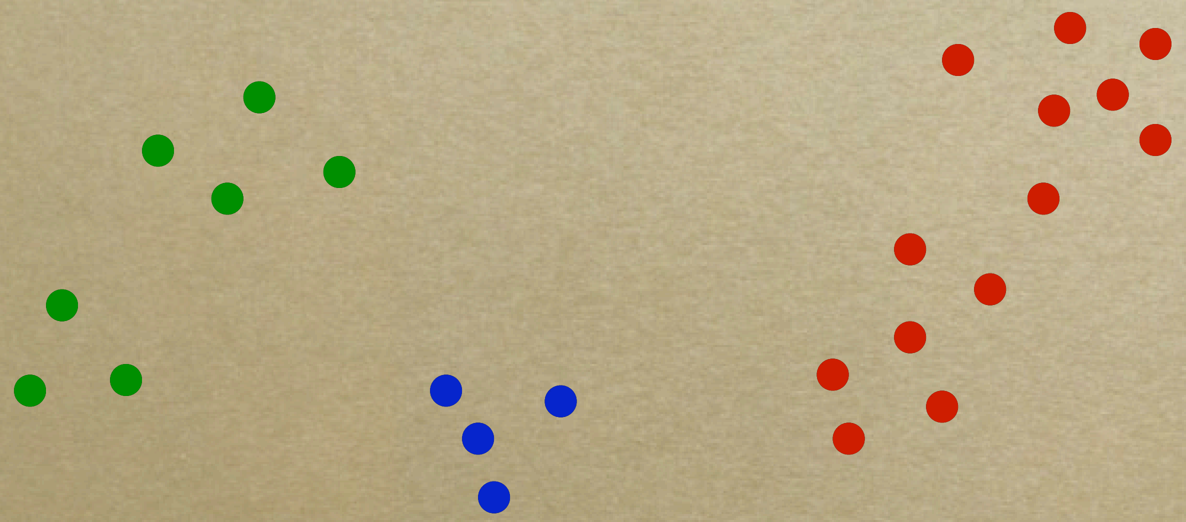
Sergei Vassilvitskii

David Arthur

(Stanford University)

Clustering

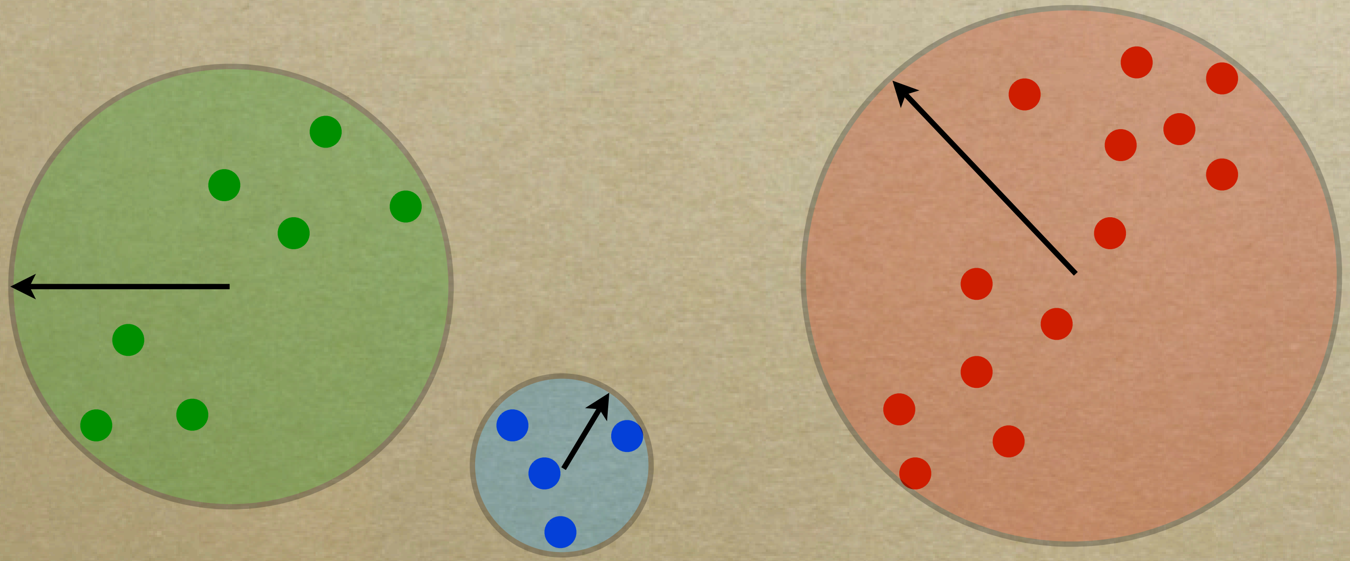
Given n points in $\mathcal{R}^d$ split them into k similar groups.



Clustering Objectives

Let $C(x)$ be the closest cluster center to x.

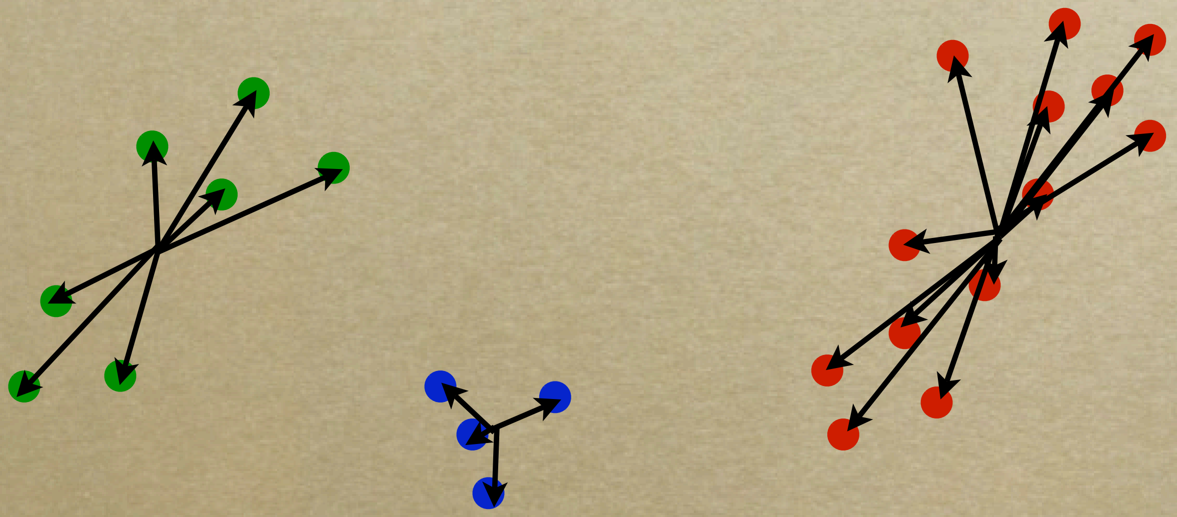
k-Center: $\min \max_{x \in X} \|x - C(x)\|$



Clustering Objectives

Let $C(x)$ be the closest cluster center to x.

k-Median: $\min \sum_{x \in X} \|x - C(x)\|$

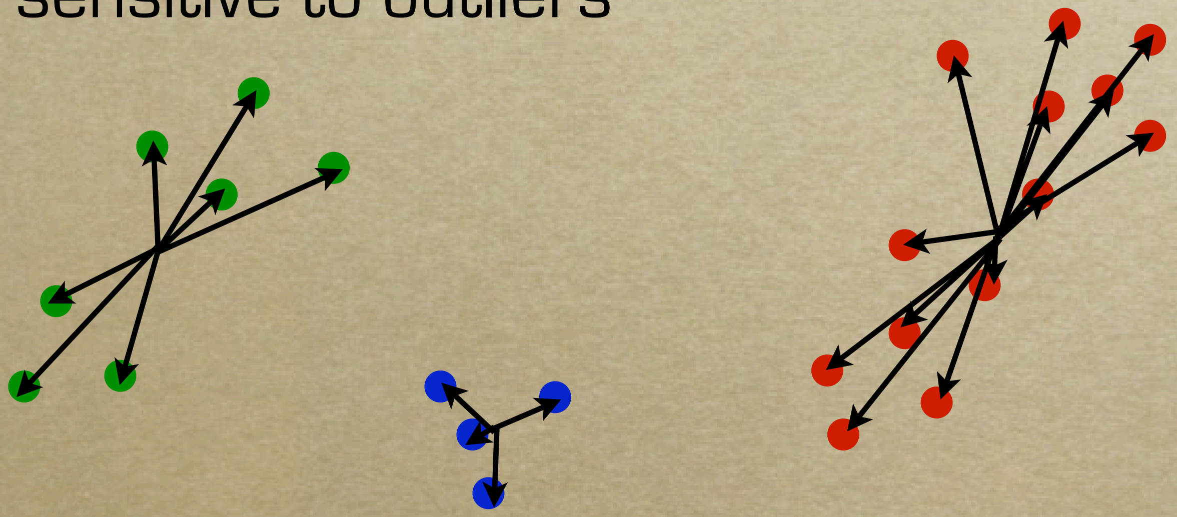


Clustering Objectives

Let $C(x)$ be the closest cluster center to x.

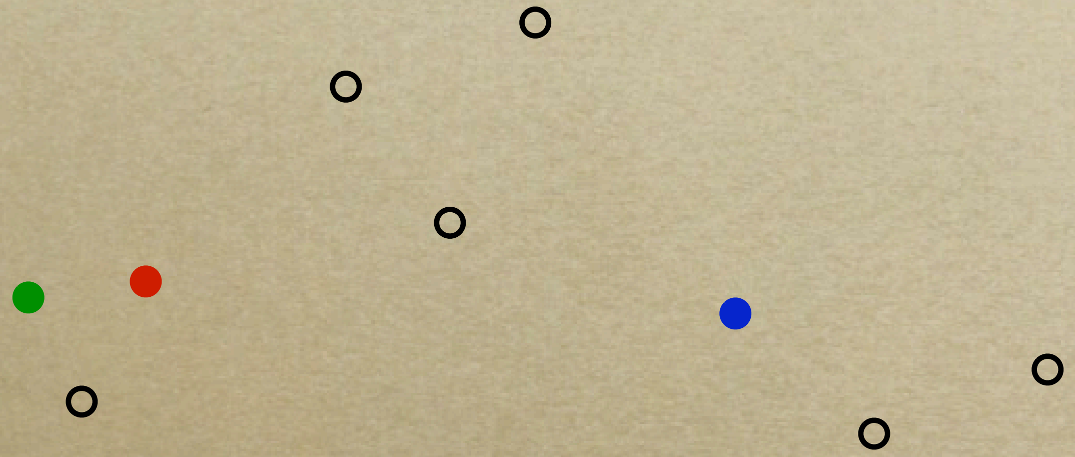
k-Median Squared: $\min \sum_{x \in X} \|x - C(x)\|^2$

Much more sensitive to outliers



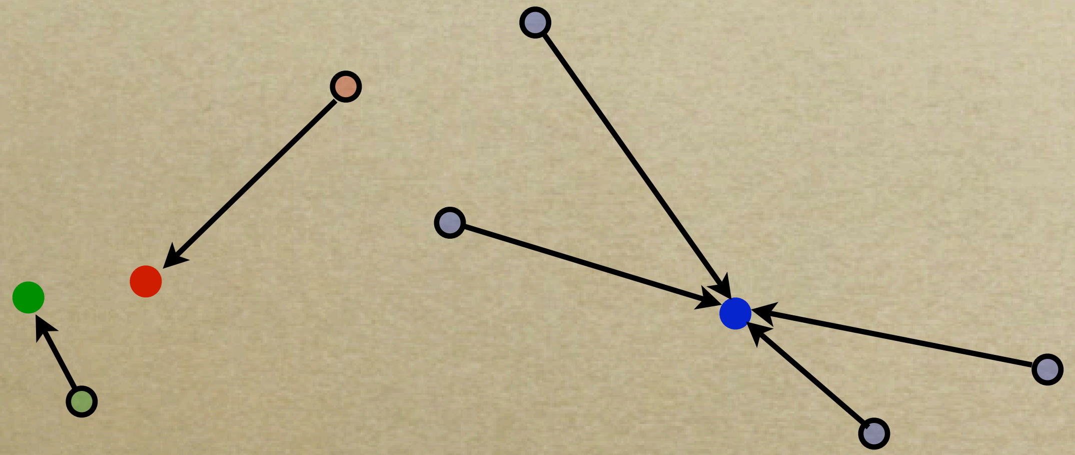
Lloyd's Method: K-means

Initialize with random clusters



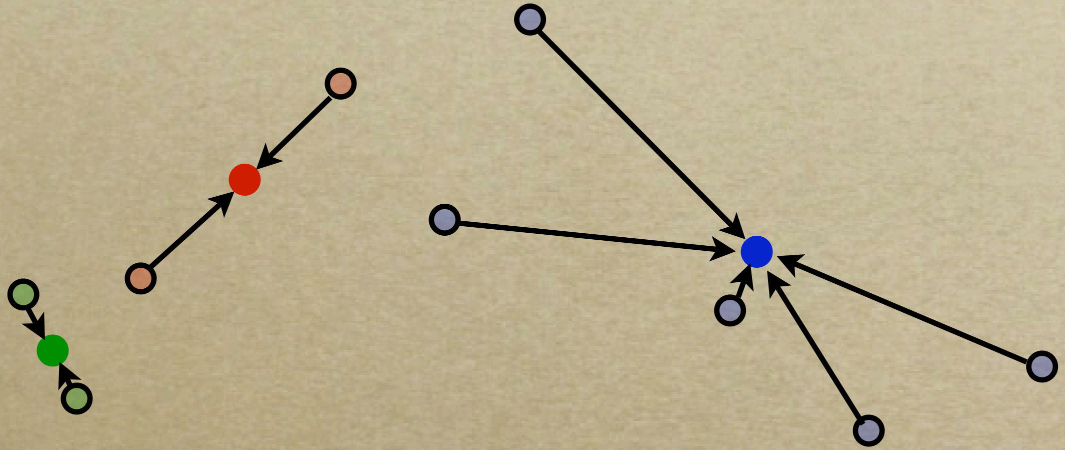
Lloyd's Method: K-means

Assign each point to nearest center



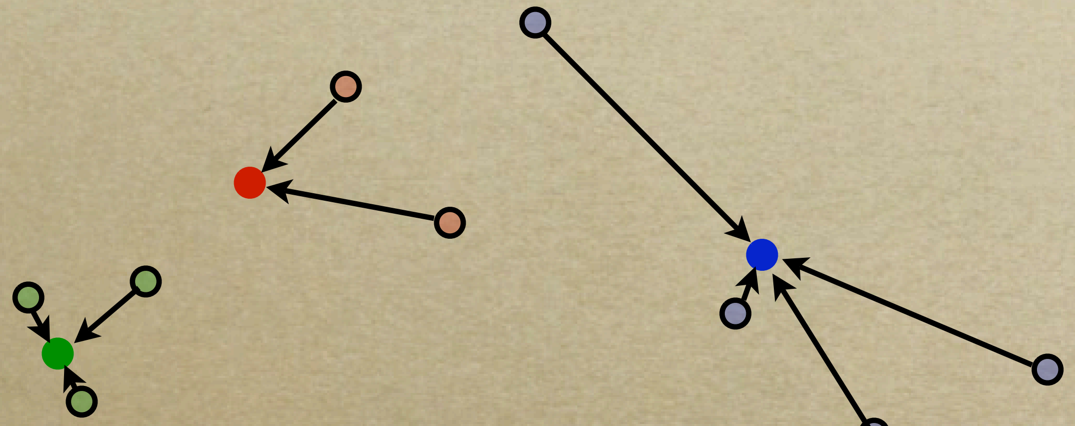
Lloyd's Method: K-means

Recompute optimum centers (means)



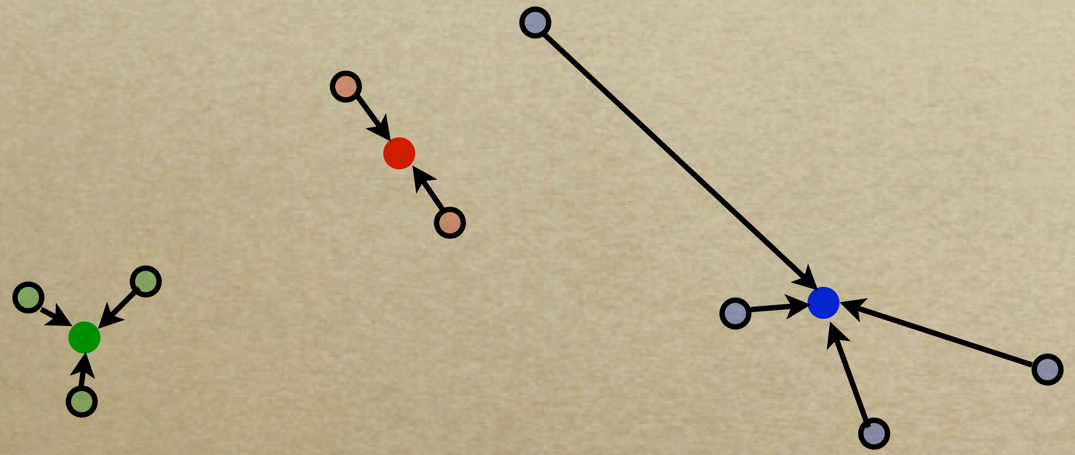
Lloyd's Method: K-means

Repeat: Assign points to nearest center



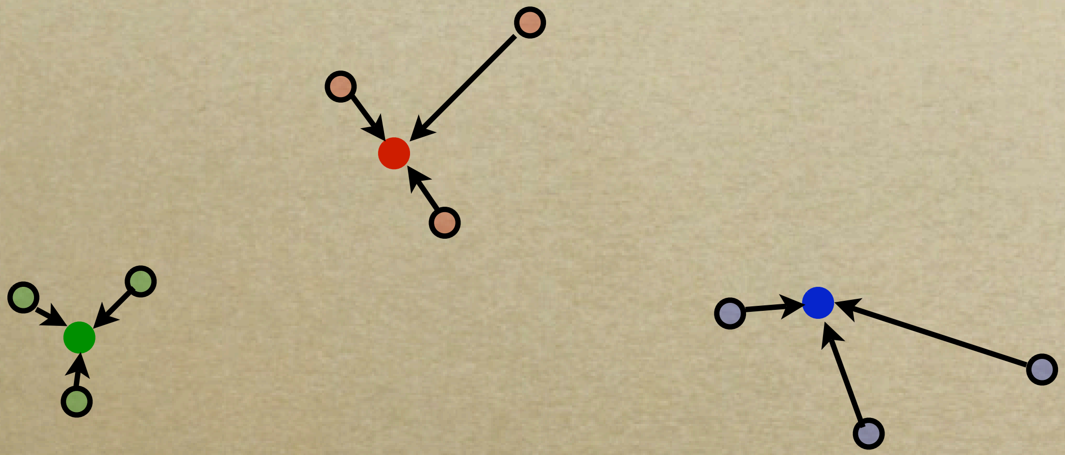
Lloyd's Method: K-means

Repeat: Recompute centers



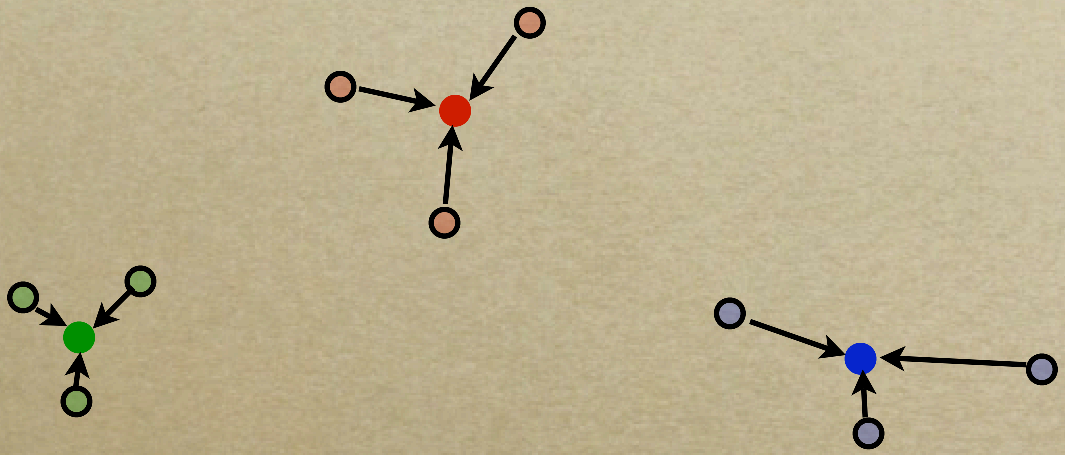
Lloyd's Method: K-means

Repeat...



Lloyd's Method: K-means

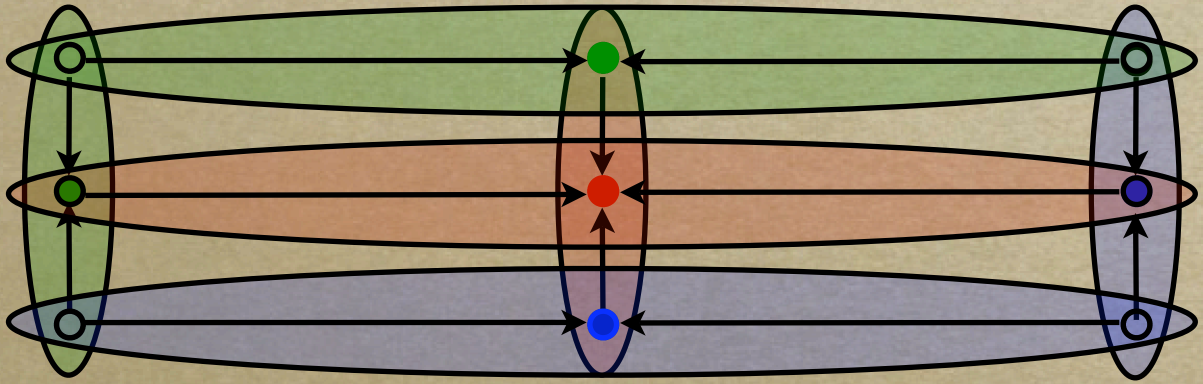
Repeat...Until clustering does not change



Analysis

How good is this algorithm?

Finds a local optimum



That’s arbitrarily worse than optimal solution

Analysis

How fast is this algorithm?

In practice: VERY fast:

e.g. Digit Recognition dataset

with $n = 60,000, d = 700$

Converges after 60 iterations

In theory: Stay Tuned.

Previous Work

Lower Bounds

- $\Omega(n)$ on the line
- $\Omega(n^2)$ in the plane

Upper Bounds:

- $\mathcal{O}(n\Delta^2)$ on the line, spread $\Delta = \dfrac{\max_{x,y} \|x-y\|}{\min_{x,y} \|x-y\|}$
- Exponential bounds:

$$\mathcal{O}(k^n), \mathcal{O}(n^{kd})$$

Our Results

Lower Bound: $2^{\Omega(\sqrt{n})}$

Smoothed Upper Bounds:

$$\mathcal{O}\left(n^{2+2/d}\left(\frac{D}{\sigma}\right)^2 2^{2n/d}\right)$$

$$\mathcal{O}\left(n^{k+2/d}\left(\frac{D}{\sigma}\right)^2\right)$$

σ is the smoothness factor

D Diameter of the pointset

Rest of the Talk

- Lower Bound Sketch
- Upper Bound Sketch
- Open Problems

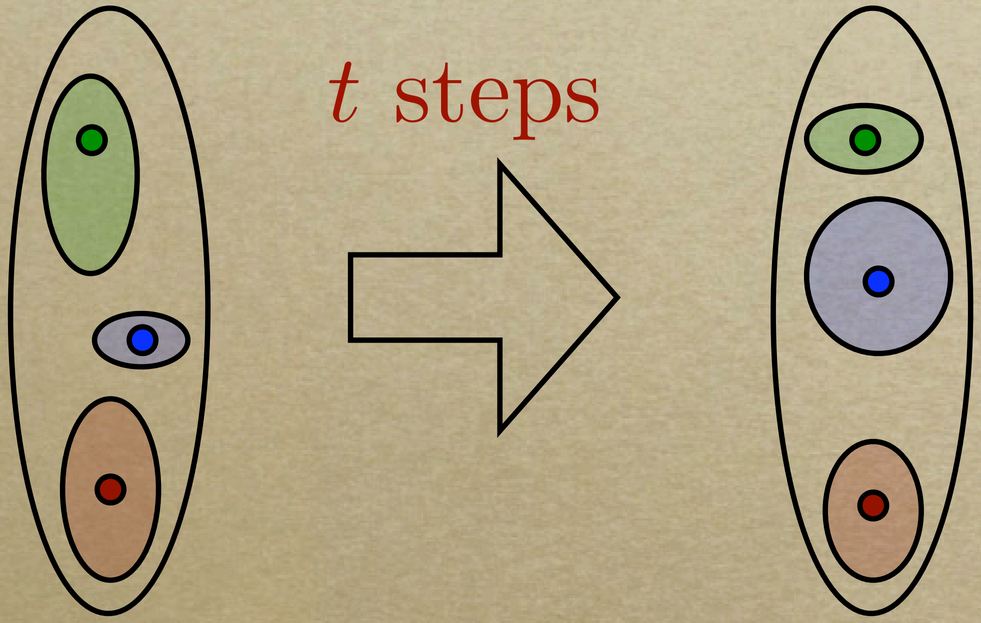
Lower Bound

General Idea:

- Make a “Reset Widget”:
  - If k-Means takes time t on X, create a new point set X', s.t. k-Means takes time $2t$ to terminate.

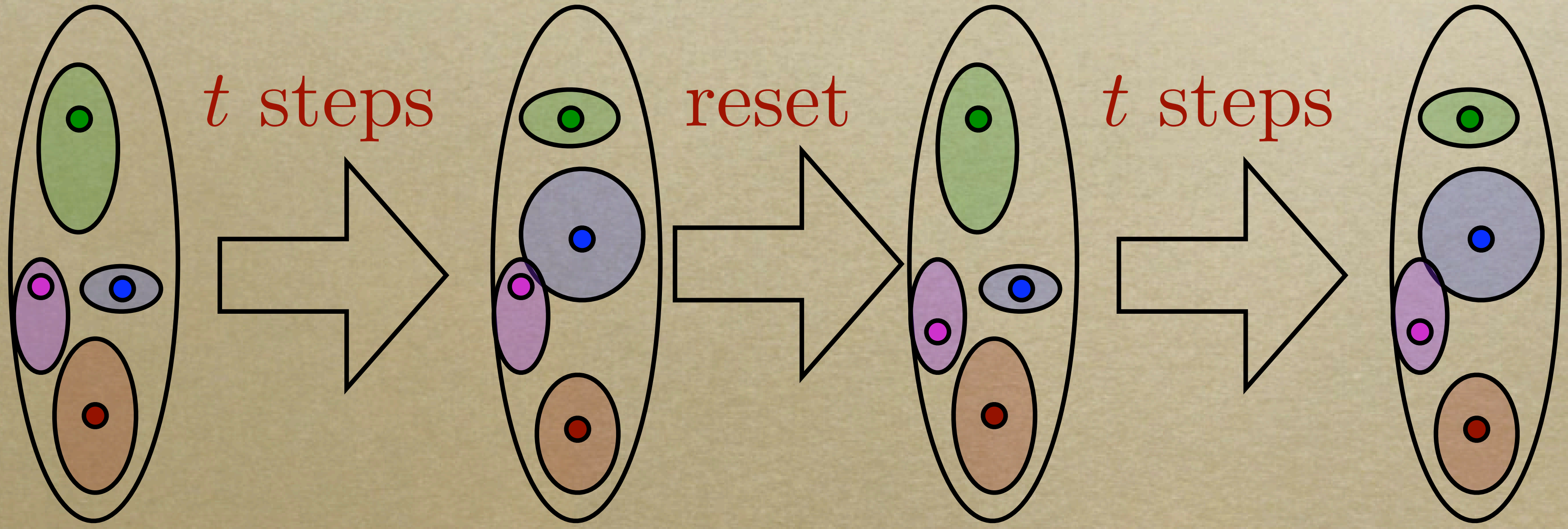
Lower Bound: Sketch

Initial Clustering:



Lower Bound: Sketch

With Widget



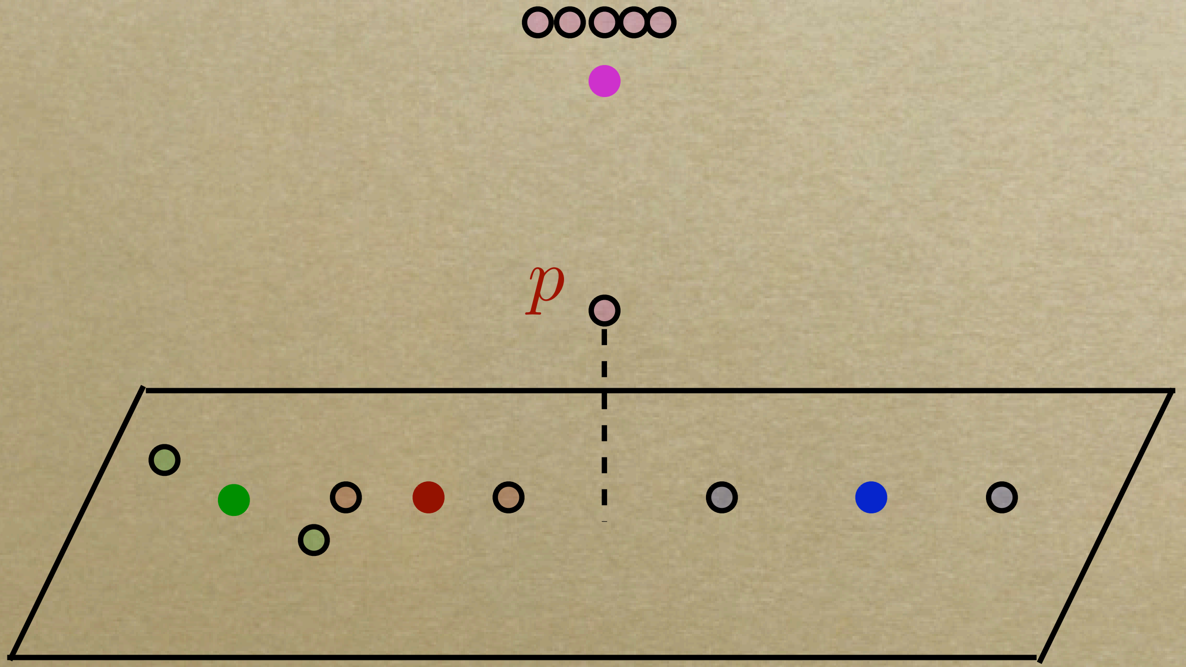
Lower Bound Details

Three Main Ideas:

- Signaling - Recognizing when to start flipping the switch
- Resetting - Setting the cluster centers back to original position
- Odds & Ends - Clean-up to make the process recursive

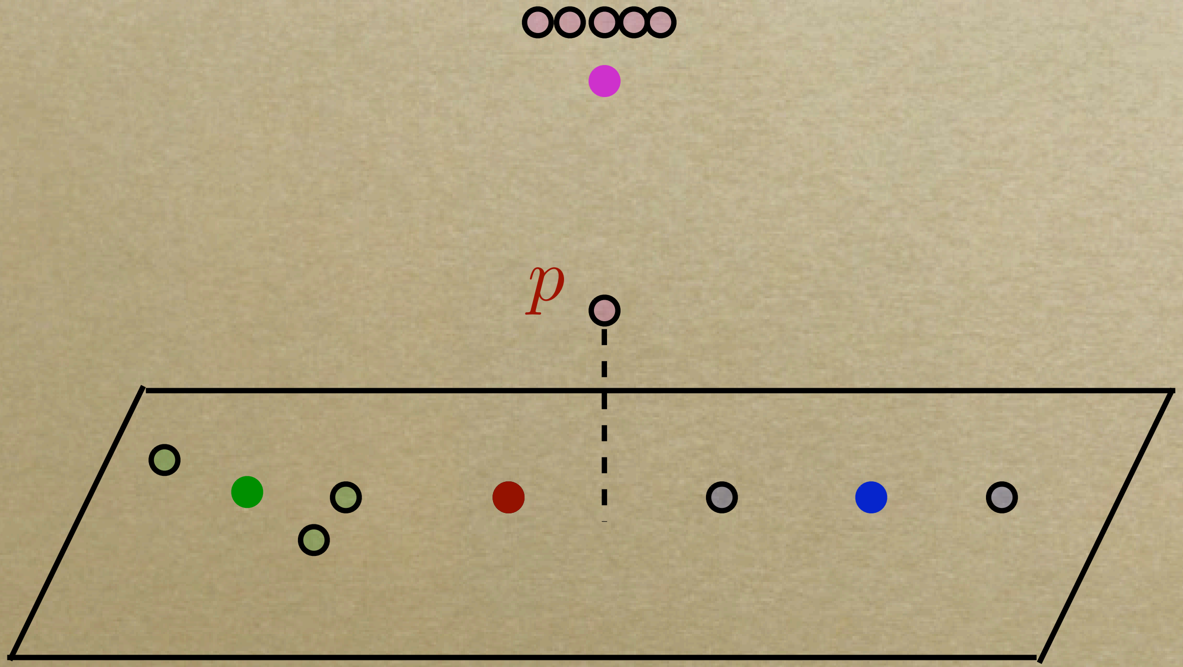
Signaling

Suppose that when k-Means terminates, there is one cluster center that has never appeared before. We use this as a signal to start the reset sequence.



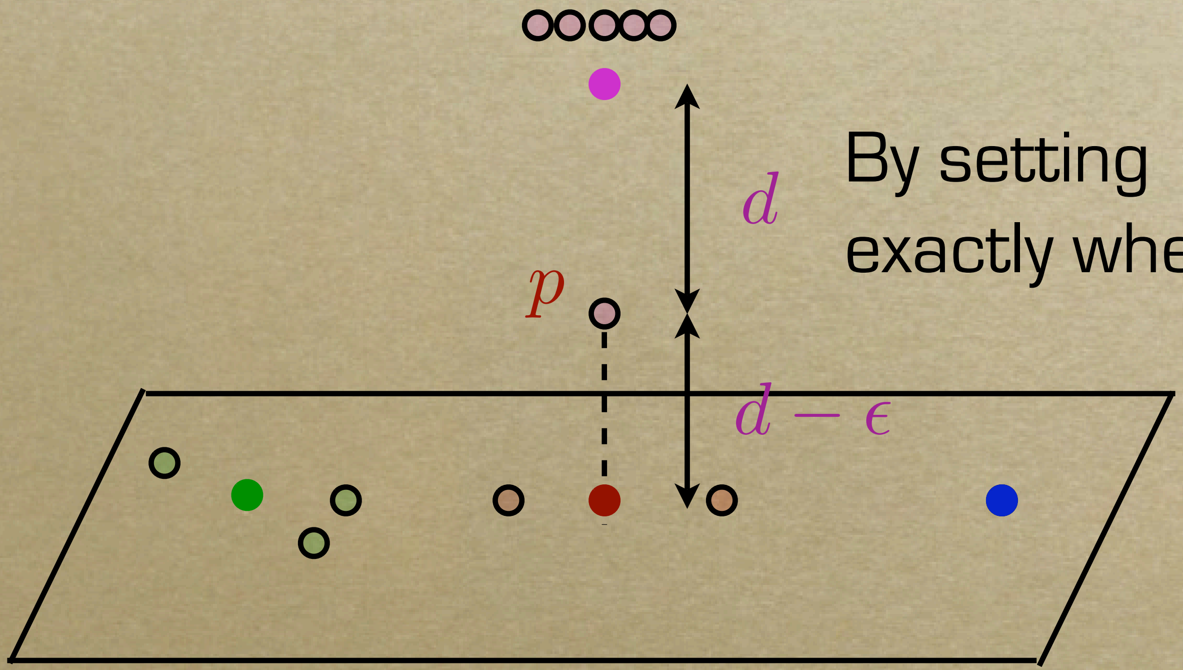
Signaling

Suppose that when k-Means terminates, there is one cluster center that has never appeared before. We use this as a signal to start the reset sequence.



Signaling

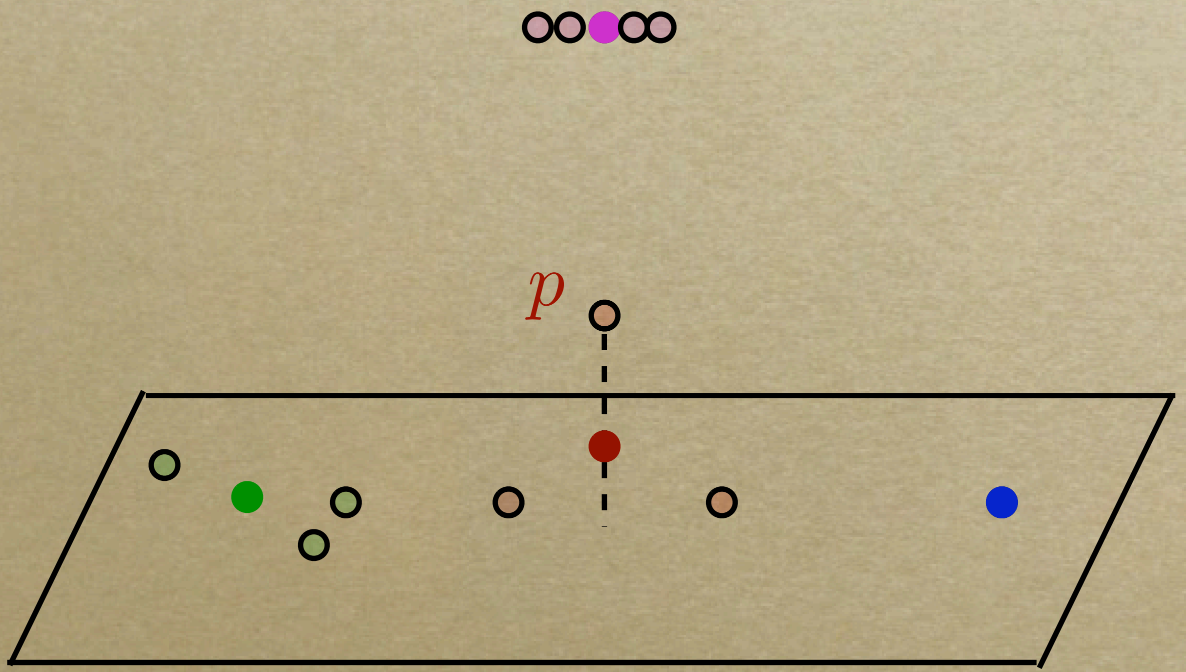
Suppose that when k-Means terminates, there is one cluster center that has never appeared before. We use this as a signal to start the reset sequence.



By setting ϵ we can control exactly when p will switch.

Signaling

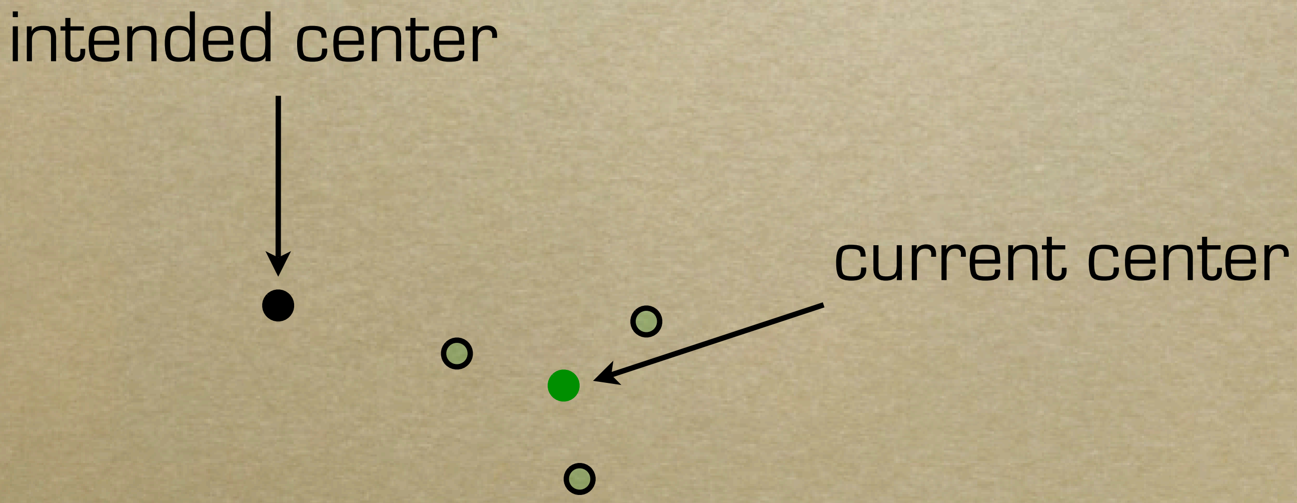
Suppose that when k-Means terminates, there is one cluster center that has never appeared before. We use this as a signal to start the reset sequence.



Resetting

k properly placed points can reset the positions of the k current centers

Easy to compute locations of the reset points, so that new cluster centers are placed correctly:



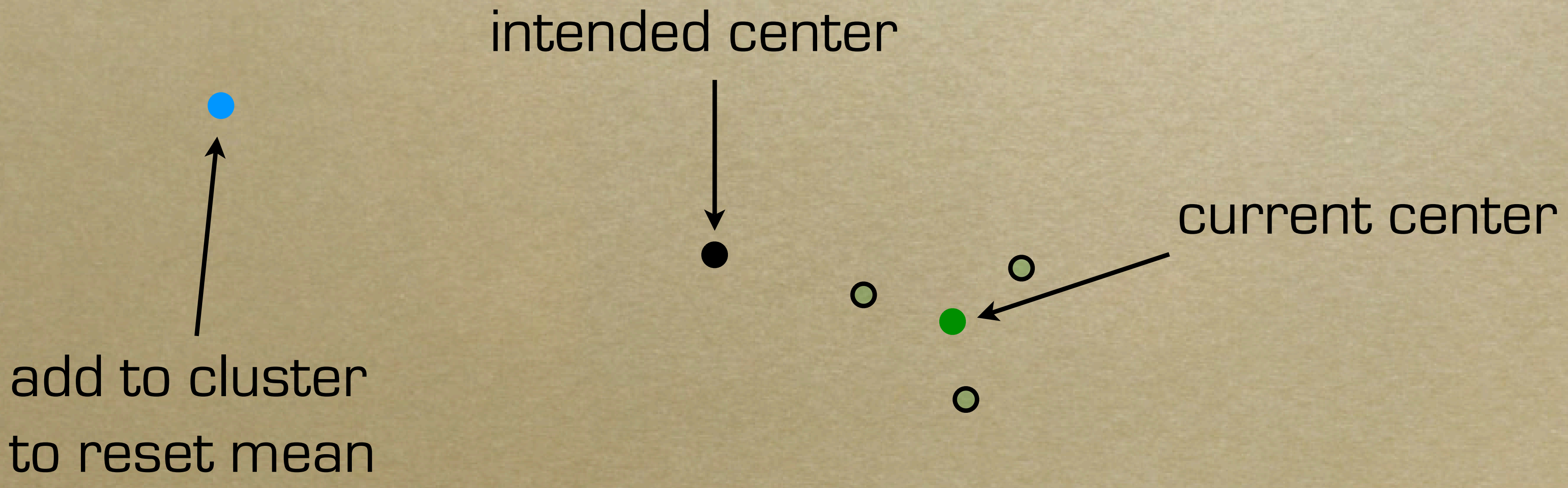
intended center

current center

Resetting

k properly placed points can reset the positions of the k current centers

Easy to compute locations of the reset points, so that new cluster centers are placed correctly:



intended center

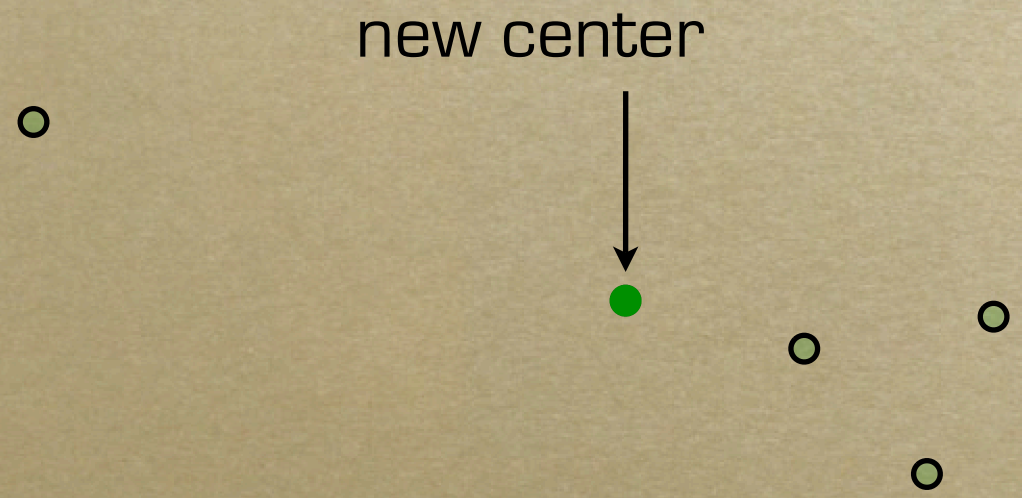
current center

add to cluster to reset mean

Resetting

k properly placed points can reset the positions of the k current centers

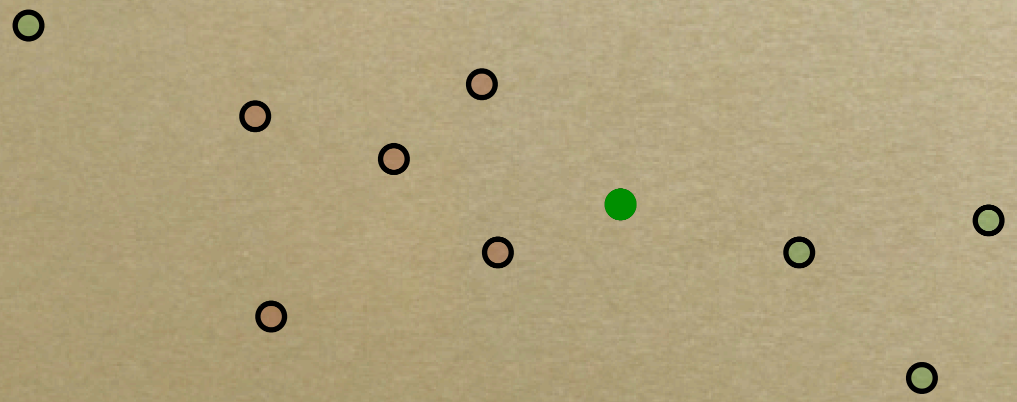
Easy to compute locations of the reset points, so that new cluster centers are placed correctly:



Resetting

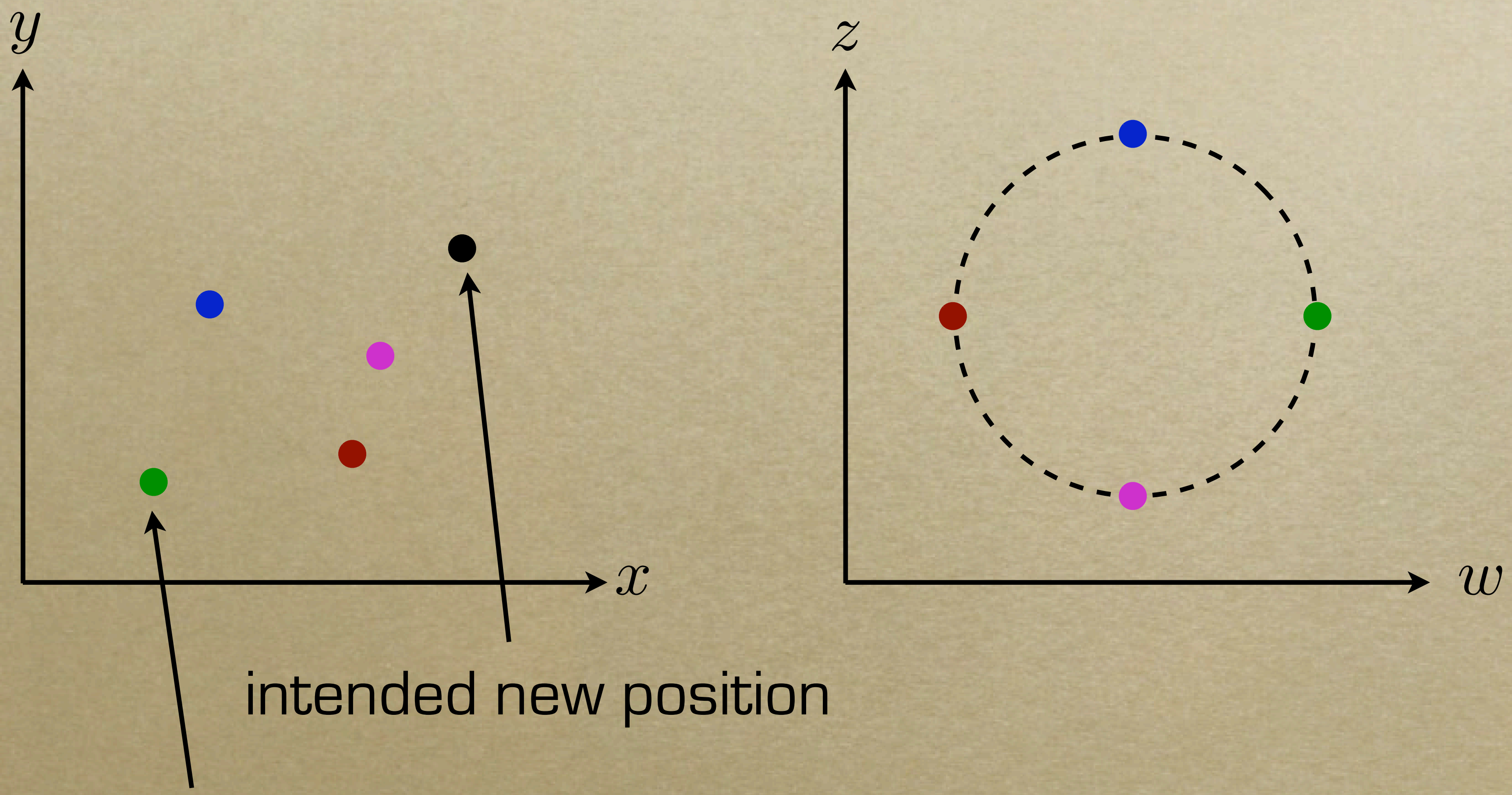
Easy to compute locations of the reset points, so that new cluster centers are placed correctly:

But must avoid accidentally grabbing other points



Resetting

Solution: Add two new dimensions

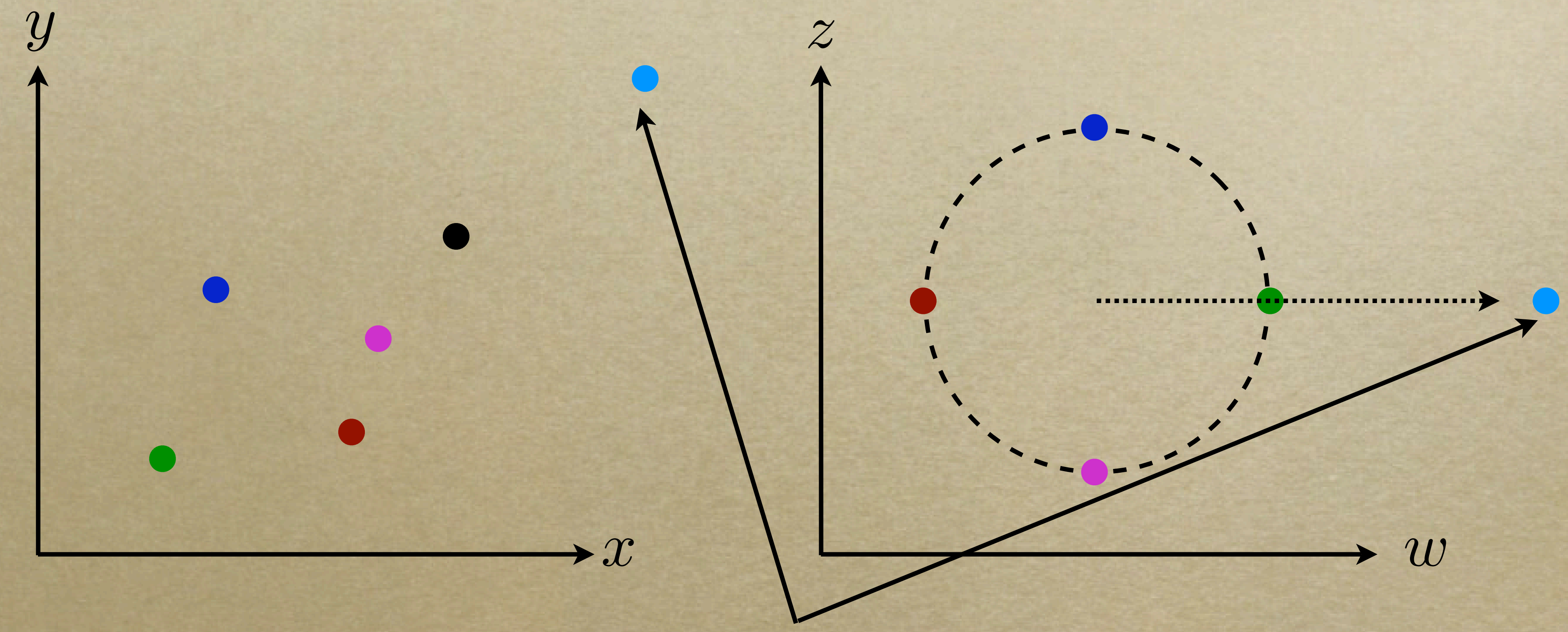


intended new position

center to reset

Resetting

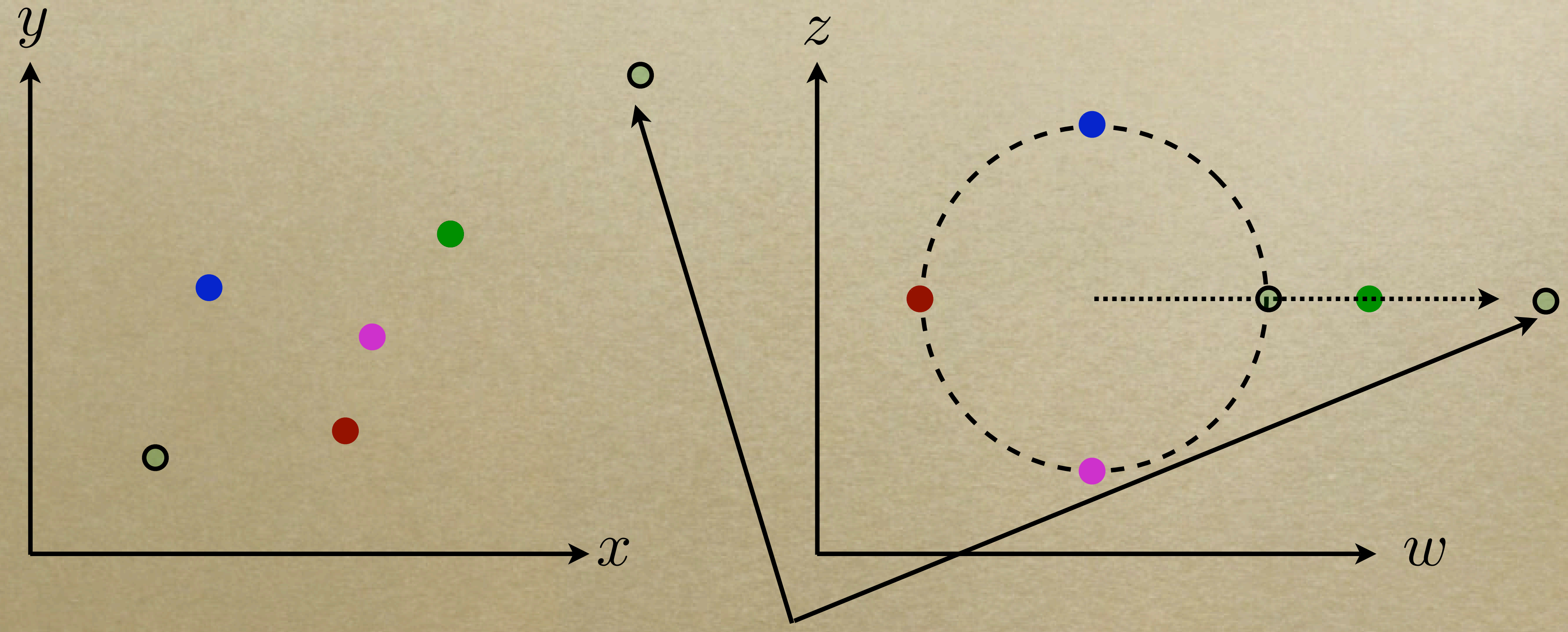
Solution: Add two new dimensions



new point added

Resetting

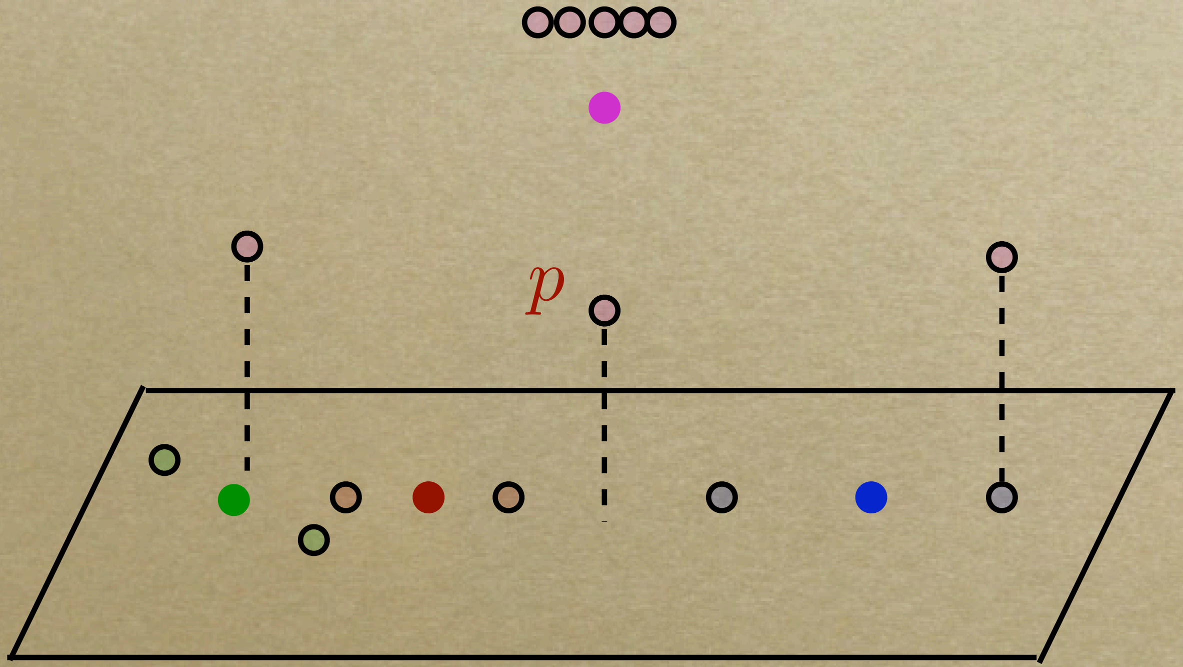
Solution: Add two new dimensions



new point added

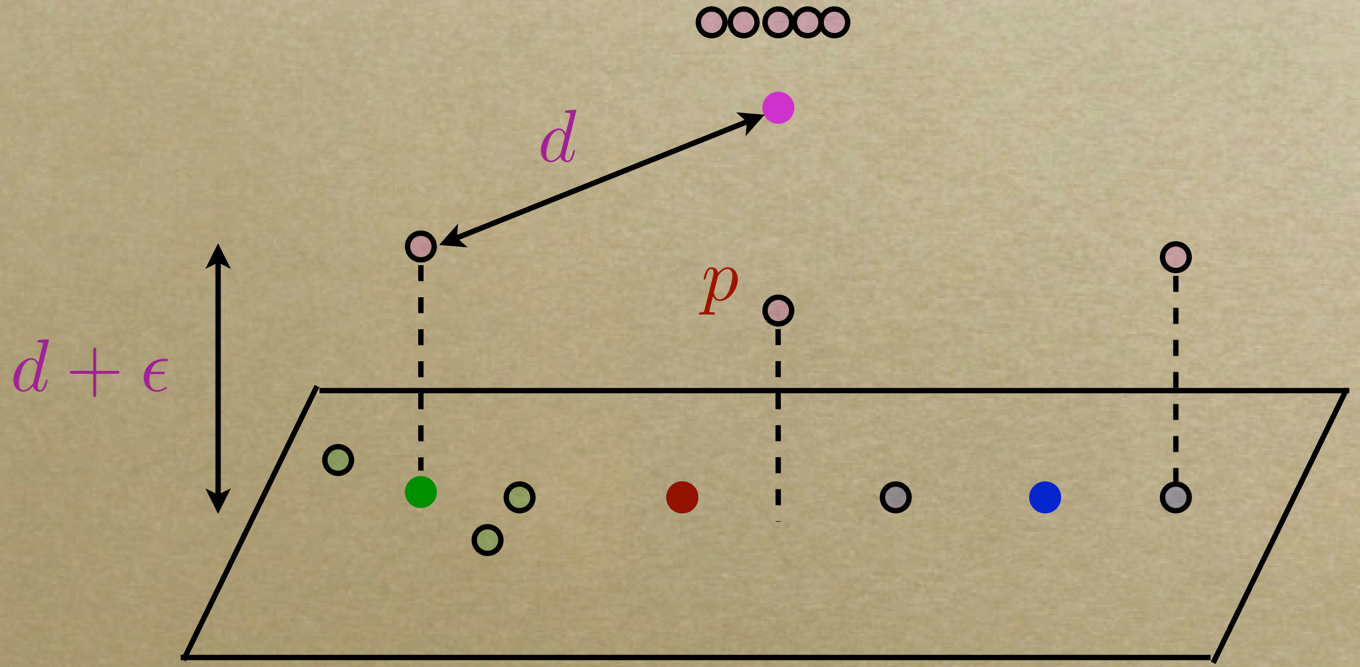
Multi-Signaling

So far have shown how to signal and reset a single cluster. Can use one signal to induce a signal from all clusters.



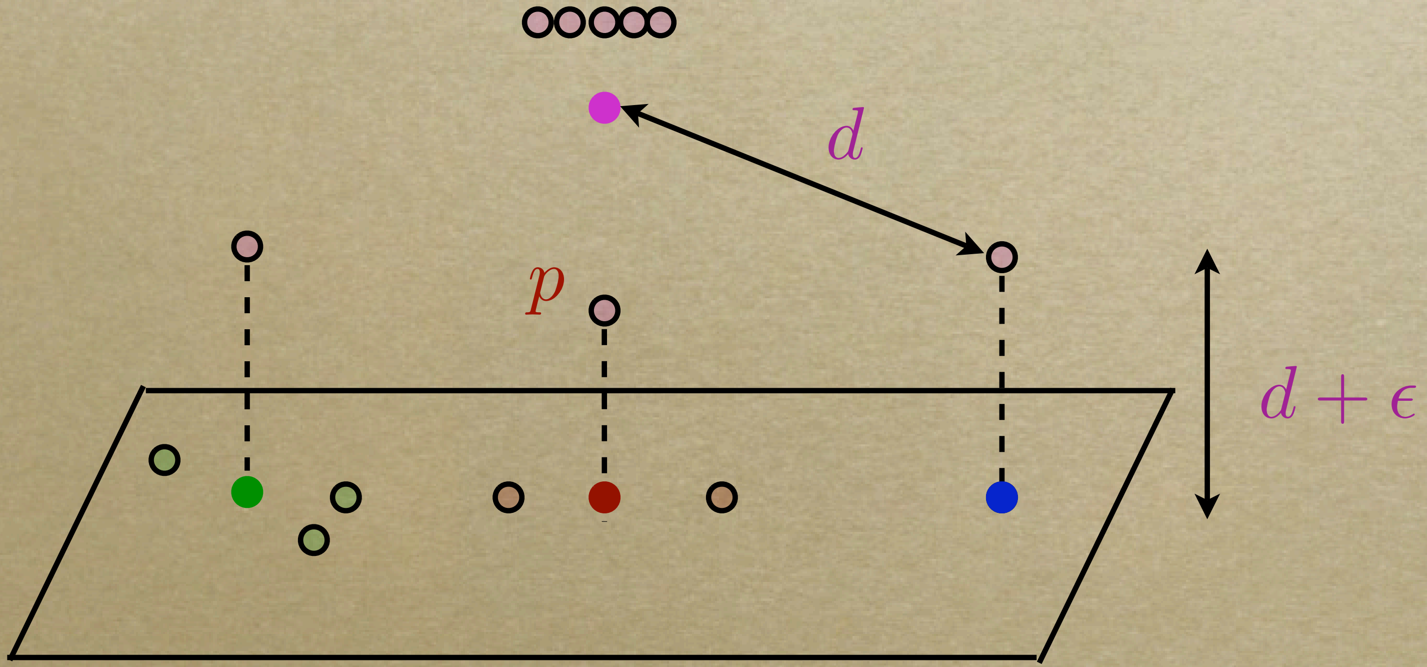
Multi-Signaling

All centers are stable before the main signaling has taken place.



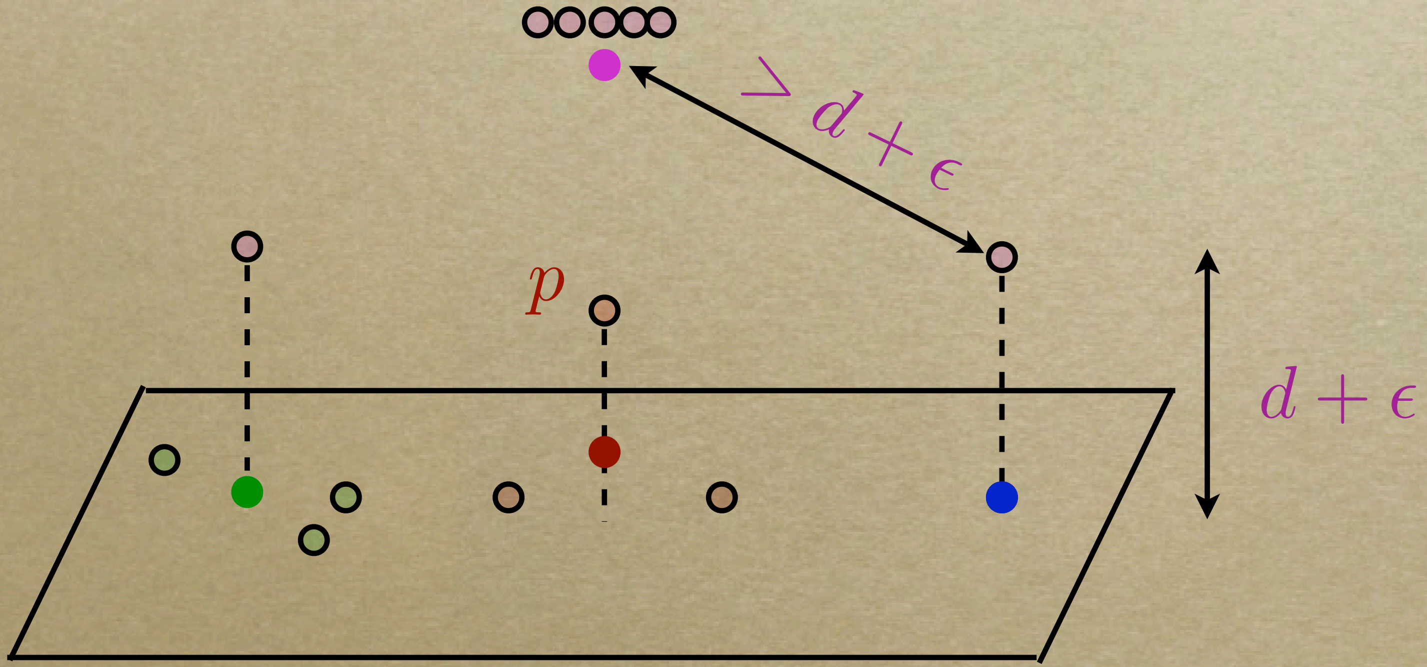
Multi-Signaling

All centers are stable before the main signaling has taken place.



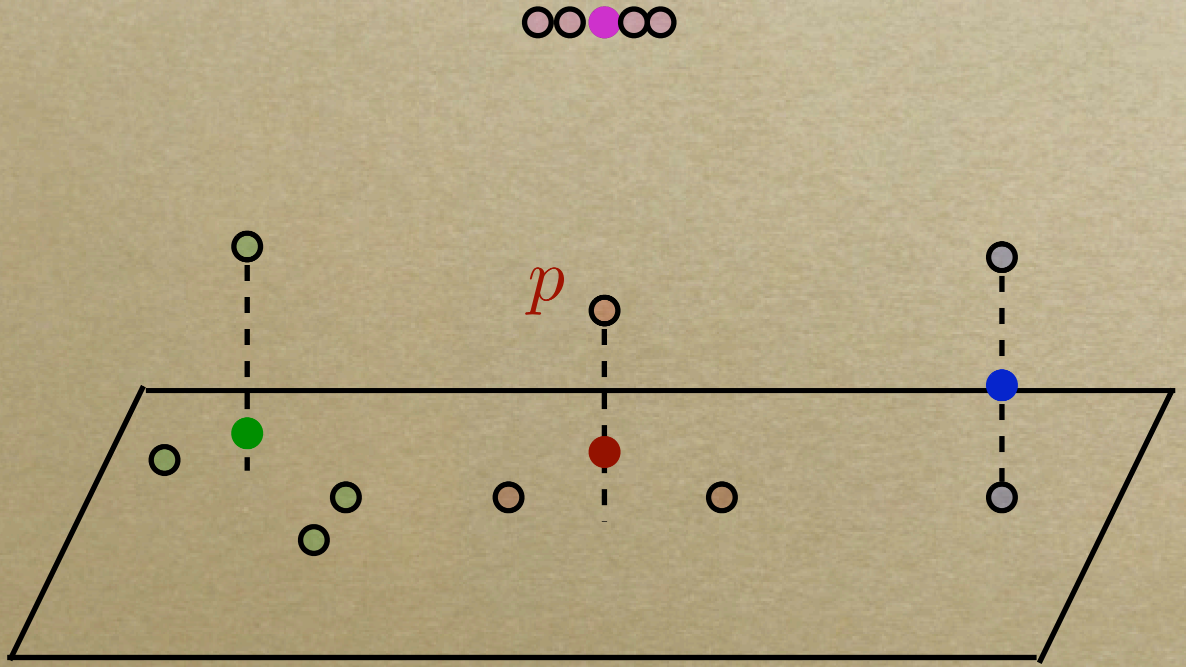
Multi-Signaling

Due to signaling the center moves away, now all centers absorb points above.



Multi-Signaling

Due to signaling the center q moves away, now all centers absorb points above. All clusters have previously unseen centers.



Put All Pieces Together

Start with a signaling configuration

Transform it, so that all clusters signal

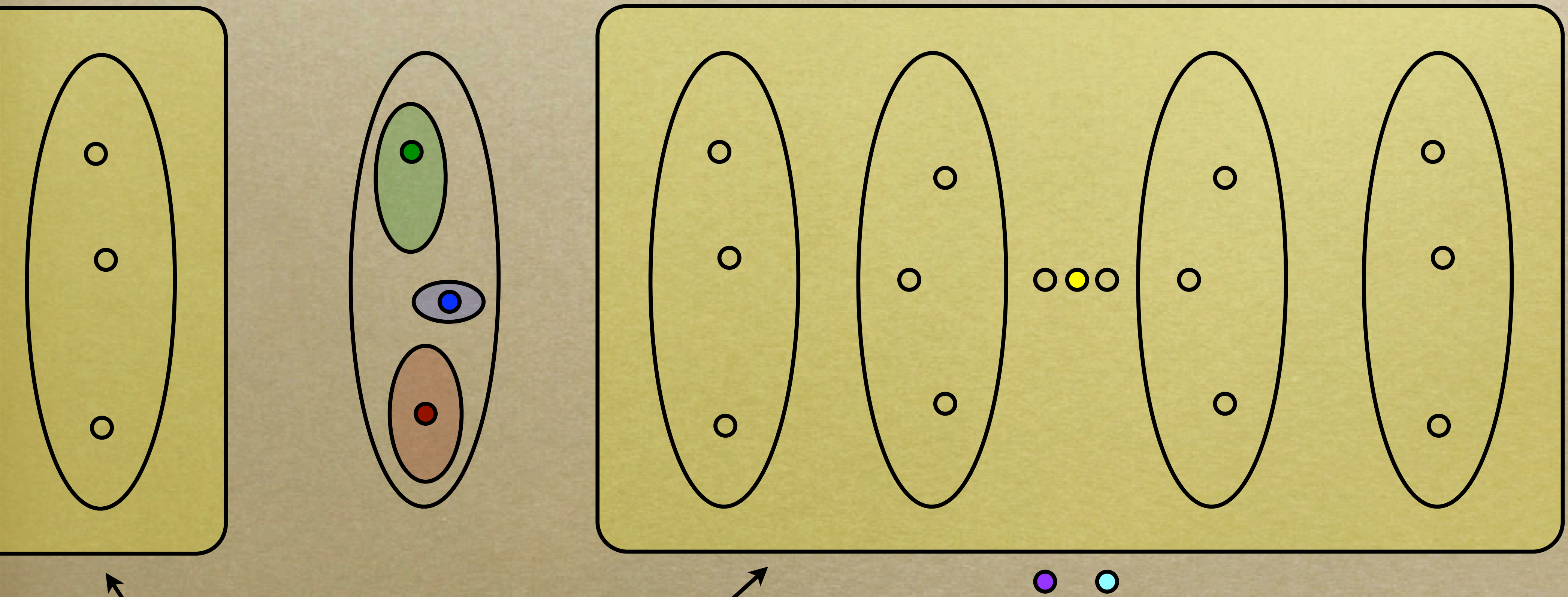
Use the new signal to reset cluster centers (and therefore double the runtime of k-Means)

Ensure the new configuration is signaling

Repeat...

Construction in Pictures

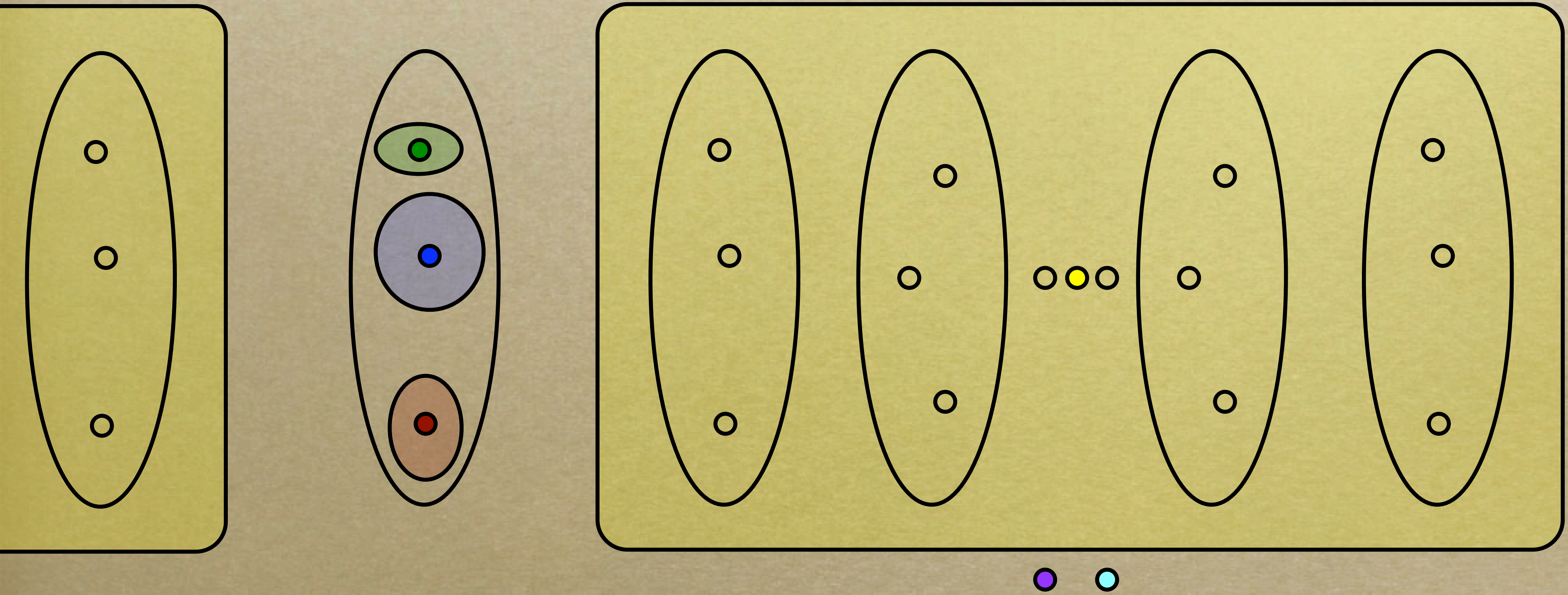
Construction:



Reflected Points

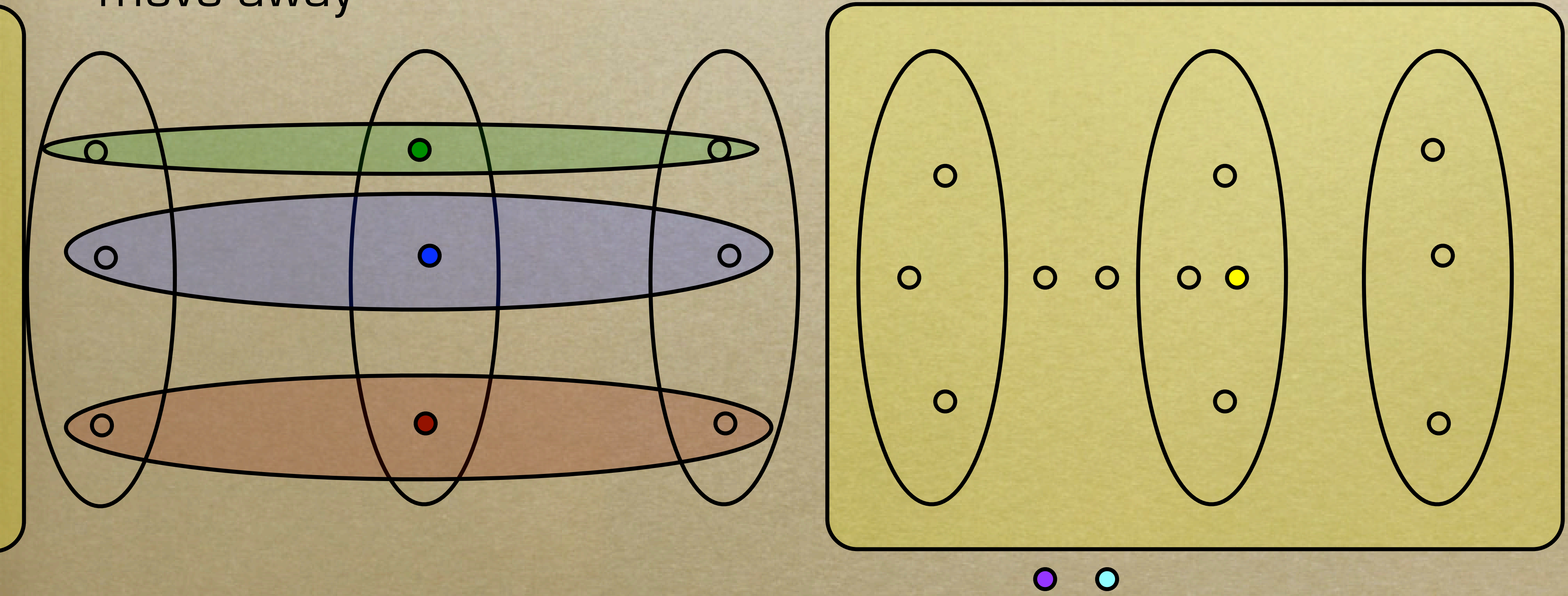
Construction in Pictures

After t steps - signal by all clusters



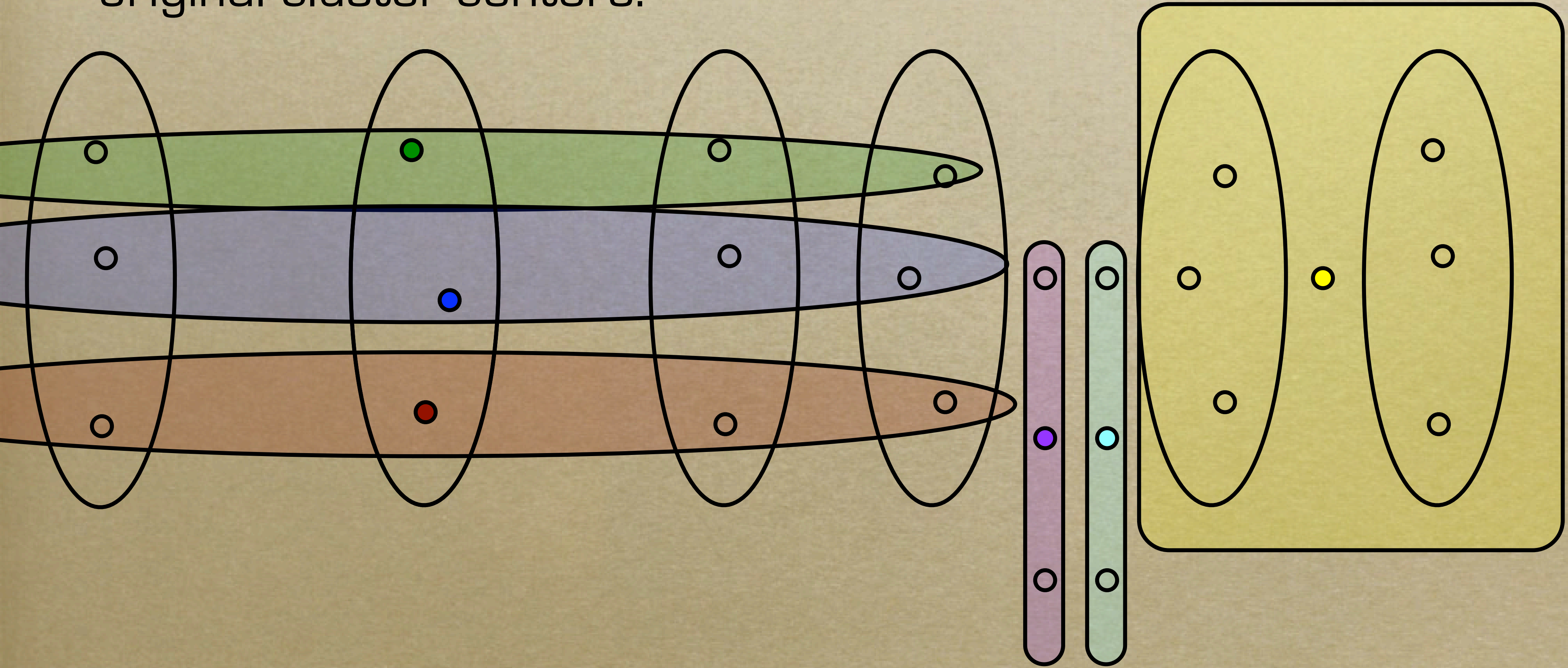
Construction in Pictures

Main Clusters absorb "catalyst" points. Yellow centers move away



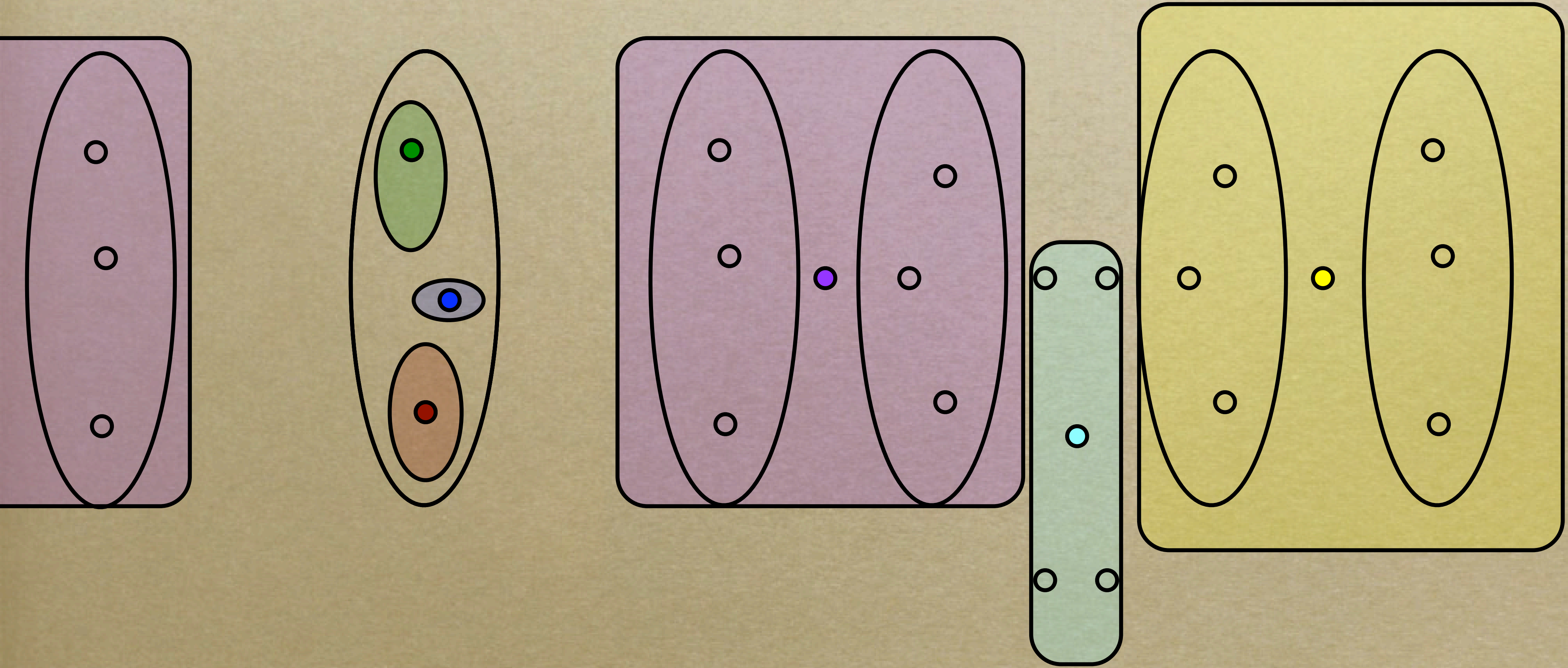
Construction in Pictures

The new points added are "reset" points - resetting the original cluster centers.



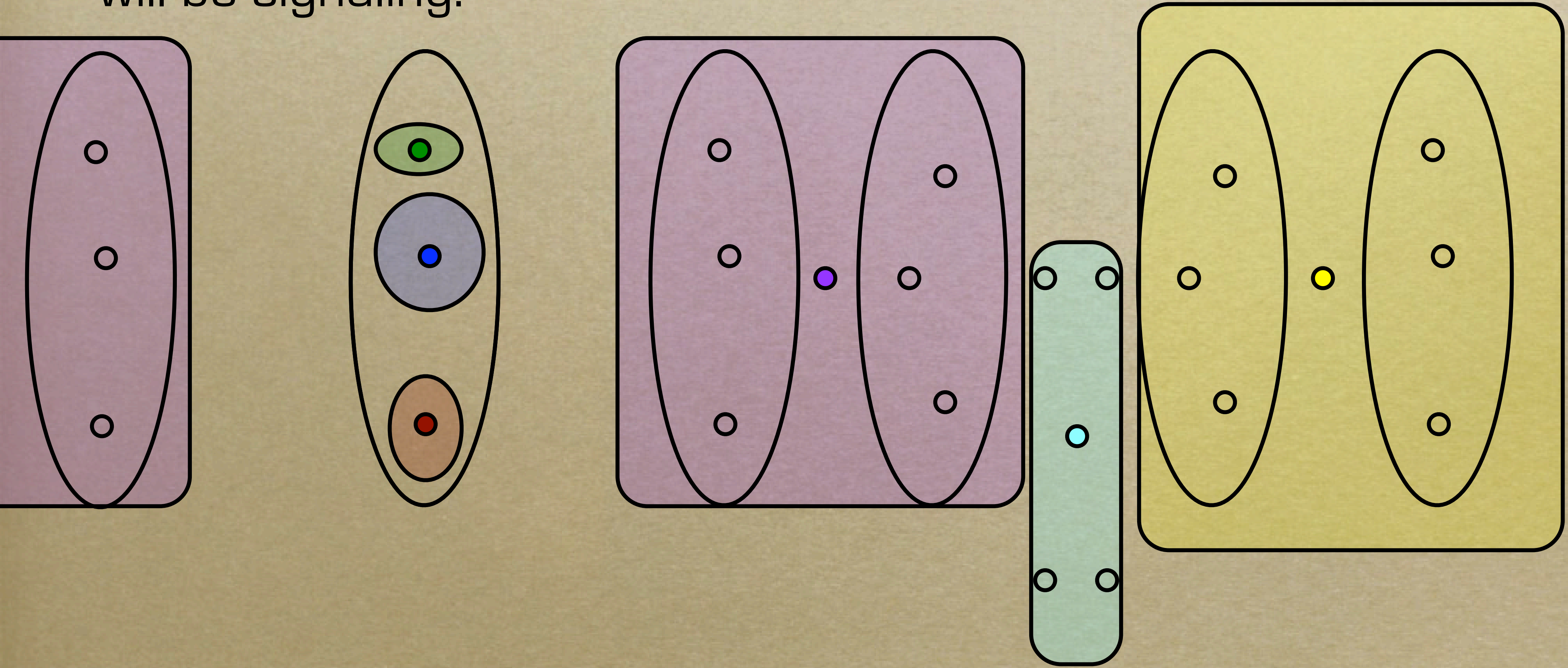
Construction in Pictures

Can ensure “catalyst” points leave the main clusters



Construction in Pictures

k-Means runs for another t steps. The original centers will be signaling.



Construction Results

If we repeat the reseting widget construction r times:

- $O(r^2)$ points in $O(r)$ dimensions
- $O(r)$ clusters
- Total running time: $2^{\Omega(r)}$

Construction Remarks

Currently construction has very large spread

- Can use more trickery to decrease the spread to be constant, albeit with a blow up in the dimension.

As presented requires specific placement of initial cluster centers, in practice centers chosen randomly from points.

- Can make construction work even in this case

Open question:

- Can we decrease the dimensionality to constant d?

Outline

- k-Means Intuition
- Lower Bound Sketch
- Upper Bound Sketch
- Open Problems

Smoothed Analysis

Assume each point came from a Gaussian distribution with variance σ^2.

- Data collection is inherently noisy
- Or add some Gaussian noise (effect on final clustering is minimal)

Key Fact: Probability mass inside any ball of radius ϵ is at most $(\epsilon/\sigma)^d$.

Potential Function

Use a potential function:

$$\Phi(C) = \sum_{x \in X} \|x - C(x)\|^2$$

Original Potential at most nD^2

Potential decreases every step.

- Reassignment reduces $x - C(X)$
- Center recomputation finds optimal Φ for the given partition

Potential Decrease

Lemma Let S be a pointset with optimal center $c^*$and c be any other point then:

$$\Phi(c) - \Phi(c^*) = |S| \|c - c^*\|^2$$

$$\Phi(c) = \sum_{x \in S} (x - c) \cdot (x - c)$$

$$= \sum_{x \in S} (x - c + c^* - c^*) \cdot (x - c + c^* - c^*)$$

$$= \sum_{x \in S} (x - c^*) \cdot (x - c^*) + (c - c^*) \cdot (c - c^*) + 2(c - c^*) \cdot (x - c^*)$$

$$= \Phi(c^*) + |S| \|c - c^*\| + 2(c - c^*) \cdot \sum_{x \in S} (x - c^*)$$

Main Lemma

In a smoothed pointset, fix an $\epsilon > 0$. Then with probability at least $1 - 2^{2n}\left(\frac{\epsilon}{\sigma}\right)^d$ for any two clusters S and T with optimal centers $c(S)$ and $c(T)$ we have that:

$$\|c(S) - c(T)\| \geq \frac{\epsilon}{2\min(|S|, |T|)}$$

Proof Sketch

Suppose $|S| < |T|$and $x, x \in S, x \notin T$. Fix all points except for x .

To ensure $\|c(S) - c(T)\| \leq \epsilon$, x must lie in a ball of diameter $|S|\epsilon$.

Since x came from a Gaussian of variance σ^2 this probability is at most $(|S|\epsilon\sigma^{-1})^d$.

Finally, union bound the total error probability over all possible pairs of sets - $2^{2n}(\epsilon/\sigma)^d$.

Potential Drop

At each iteration, examine a cluster S whose center changed from c to c':

- $\|c - c'\| \geq \frac{\epsilon}{|S|}$
- Therefore, the potential drops by $|S|\frac{\epsilon^2}{|S|^2} \geq \frac{\epsilon^2}{4n}$
- After $m = \frac{4n^2D^2}{\epsilon^2}$ iterations, the algorithm must terminate.

To Finish Up:

Chose $\epsilon = \sigma n^{-\frac{1}{d}} 2^{-\frac{2n}{d}}$. Then the total probability of failure is: $2^{2n}(\epsilon/\sigma)^d = 1/n$.

The total running time is

$$m = \frac{4n^2 D^2}{\epsilon^2} = \mathcal{O}\left(n^{2+2/d}\left(\frac{D}{\sigma}\right)^2 2^{2n/d}\right)$$

Remark: polynomial for $d = \Omega(\frac{n}{\log n})$

Upper bound (2)

We used the union bound over all possible sets. However, due to the geometry, not all sets arise. The total number of distinct clusters that can appear is $O(n^{kd})$.

Carrying the same calculations through we can bound the total number of iterations as:

$$\mathcal{O}\left(n^{k+2/d}\left(\frac{D}{\sigma}\right)^2\right)$$

Remarks

The noise need not be Gaussian, need to avoid large probabilistic point masses.

- e.g. Lipschitz conditions are enough.

Outline

- k-Means Intuition
- Lower Bound Sketch
- Upper Bound Sketch
- Open Problems

Conclusion - Lower Bounds

Showed super-polynomial lower bound on the execution time of k-Means:

- However - construction requires many dimensions, does not preclude an $O(n^d)$ upper bound

Conclusion - Upper Bounds

Can use smoothed analysis to reduce the best known upper bounds for k-Means.

- But, is not polynomial for small values of d, or large values of k.
- Even with smoothness there is an $\Omega(n)$ lower bound, which is never observed in practice.

Thank you

Any Questions?