Evaluation

We'll return to features and distance measures later but let's first consider evaluation.

Evaluation: How do you determine the effectiveness of design choices:

- features
- distance measure
- <for project: spectroscan computation>

Given a method $d(\cdot,\cdot)$ for computing the similarity between two images, how do we evaluate it.

If we had a set of evaluation images for which we had the true similarity value, we could just compute the differences with our "predicted" value.

For a given query image Q, if we had v_i corresponding to how similar Q is to evaluation image T_i then we could compute:

$$\text{error}_{\text{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}\left(v_i' - v_i\right)^2}$$

where n is # of evaluation images

$$v_i = d(Q, T_i)$$

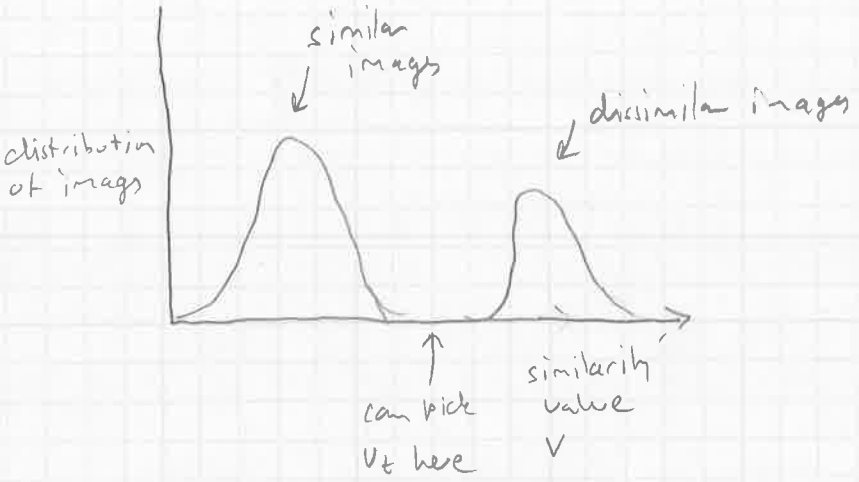
However, difficult to determine true similarity values v_i'. Also, would need to do it for each query image Q.

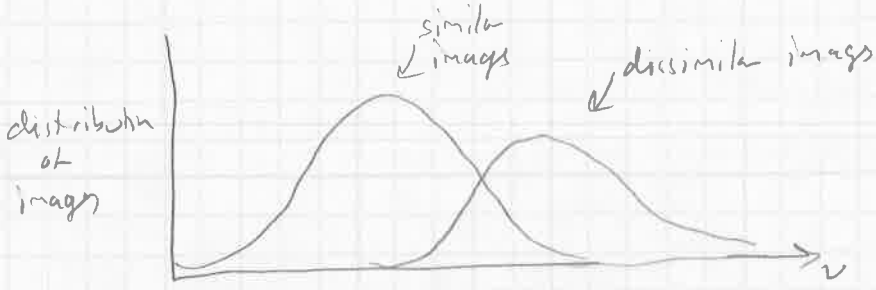
Note, that even if our predicted similarity values matched the true values, in order to retrieve the similar images from a target dataset, we could still need to determine what "similar" means in terms of the similarity value.

That is, we need to determine a threshold v_t so we can issue the query

Given Q, return all T_i such that $d(Q, T_i) < v_t$.

This is only possible if the similarity value for similar images does not overlap the value for dissimilar images.





can't pick value V_t that separates similar and dissimilar values.

Suppose that instead of knowing true similarity value for image, we know or can determine whether it is similar or not to a query image.

Concepts of precision and recall.

Remember, we have two types of queries:

ε-query: return all T_i such that $d(Q, T_i) < \varepsilon$

kNN query: return k most similar T_i to Q.

These are, of course, related. There is a k corresponding to every setting of ε and there is a (range of) ε corresponding to every value of k.

Precision

Suppose we retrieve k target images
(either because we performed a kNN query or k images were returned based on our setting of ε)

Precision is then the proportion of retrieved images that are similar to our query.

This can be computed as the number of retrieved images that are similar divided by the number of retrieved images

$$\text{precision} = \frac{\#\text{ retrieved images that are similar to query}}{k} \in [0,1]$$

A value of 1 is "perfect".

Again, we can compute this if we know or can determine which images are similar to our query.

If our framework works perfectly then, as we increase k (equivalently ε), precision will stay at 1 until we have retrieved all similar images in our target set. Then, it will necessarily drop.

If our framework does not work perfectly, i.e., we assign lower distance values to dissimilar images than to similar images, then precision will decrease before we have retrieved all similar images.

Note that to compute precision, we only need to determine if retrieved images are similar to the query, we don't need to know the similarity label the target images.

Recall

Suppose we know the similarity label of all the target images.

Then, recall is the proportion of similar images that were retrieved.

Recall can be computed as the number of retrieved images that are similar divided by the number of similar images in the target dataset

$$\text{recall} = \frac{\#\text{ images that are similar to query}}{\#\text{ images in target set that are similar}} \in [0,1]$$

A recall value of 1 indicates we have retrieved all of the similar images.

A recall value of 0 indicates we have retrieved none of the similar images.

As k increases (ε increases), recall increases.

If there are n images in our target set then

recall = 1 when $k = n$.

(Set explanation of precision and recall on page 10)

Precision vs. recall

Note that precision and recall depend on k, the size of our retrieval set.

That is, a particular CBIR framework does not have a single precision and/or recall value.

Instead, to compare two frameworks (different features, distance measures, etc.) we compare how precision varies with recall.

This is called precision vs. recall and is usually a plot.

We vary recall by starting with $k=0$ and increasing it until $k=n$, the number of images in our target set.

Note that area under PR curve gives summary of PR as k varies.

4/18/19

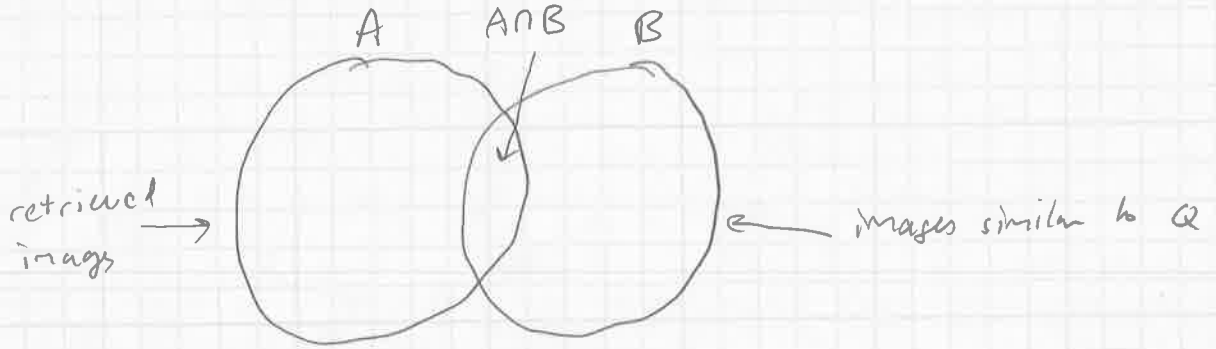
on page 9

Set interpretation of precision and recall

For a particular query image Q and a set of retrieved images, let

A = set of retrieved images

B = set of target images that are similar to Q



A∩B: set of retrieved images that are similar to Q

$$\text{precision} = \frac{|A \cap B|}{|A|}$$

where $|x|$ is the number of elements in set x.

$$\text{recall} = \frac{|A \cap B|}{|B|}$$

Note that PR curve doesn't necessarily tell you how to pick threshold V_t. But, it tells you the tradeoff (between precision and recall) as you vary V_t.

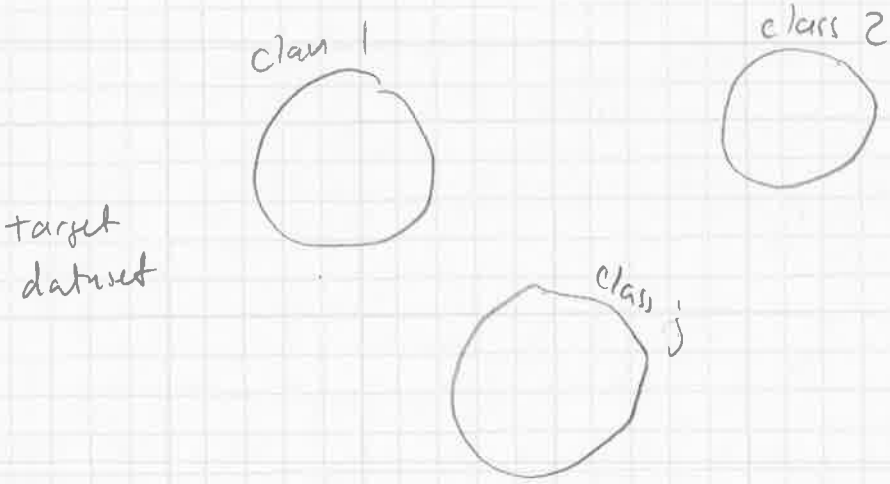
To compute precision and recall, we need to know whether each of our target images is similar to the query image.

And, usually want to evaluate framework using multiple query images.

Rather than assigning a similar/dissimilar binary label to every test image for each query image, we group our test images into classes.

The assumption is that two images are similar if they have the same class label.

The number of classes can be 2 or more.



For a query image with class label C, a target image is considered similar if it also has class label c.

It is considered dissimilar if it doesn't have class label c.

Limitations of PR curves

Remember that can't compute recall if don't know similarity of all target images.

Image search on the internet.
- how do you evaluate it since don't know set of all similar images?
- you can't use PR

Computing PR curves

How to compute PR curves particularly so they can be averaged over multiple queries which possibly have varying number of similar images in the evaluation dataset (the classes have different numbers of images)

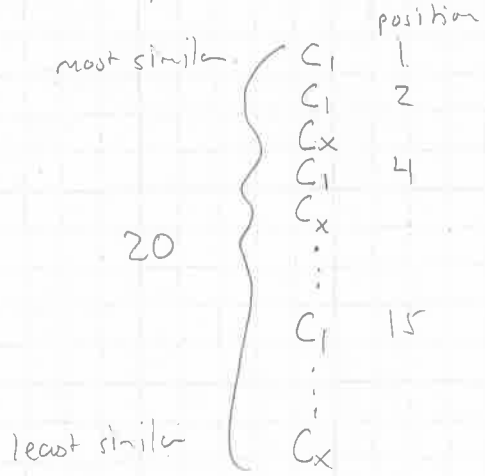
Suppose
- Our query image has class C_1
- There are 20 images in our evaluation dataset and we know the class of each.
- There are four images in the evaluation dataset with class C_1 – that is, they are similar to the query. All the rest have different classes and are thus dissimilar.

Steps:

1) Compute distance $d(Q, T_i)$ $i=1,\ldots,20$ for each image in evaluation dataset.

2) Order these in decreasing similarity (increasing distance).

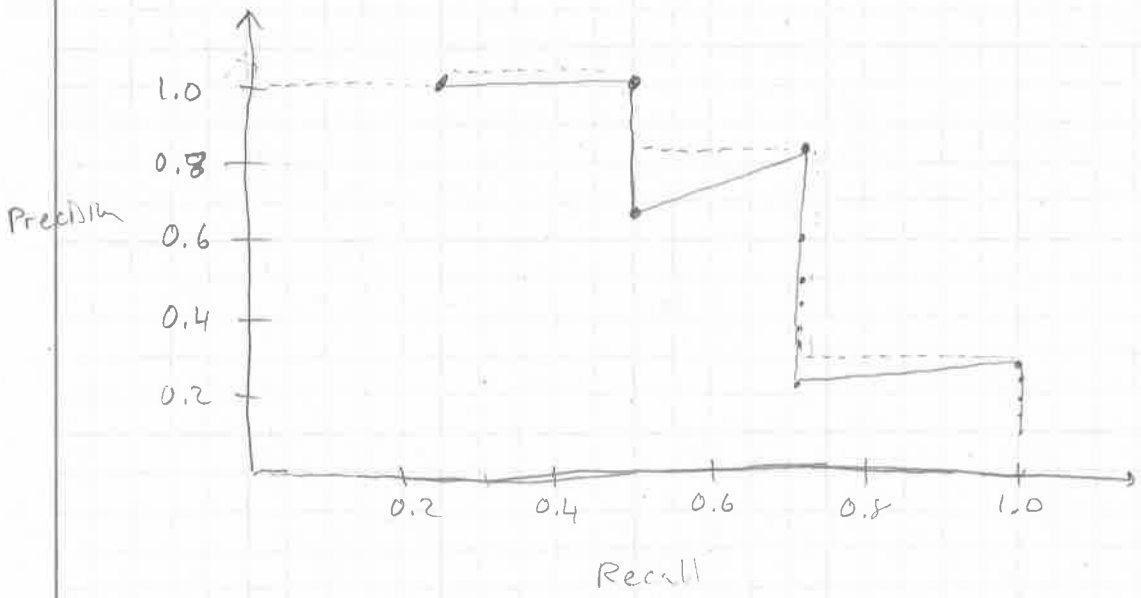
Suppose the similar images appear at positions 1, 2, 4 and 15 in this ranking



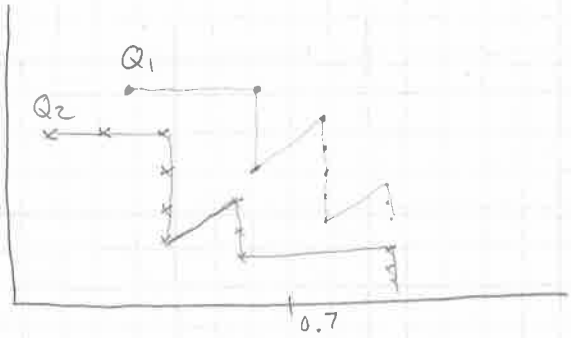
Compute precision and recall for each set of top k retrievals as k: 1 to 20.

(We'll show how this can be simplified)

k	Precision	Recall
1	1/1 = 1	1/4 = 0.25
2	2/2 = 1	2/4 = 0.5
3	2/3 = 0.67	2/4 = 0.5
4	3/4 = 0.75	3/4 = 0.75
5	3/5 = 0.6	3/4 = 0.75
6	3/6 = 0.5	3/4 = 0.75
7	⋮	⋮
8		
9		
10		
11		
12		
13		
14	3/14 = 0.21	3/4 = 0.75
15	4/15 = 0.27	4/4 = 1
16	4/16 = 0.25	4/4 = 1
17	⋮	⋮
18		
19		
20	4/20 = 0.2	4/4 = 1



Note: will have a different set of discrete recall values if the number of similar images in the evaluation set is different (say, for a different query image).



What is overall precision here over queries Q_1 and Q_2?

We could compute this as the average of the precision of Q_1 and Q_2 at recall = 0.7 if we had those values but we don't.

So need to interpolate between PR points for individual queries.

Conceptually, precision should monotonically decrease as recall increases.

This motivates the following interpolation scheme.

Let R_i be the set of recall points with precision value P_i

i	R_i	P_i
1	0.25	1
2	0.5	1
3	0.75	0.75
4	1	0.27

then

$$p(r) = P_i \quad \text{for} \quad R_{i-1} < r \leq R_i$$

Set $R_0 = 0$

<return to plot>

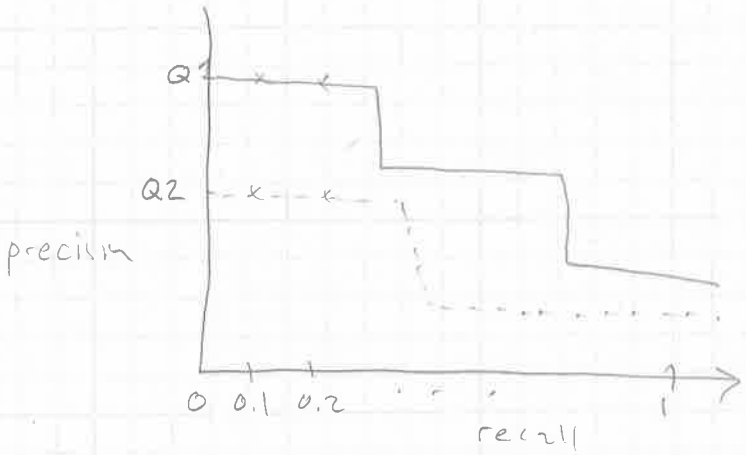
To compute one PR curve for a set of queries.

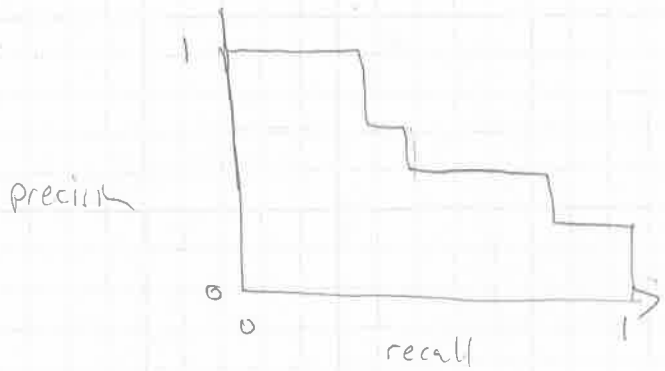
1) Pick a set of reference recall values.

   0, 0.1, ..., 0.9, 1 or finer

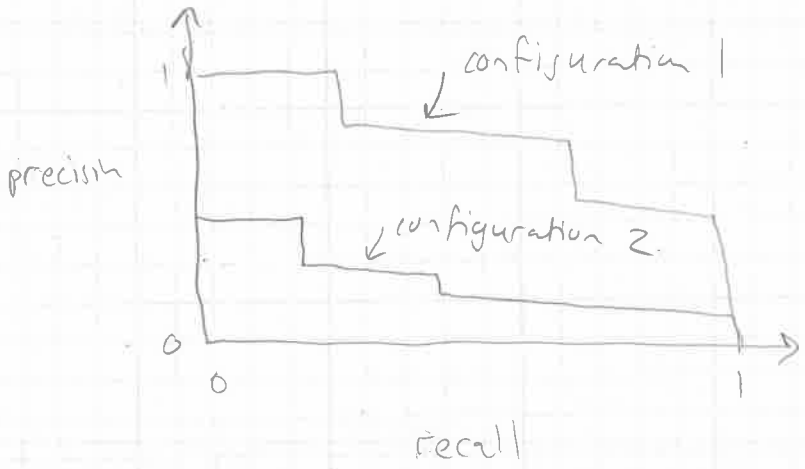
2) For each query image, compute precision at each reference recall value using interpolation if necessary.

3) Average these precision values over queries to derive a single PR curve.

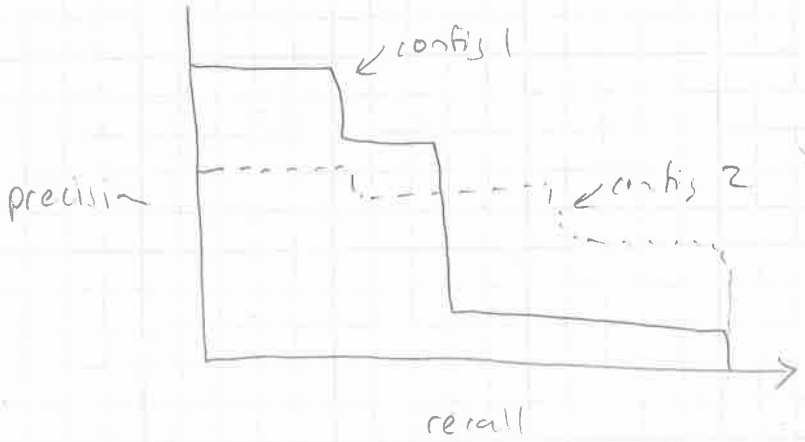




We can now compute a single PR curve for each design choice (features, distance measure, etc.) and plot them simultaneously



Which is better? → configuration 1



Which is better? Not clear

Can compute and compare the average precision over recall values as a way to compare different configurations

$$\text{Average precision} = \frac{1}{j}\sum_{i=1}^{j} p(r_i)$$

where $r_i = 0, 0.1, 0.2, \ldots, 0.9, 1$ for example

This is the same thing as area under the PR curve. Higher average precision indicates better performance overall.