Lecture 10
3/3/15
clustering algs: k-means, single-link

Outline: clustering: ~~partition into~~ (for now): partition into k non-empty disjoint sets. (1)

k-means

- prove equivalence of within-cluster scatter (1) and w/in cluster variation (2)
  - within-cluster scatter → looks good, but computationally hard
  - w/in cluster variation → suggests an algorithm, but why is this optimizing a "good thing"?

<resolve Δ inequality "paradox">

- give k-means alg, including examples.
- show k-means is improving its opt. criterion (no guarantee of optimality, tho')
- ~~show some successes? failures? (if we want for handout? or just point to web pgs?)~~

single-link

- consider alternate criterion:
  - mention problem w/ (6) last time

  spacing: maximize between-cluster distance, defined as

$$d(C_j, C_{j'}) = \min_{x \in C_j,\, x' \in C_{j'}} \|x - x'\|_2^2$$

(actually, other ~~metrics~~ functions ok here, too, only require $f(x,x)=0$, $f(x,x') = f(x',x)$, $f(x,x') > 0$ if $x' \neq x$.)

- single-link clustering finds an optimal solution
- behaviors

Announcements:

- typo in A1 Q2 re: error bars. Ok if you used "typo" version – don't redo.
- updated lec 9 handout posted ← or just print new copy?

From last time:

n data points $x_1, \dots, x_n$

K clusters $C_1, \dots, C_K$;

for cluster C_j,

n_j = # points in C_j. Req'd to be > 0.

$r_j = \frac{1}{n_j} \sum_{x \in C_j} x$, the centroid.

think of "r" as standing for "representative".

(1) within-cluster scatter:

$$\sum_j \sum_{\substack{x_t, x_s \in C_j \\ t < s}} \|x_t - x_s\|_2^2$$

(2) within-cluster variation

$$\sum_j n_j \sum_{x \in C_j} \|x - r_j\|_2^2$$

(8) lemma: for any point z,

$$\sum_{x \in C_j} \|x - z\|_2^2 = \left(\sum_{x \in C_j} \|x - r_j\|_2^2 \right) + n_j \|r_j - z\|_2^2$$

$\boxed{\text{k-means}}$

<argument follows Hopcroft & Kannan 2014, section 8.2>

Idea: [I]. (1) - general, intuitive, but computationally hard?

[II]. if (8) were true, then (1) ≡ (2)
(we'll omit the pf, which is just algebra)

~~looks easy~~ fewer things to keep track of, @ any rate.

[III] • ~~proof of (8)~~
(2) suggests an algorithm

[IV] proof of (8) shows why the re-computation of centroids improves (2)

k-means algorithm Start with some initialization $\hat{\mathbf{r}}_j{}^0$, $j \in 1, \ldots, K$ (superscripts = iteration number, we start with $i = 0$). Repeat until "convergence":

1. Assign each $\mathbf{x}$ to its nearest *representative* $\hat{\mathbf{r}}_j{}^i$.
2. Set $\hat{\mathbf{r}}_j{}^{i+1}$ to be the centroid of the $\mathbf{x}$s now assigned to it.
3. Increment i

---

single-link algorithm We start with n clusters, each containing a single point. Here, think of clusters as connected components in an undirected graph. Repeat until there are only K clusters:

1. Add an edge between the two points x and $\mathbf{x}'$ in different clusters that have the minimum distance between them, thus merging their component clusters.