Locality-Sensitive Hashing

Focusing on Similar Minhash Signatures
Other Applications Will Follow

Locality-Sensitive Hashing

- General idea: Generate from the collection of all elements (signatures in our example) a small list of *candidate pairs*: pairs of elements whose similarity must be evaluated.
- For signature matrices: Hash columns to many buckets, and make elements of the same bucket candidate pairs.

这样做的缺点是当文档数量很多时，要比较的次数会非常大。那么我们可不可以只比较那些相似度可能会很高的文档，而直接忽略过那些相似度很低的文档。

Candidate Generation From Minhash Signatures

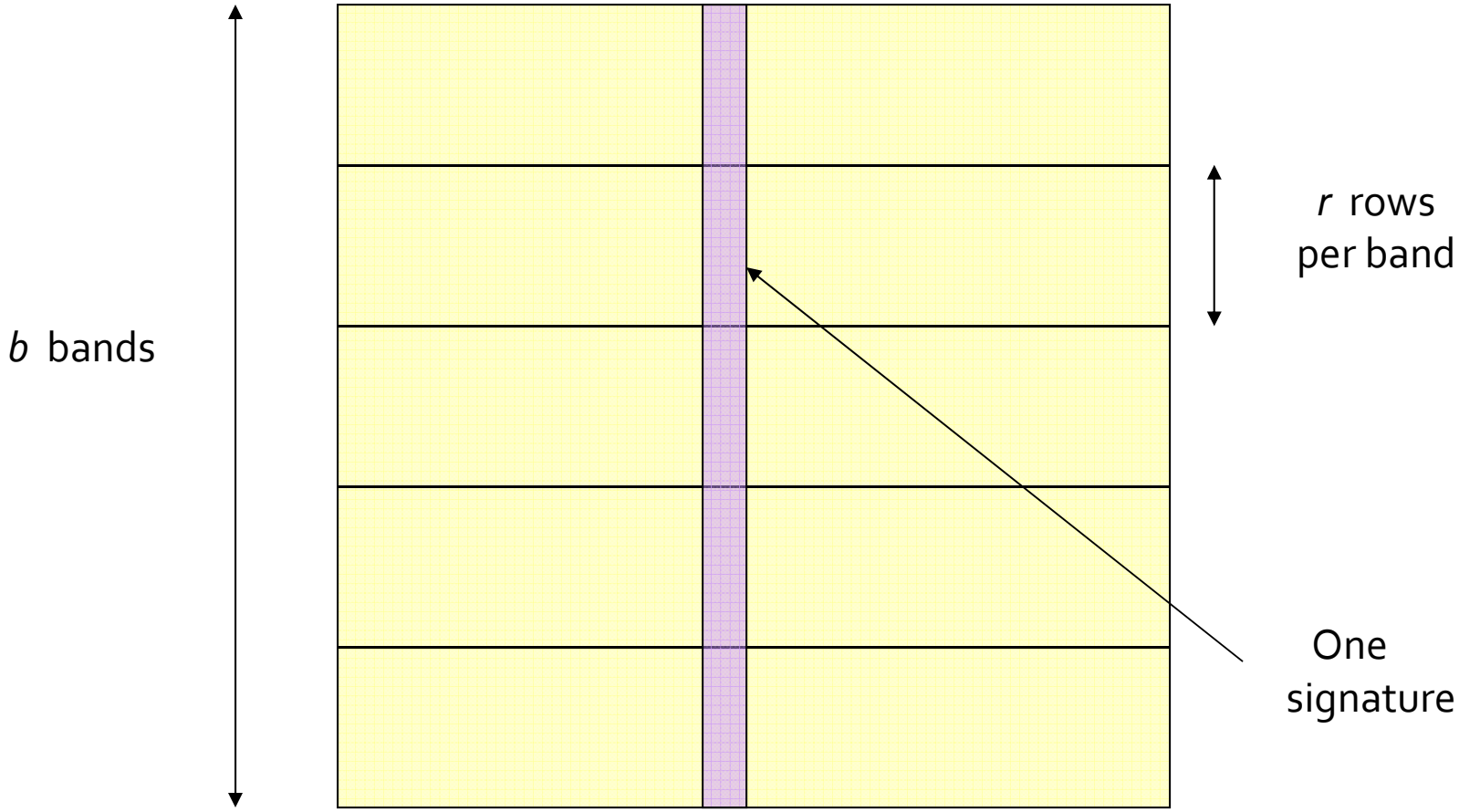
https:// https://blog.csdn.net/liujan511536/article/details/47729721

- Pick a similarity threshold *t*, a fraction < 1.
- We want a pair of columns *c* and *d* of the signature matrix M to be a *candidate pair* if and only if their signatures agree in at least fraction *t* of the rows.
  - I.e., $M(i, c) = M(i, d)$ for at least fraction *t* values of *i*.

LSH for Minhash Signatures

- Big idea: hash columns of signature matrix *M* several times.
- Arrange that (only) similar columns are likely to hash to the same bucket.
- Candidate pairs are those that hash at least once to the same bucket.

Partition Into Bands

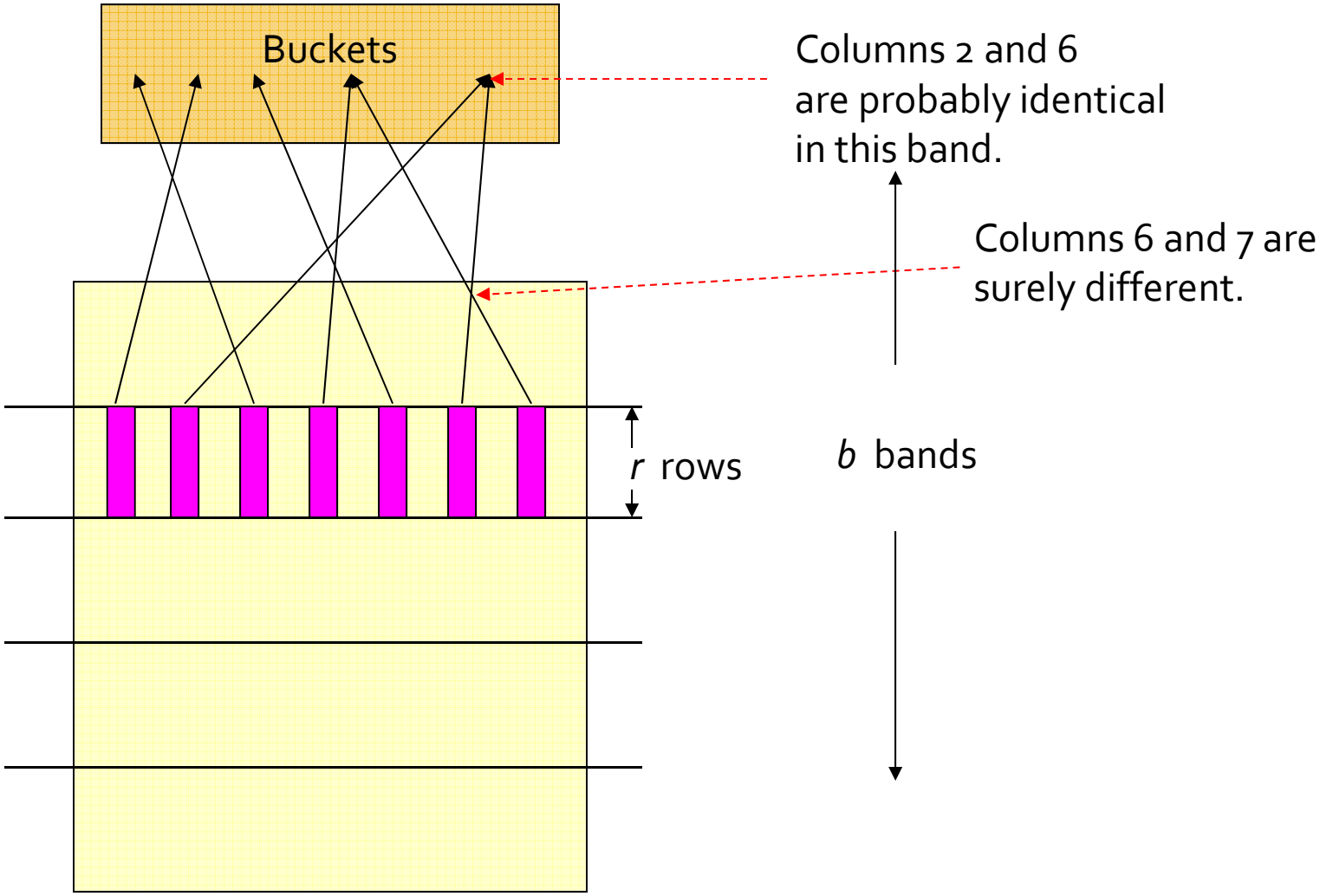


Partition into Bands – (2)

- Divide matrix M into b bands of r rows.
- For each band, hash its portion of each column to a hash table with k buckets.
  - Make k as large as possible.
- *Candidate* column pairs are those that hash to the same bucket for ≥ 1 band.
- Tune b and r to catch most similar pairs, but few nonsimilar pairs.

可以对所有行条使用相同的哈希函数，但是对于每个行条我们都使用一个独立的桶数组，因此即便是不同行条中的相同列向量，也不会被哈希到同一个桶中。这样，只要两个集合在某个行条中有落在相同桶的两列，这两个集合就被认为可能相似度比较高，作为后续计算的候选对；而那些在所有行条中都不落在同一个桶中的两列，就会被认为相似度不会很高，而被直接忽略。

Hash Function for One Bucket



Example – Bands

- Suppose 100,000 columns.
- Signatures of 100 integers.
- Therefore, signatures take 40Mb.
- Want all 80%-similar pairs of documents.
- 5,000,000,000 pairs of signatures can take a while to compare.
- Choose 20 bands of 5 integers/band.

4byte 一个

100，000取2

Suppose C_1, C_2 are 80% Similar

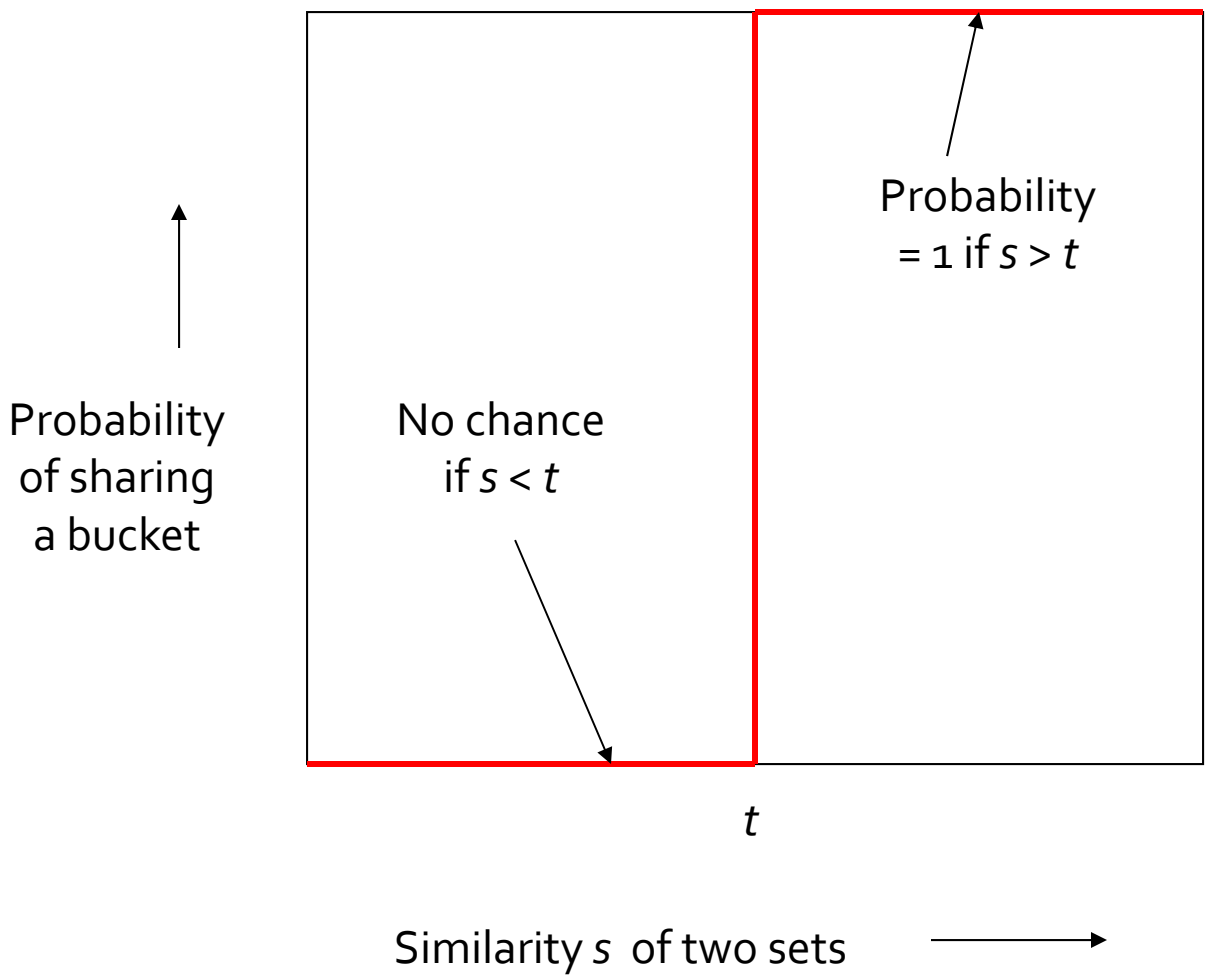
- Probability C_1, C_2 identical in one particular band: $(0.8)^5 = 0.328$.
- Probability C_1, C_2 are *not* similar in any of the 20 bands: $(1\text{-}0.328)^{20} = .00035$.
  - i.e., about 1/3000th of the 80%-similar underlying sets are false negatives.

不是candidate的概率

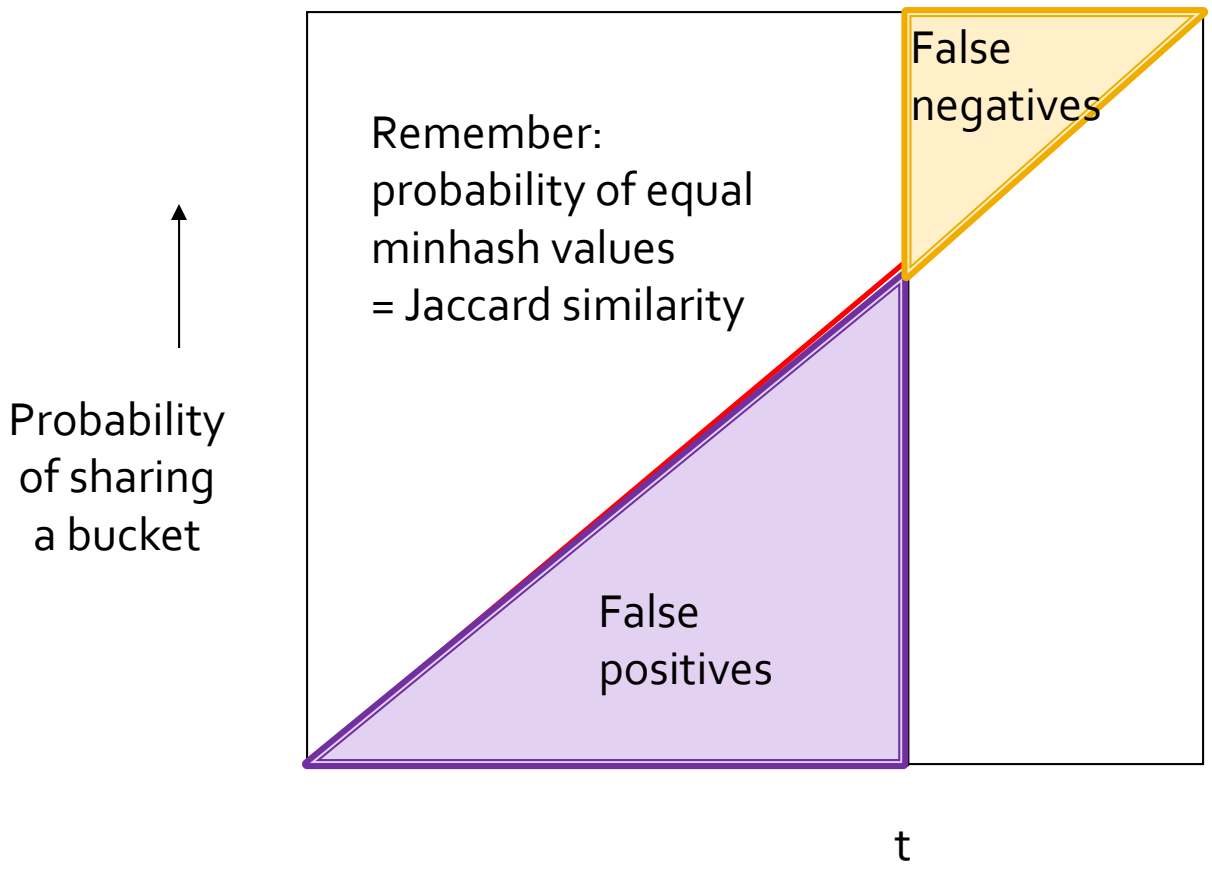
Suppose C_1, C_2 Only 40% Similar

- Probability C_1, C_2 identical in any one particular band: $(0.4)^5 = 0.01$.
- Probability C_1, C_2 identical in $\geq$ 1 of 20 bands: $\leq 20 * 0.01 = 0.2$.
- But false positives much lower for similarities << 40%.

Analysis of LSH – What We Want

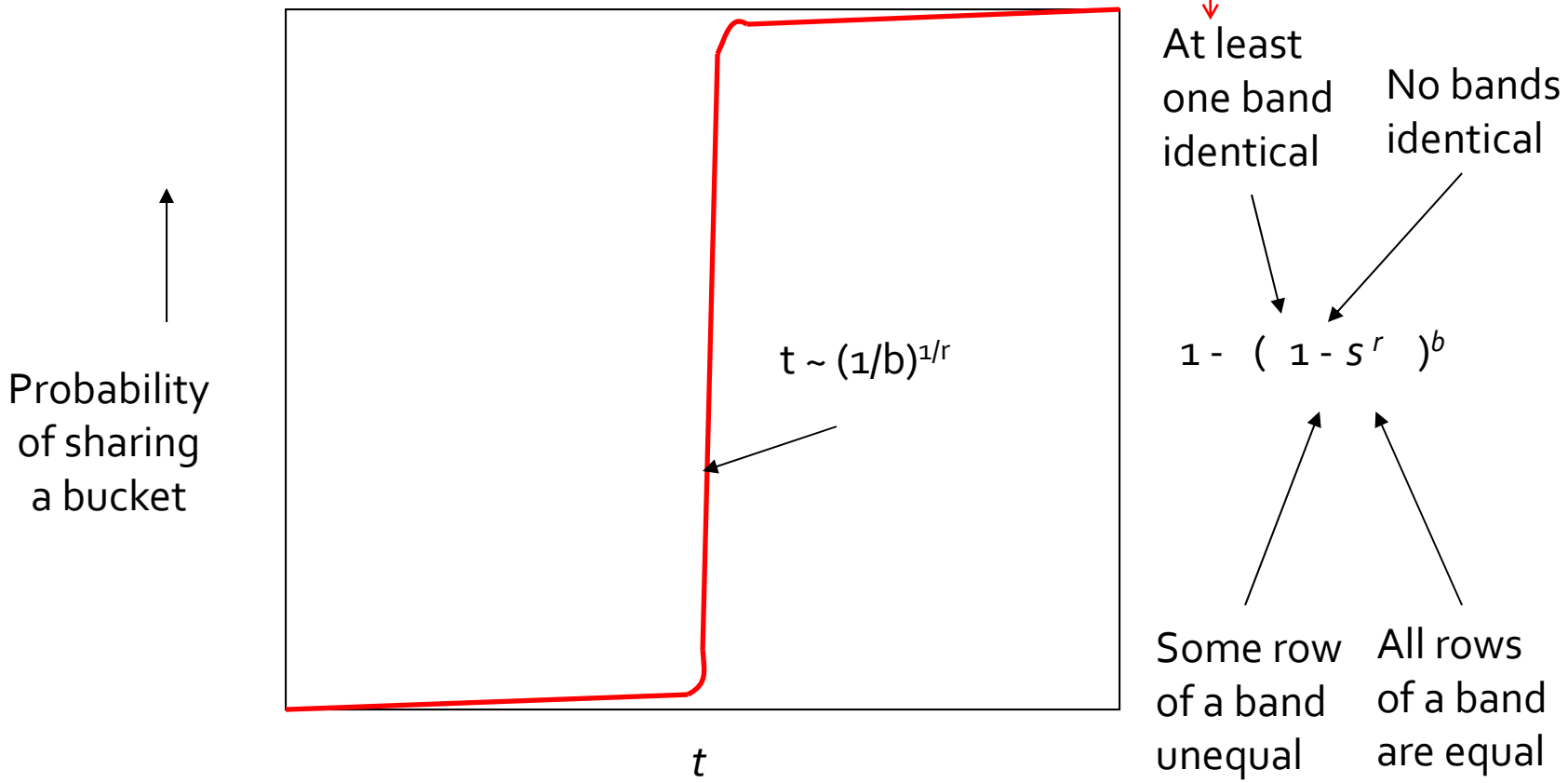


What One Band of One Row Gives You



What *b* Bands of *r* Rows Gives You

至少有一个是一样的，candidate，分到一个类



Example: $b = 20$; $r = 5$

s	$1-(1-s^r)^b$
.2	.006
.3	.047
.4	.186
.5	.470
.6	.802
.7	.975
.8	.9996

LSH Summary

- Tune to get almost all pairs with similar signatures, but eliminate most pairs that do not have similar signatures.
- Check that candidate pairs really do have similar signatures.
- Optional: In another pass through data, check that the remaining candidate pairs really represent similar *sets* .

By computing the Jaccard similarity of the underlying sets,
we can eliminate the false positives.
Unfortunately, we cannot eliminate false negatives this way.