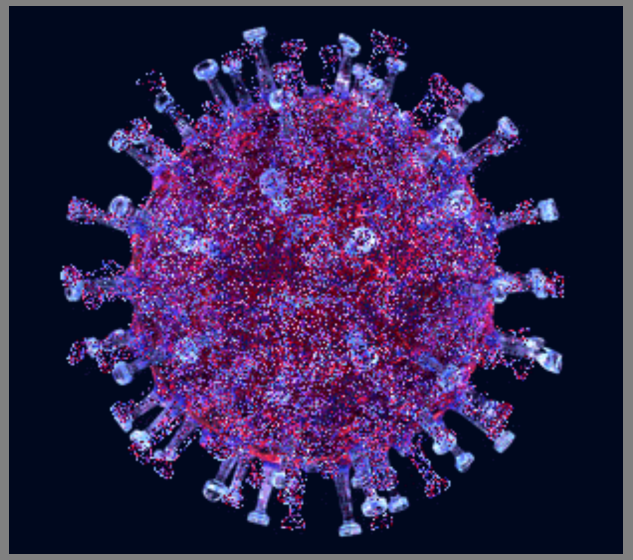
COS324: Introduction to Machine Learning

Topic: Unsupervised Learning, Clustering, Embeddings

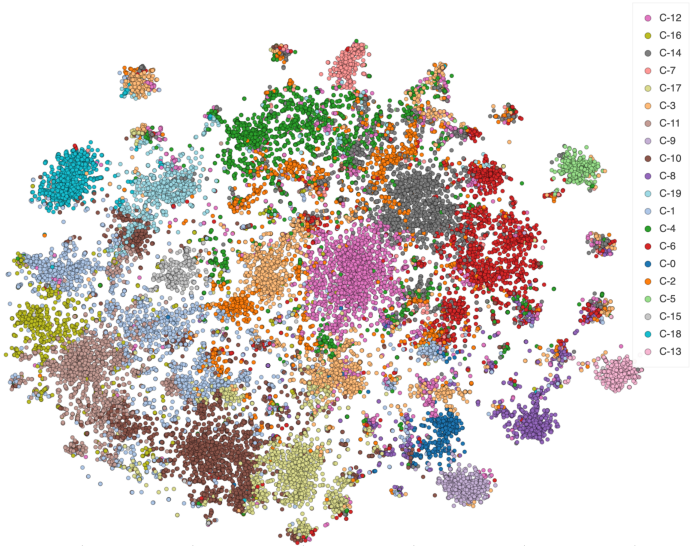
Prof. Yoram Singer

April 28, 2020

Pressing Problem



Clustering of Scientific Articles on Covid-19



Unsupervised Learning

- So far discussed supervised learning:
  - Examples $(\mathbf{x}, y)$ are input-target pairs in $\mathcal{X} \times \mathcal{Y}$
  - Learning amounts to finding a function $f : \mathcal{X} \to \mathcal{Y}$
  - Loss measures discrepancy between y and $\hat{y} = f(\mathbf{x})$: $\ell(y, \hat{y})$
- Sometimes we have plentiful of instances $\mathbf{x}_i$

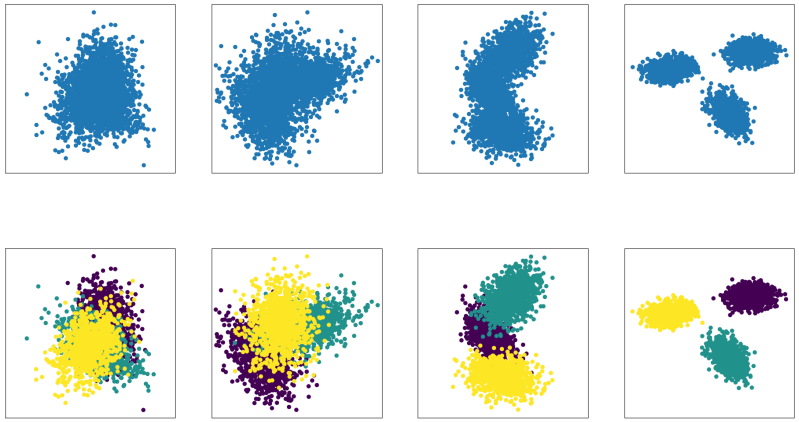
$$S = \left\{(x_i, ?)\right\}_{i=1}^{m} \cup \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{n}$$

  ... but only handful of labels $m \gg n$
  ... or none at all $n = 0$

- It is nonetheless useful to find "structure" or meaningful patterns in the data

Goals of Clustering

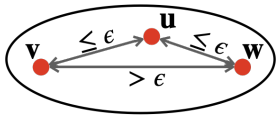
- Intuitively, grouping a set of objects such that
  - Similar objects end up in the same cluster
  - Dissimilar objects place in different clusters
- Imprecise & potentially ambiguous definition
- Disappointingly, no single simple definition



Sources of Difficulty

- Inherit problem: lack of "ground truth" and tangible objective
- Technical difficulty:
  - Similarity & distance functions are not transitive

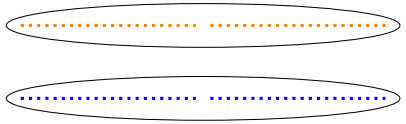
$$\|\mathbf{u}-\mathbf{v}\| \le \varepsilon \ \wedge \ \|\mathbf{u}-\mathbf{w}\| \le \varepsilon \quad \not\Rightarrow \quad \|\mathbf{v}-\mathbf{w}\| \le \varepsilon$$



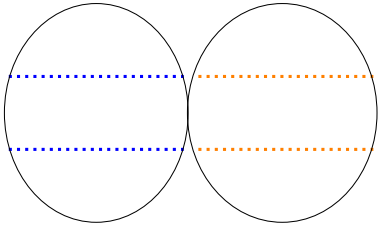
  - Cluster (set) membership is transitive
    - Define $\mathbf{u} \sim \mathbf{v}$ iff $\mathbf{u}$ and $\mathbf{v}$ belong to the same cluster
    - Then, $\mathbf{u} \sim \mathbf{v} \ \wedge \ \mathbf{v} \sim \mathbf{w} \quad \Rightarrow \quad \mathbf{u} \sim \mathbf{w}$

Clustering is Ambigious

similar objects in same cluster

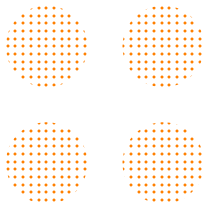


dissimilar objects are separated

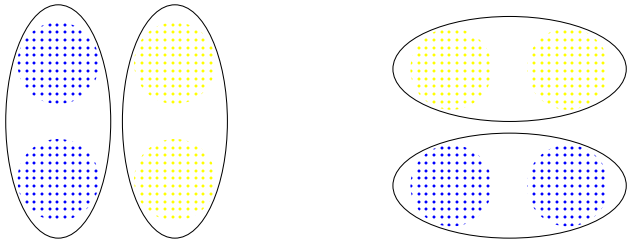


Lack of Ground Truth

Partition points into **two** clusters:



We have two well justifiable solutions:



Clustering Problem Setting

- Input: $S = \{\mathbf{x}_i\}_{i=1}^{m}$ where $\mathbf{x}_i \in \mathbb{R}^d$
- Distance $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ or similarity $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ $\left[s \text{ need not be symmetric, } s(\mathbf{u},\mathbf{v}) \neq s(\mathbf{v},\mathbf{u})\right]$
- Output: partition $\mathcal{C} = \{C_i\}_{i=1}^{k}$ of set S into k subsets s.t.

$$S = \bigcup_{i=1}^{k} C_i \qquad\qquad C_i \cap C_j = \emptyset$$

- Number of clusters k may be part of input or unkown

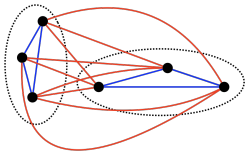
Cost-based Clustering

- We focus on distance-based $d(\mathbf{u},\mathbf{v})$ clustering
- Cost of partitioning $\mathcal{C} = \{C_i\}_{i=1}^{k}$ of S ?
- Define indicator

$$\mathbb{1}[i,j|\mathcal{C}] = \begin{cases} +1 & \exists r : \mathbf{x}_i \in C_r \wedge \mathbf{x}_j \in C_r \\ -1 & \text{o.w.} \end{cases}$$

- Penalize for large intra-cluster & small inter-cluster distances

$$\ell(S,\mathcal{C}) = \sum_{i,j=1}^{|S|} \mathbb{1}[i,j|\mathcal{C}]\, d(\mathbf{x}_i,\mathbf{x}_j)$$



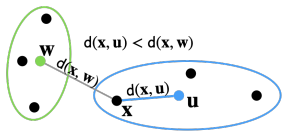
- Number of instances to compare $O(n^2)$

$$|C_i| \approx \frac{n}{k} \Rightarrow \begin{array}{ll} \binom{k}{2}\left(\frac{n}{k}\right)^2 \equiv O(n^2) & \text{inter-cluster pairs} \\ k\binom{\frac{n}{k}}{2} \equiv O\left(\frac{n^2}{k}\right) & \text{intra-cluster pairs} \end{array}$$

k-Center Clustering

- Centroid-based clustering: intuitive, transitive, "aesthetic"
- Associate a center $\mathbf{w}_j \in \mathbb{R}^d$ with partition C_j

$$\mathbf{x}_i \in C_j \quad \Leftrightarrow \quad \forall r \neq j : \; d(\mathbf{x}_i,\mathbf{w}_j) < d(\mathbf{x}_i,\mathbf{w}_r)$$



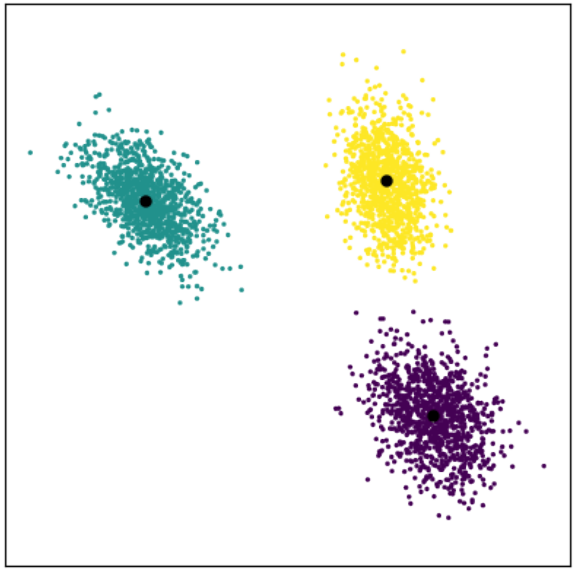
- Induces partitioning

$$C_j = \left\{i : \forall r \neq j \; d(\mathbf{x}_i,\mathbf{w}_j) < d(\mathbf{x}_i,\mathbf{w}_r)\right\}$$

- Loss of k-centers

$$\ell(S,\mathcal{C}) = \sum_{j=1}^{k}\sum_{i\in C_j} d(\mathbf{x}_i,\mathbf{w}_j) \;=\; \sum_{i=1}^{m} \min_{j=1}^{k} d(\mathbf{x}_i,\mathbf{w}_j)$$

Example of 3-Center Clustering



Skeleton of Metric Clustering

- Intialize each $\mathbf{w}_j^0$ to a vector in $\mathbb{R}^d$
- For $t = 1, \ldots, T$:
  - Associate each $\mathbf{x}_i$ with its nearest centroid

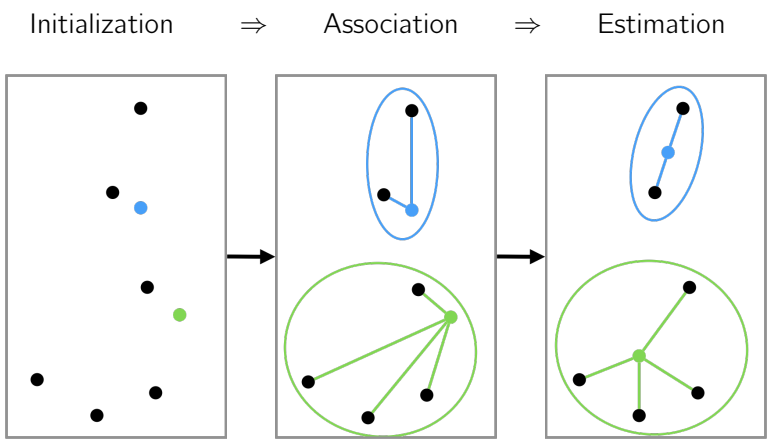
$$\forall i: \; a^t(i) = \arg\min_{j=1}^{k} d(\mathbf{x}_i, \mathbf{w}_j^{t-1})$$

  - Restimate centroids from associations

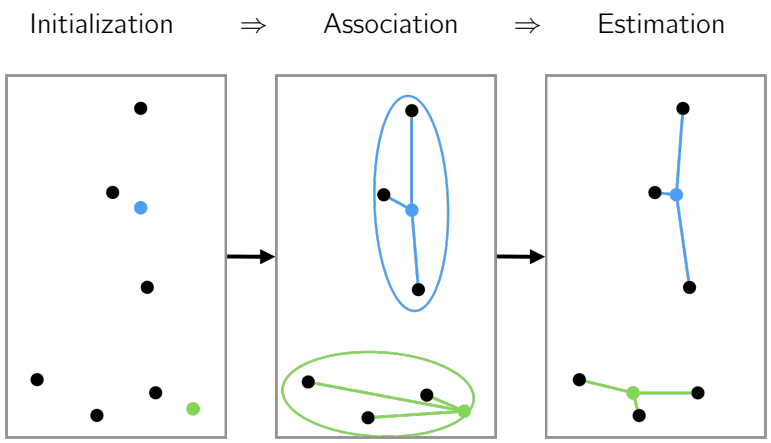
$$\forall j: \; \mathbf{w}_j^t = \min_{\mathbf{w}} \sum_{i:a^t(i)=j} d(\mathbf{x}_i, \mathbf{w})$$

  - If $\forall i: \; a^t(i) = a^{t-1}(i)$ break

Initialization makes a difference: Example I.a



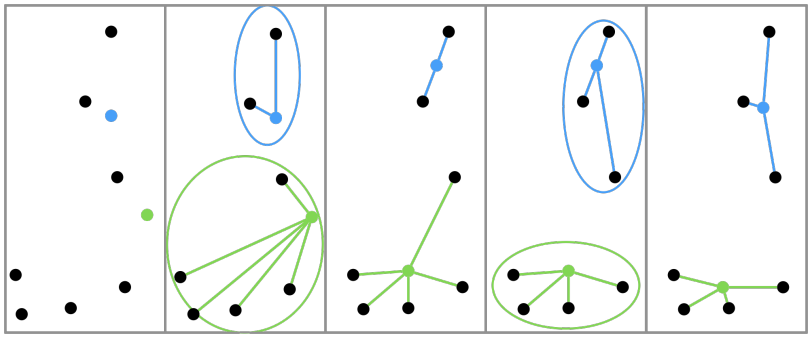
Initialization makes a difference: Example I.b



Example II

A single example might change clustering entirely:

Initialization $\Rightarrow$ Associate $\Rightarrow$ Estimate $\Rightarrow$ Associate $\Rightarrow$ Estimate



Convergence of Metric-based Clustering

- Centers at iteration t

$$\mathcal{W}^t = \{\mathbf{w}_j^t\}_{j=1}^k$$

- Partition at iteration t

$$\mathcal{A}^t = \{a^t(i)\}_{i=1}^m$$

- Loss of partition and centers

$$\ell(S, \mathcal{A}, \mathcal{W}) = \frac{1}{m}\sum_{i=1}^m d(\mathbf{x}_i, \mathbf{w}_{a(i)})$$

- Then, $\ell\big(S, \mathcal{A}^{t-1}, \mathcal{W}^{t-1}\big) > \ell\big(S, \mathcal{A}^{t}, \mathcal{W}^{t-1}\big) > \ell\big(S, \mathcal{A}^{t}, \mathcal{W}^{t}\big)$

- Since $\ell(S, \mathcal{A}, \mathcal{W}) \geq 0$ and $\forall t, j : \mathbf{w}_j^t \in \bar{S}$
  $\Rightarrow$ $\ell\big(S, \mathcal{A}^{t}, \mathcal{W}^{t}\big)$ converges to a local minimum

k-Means

- Use $d(\mathbf{u}, \mathbf{v}) \stackrel{\text{def}}{=} \|\mathbf{u} - \mathbf{v}\|_2^2$

- Solving $\min_{\mathbf{w}} \sum_{i:a(i)=j} \|\mathbf{x}_i - \mathbf{w}\|^2$ amounts to

$$\mathbf{w}_j = \frac{1}{n_j}\sum_{i:a(i)=j} \mathbf{x}_i \quad \text{where} \quad n_j \stackrel{\text{def}}{=} \big|\{i : a(i) = j\}\big|$$

- Namely, center of mass of examples in cluster

- Runtime is: $T\,k\,n$

k-Medians

- Use $d(\mathbf{u}, \mathbf{v}) \stackrel{\text{def}}{=} \|\mathbf{u} - \mathbf{v}\|_1$

- Solving $\min_{\mathbf{w}} \sum_{i:a(i)=j} \|\mathbf{x}_i - \mathbf{w}\|_1$ amounts to

$$\begin{aligned}\mathbf{w}_j[r] &= \min_{\omega} \sum_{i:a(i)=j} \big|\mathbf{x}_i[r] - \omega\big| \\ &= \text{median}\big\{\mathbf{x}_i[r] \;:\; a(i) = j\big\}\end{aligned}$$

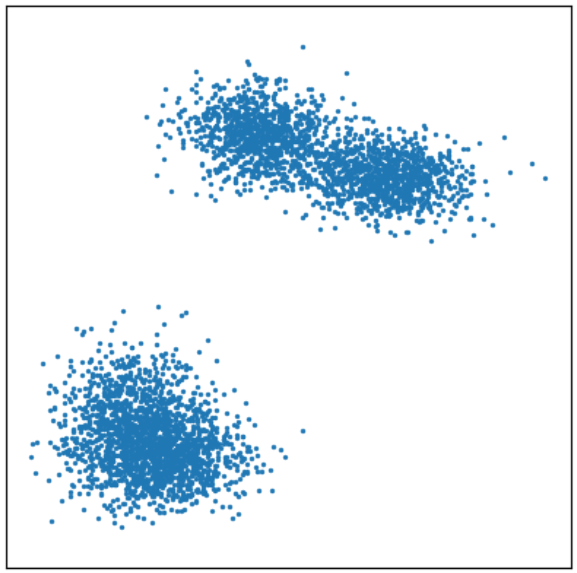
- $\mathbf{w}_j[r]$ is median of r'th coordinate of examples in cluster

- Runtime is: $T\,k\,n$

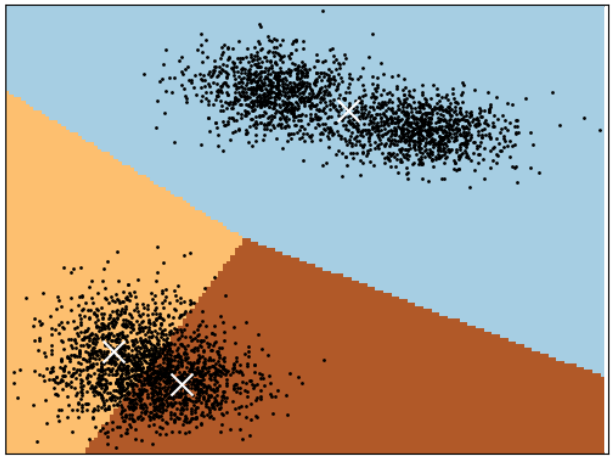
Tricks & Treats

- Initialization:
  - At random
  - Agglomeratively: warm-start from $k-1$ clusters
  - Agglomeratively: hierarchical from $2 \times \frac{k}{2}$ clusters
  - Using other clustering methods (e.g. spectral)
- Art of choosing number of clusters k ...
- Small amounts of labeled data:
  - Determine number of clusters
  - Good initialization
  - Metric adjustment prior to clustering

Data Generated by k Gaussians

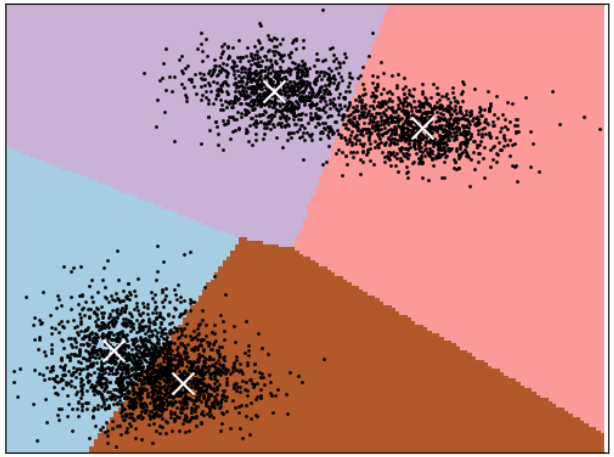


Clustering with $\hat{k} = 3$

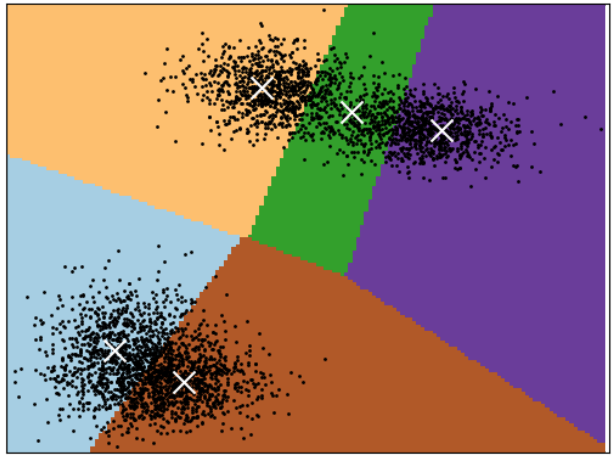


Why do we see straight decision boundaries ?

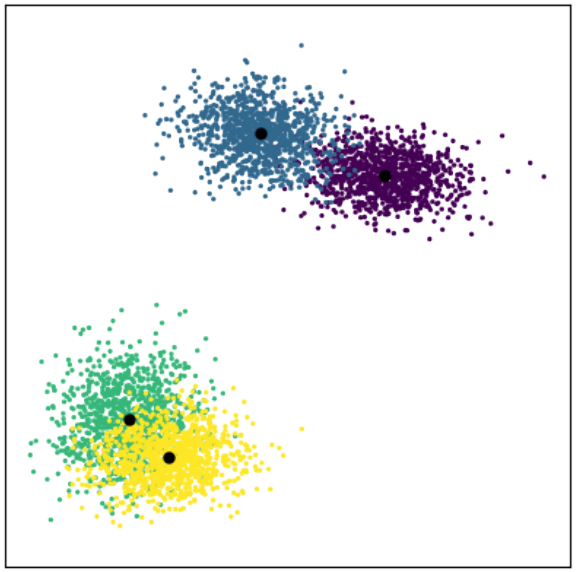
Clustering with $\hat{k} = 4$



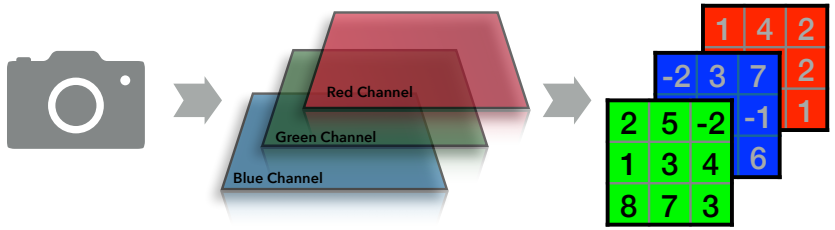
Clustering with $\hat{k} = 5$



Means of Data Generetor



Color Segmentation Using Clustering



- Color images are captures using a CCD device
- Resulting image: 3-dimensional $3 \times n_x \times n_y$ tensor M
- Intensity for color-channel c at location (i, j) in image: $M[c, i, j]$
- $M[c, i, j] \in [p]$ where p ranges from 2^8 ("sdef") to 2^{24} ("true")
- Color pallet is $p^3 \gg n_x \times n_y \Rightarrow$ compress image?

Color Segmentation Using Clustering

- View each pixel as a vector in $\mathbb{R}^3$
- Flatten image, creating a matrix X of $n_x \times n_y$ rows
- Run k-Means on X where $k \ll p^3$
- Resulting k centroids are also colors
- Replace each original pixel with its nearest centroid
- Requires $3pk$ bits to store the exact values of centroids
- Each pixel is replaced with index ($\log(k)$ bits) of its centroid
- Number of bits:

Compressed	$3pk + n_x n_y \log(k)$
Original	$n_x n_y p^3$

- Moreover, images are naturally segmented by color

▸ Code (jn –port 9999)

In [1]:
```
# General imports

import urllib
import matplotlib.pyplot as plt
from matplotlib.image import imread
from sklearn.cluster import KMeans
```

In [2]:
```
# Data Loading

droot = "https://raw.githubusercontent.com/ageron/handson-ml2/master/"
filename = "ladybug.png"
url = droot + "images/unsupervised_learning/" + filename
with urllib.request.urlopen(url) as f:
    image = imread(f)
    X = image.reshape(-1, 3)
```

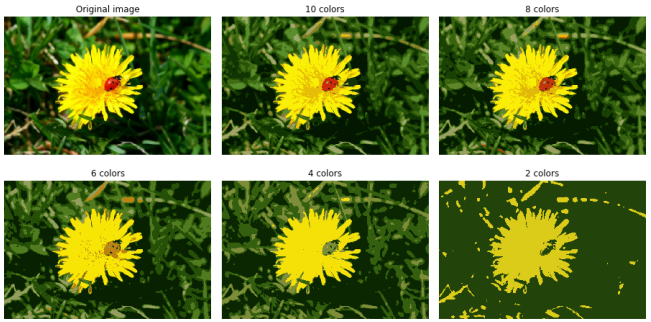
In [3]:
```
# Clustering using sklearn.KNeans and then using centroids for segmentation

segmented_imgs = []
n_colors = (10, 8, 6, 4, 2)
for n_clusters in n_colors:
  kmeans = KMeans(n_clusters=n_clusters, random_state=17).fit(X)
  segimg = kmeans.cluster_centers_[kmeans.labels_]
  segmented_imgs.append(segimg.reshape(image.shape))
```


In [4]:
```
# Plot images replacing each pixel with each nearest centroid

plt.figure(figsize=(16,8))
plt.subplots_adjust(wspace=0.05, hspace=0.1)
plt.subplot(231)
plt.imshow(image)
plt.title("Original image")
plt.axis('off')

for idx, n_clusters in enumerate(n_colors):
  plt.subplot(232 + idx)
  plt.imshow(segmented_imgs[idx])
  plt.title("{} colors".format(n_clusters))
  plt.axis('off')
```



Finished $\not\Rightarrow$ Complete

When you learn from the wrong person you are finished.
When you learn from the right person you are complete.

[circa 1995]

- Decision trees
- Temporal Modeling
  (Markov Models & Random Fields, Recurrent Networks)
- Ensemble Methods & Boosting
- Dimensionality Reduction
- Planning & Markov Decision Processes