Statistical Hypothesis Tests for NLP

or: Approximate Randomization for Fun and Profit

William Morgan

ruby@cs.stanford.edu

Stanford NLP Group

You have two systems...

Motivation:

- Two systems, A and B.
- They produce output o_A and o_B.
- Want to show that B is better than A.

The simple approach:

- Using some evaluation metric e, show $e(o_B) > e(o_A)$.
- And that's enough...

. . . or is it?

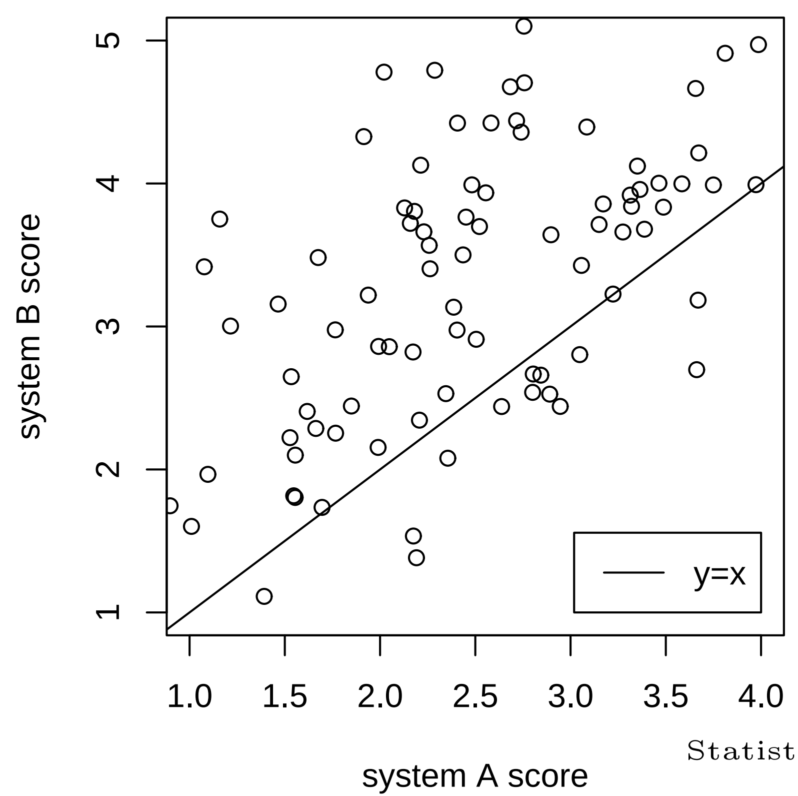
Sadly, no:

- The difference between $e(o_A)$ and $e(o_B)$ could be due to *sheer dumb luck*.
- We want to show that's not the case.
- Statistical significance tests give us a way of quantifying the probability that the difference between two systems is due to luck.
  - If low, we can believe the difference is real.
  - If high, then either:
    1. the systems are not different; or
    2. the data are insufficient to show that the systems aren't different.

Quick question

Why not just compare confidence intervals?

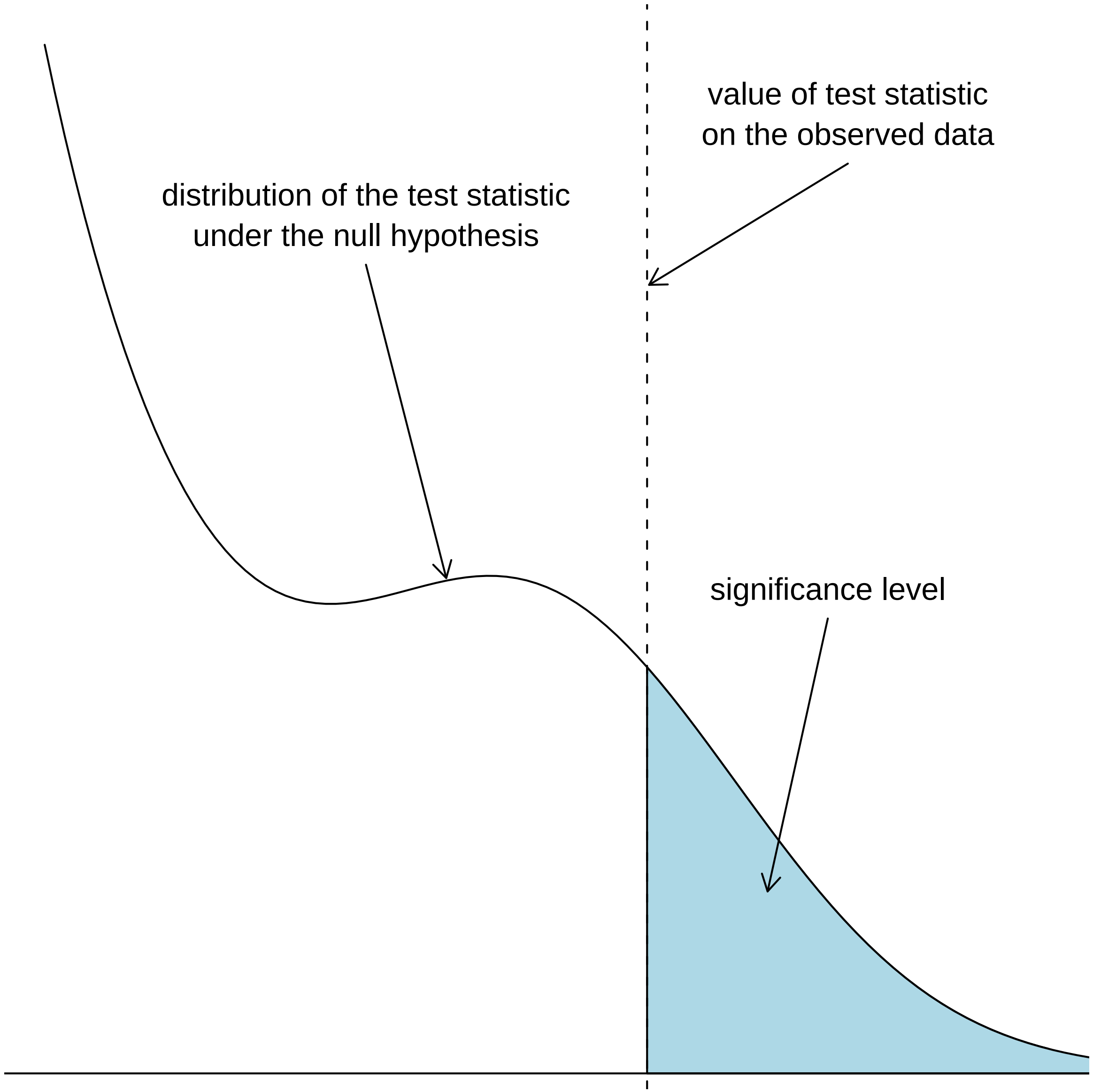
- You sometimes seem people determining statistical significance of system differences by looking at whether the confidence intervals overlap.
- This approach is overly conservative:



Statistical significance tests

1. Forget about bigger or smaller. Let's just think about "difference" or "no difference". (A "two-tailed" test.)
2. Call the hypothesis that there's no difference between A and B the *null hypothesis*.
3. Pretend o_A and o_B are "sampled" independently from a "population" (so o_A and o_B decomposable).
4. Call $t(o_1, o_2) = |e(o_1) - e(o_2)|$ the "test statistic" (so $t : \text{system output} \times \text{system output} \rightarrow \mathbb{R}$).
5. Find the distribution of t under the null hypothesis, i.e. assuming the null hypothesis is true.
6. See where $t(o_A, o_B)$—the thing we actually observed—lies in this distribution.
    - If it's somewhere weird (unlikely), that's evidence that the null hypothesis is false, i.e. the systems *are* different.

Pretty picture



The significance level

- The area to the right of $t(o_A, o_B)$ is the "significance level"—the probability that some $t^* \geq t(o_A, o_B)$ would be generated *if the null hypothesis were true*.
  - Also called the p-value.
- Small values suggest the null hypothesis is false, given the observation of $t(o_A, o_B)$.
- Corollary: all else being equal, a large difference between $e(o_A)$ and $e(o_B)$ yields a smaller significance level (as one would hope!).
- Values below 0.05 are typically considered "good enough."

So all we have to do is calculate the distribution of t.

Calculating the distribution

The classical approach:

- Keep adding assumptions until we arrive at a known distribution which we can calculate analytically.
- E.g.: Student's t-test.
  - Assume that $e(o_A)$ and $e(o_B)$ are sample means from a bivariate Normal distribution with zero covariance. Then we know t is distributed according to Student's t-distribution if the null hypothesis is true.
- Back in the stone age, computing with rocks and twigs, making those assumptions made the problem tractable.
- But the problem with this approach is that you may falsely reject the null hypothesis if one of the additional assumptions is violated. (Type I error.)

What you SHOULD do

- Simulate the distribution using a *randomization* test.
- It's just as good as analytical approaches, even when the analytical assumptions are met! (Hoeffding 1952)
- And it's better when they're not. (Noreen 1989)
- Best of all: dirt simple.

Intuition:

- Erase the labels "output of A" or "output of B" from all of the observations.
- Now consider the population of every possible labeling. (Order relevant.)
- If the systems are really different, the observed labeling should be unlikely under this distribution.

Basic approximate randomization

- "Exact" randomization requires iterating through the entire set of possible labelings. (E.g. Fisher's exact test.)
- That's huge! Instead, sample from it.
- Let $o_A = \{o_A^1, \cdots, o_A^n\}$ and $o_B = \{o_B^1, \cdots, o_B^m\}$ be the output of the two systems.
- Repeat R times: randomly assign each of $\{o_A^1, \cdots, o_A^n, o_B^1, \cdots, o_B^m\}$ into classes X (size n) and Y (size m). Calculate $t(X, Y)$.
- Let r be the number of times that $t(X, Y) \geq t(o_A, o_B)$.
- As $R \to \infty$, r/R approaches the significance level.
  - Actually, should use $\frac{r+1}{R+1}$ for "statistical reasons" (not that it matters for, say, $R \geq 19$)

Some comments

- That was easy.
- Random assignment is done *without* replacement, in contrast to bootstrap.
- Randomization tests are statistically “valid”, meaning the probability of falsely rejecting the null hypothesis is no greater than the rejection level of the test (i.e. choosing a threshold for the significance level a priori).
  - That’s important!
  - (and where those $+1$’s come from.)
- $R = 1000$ is the typical case.

Now how do we use that for NLP applications?

Applying this to NLP

- Our output bits $o_A^1, \cdots, o_A^n, o_B^1, \cdots, o_B^m$ are going to be something like tagged/translated/parsed sentences.
- Our metric function e is going to be something like BLEU/WER/F-measure.
- We'll run two systems *on the same input* and see how their output differs.
- Wait! Now we've violated an assumption of all s.s.t.'s—that each bit of output is independently sampled:
  - Each o_A^i and o_B^i pair are dependent. (Output of system A on input i will probably be similar to output of system B on input i.)
- A statistician would recognize this situation as requiring a "paired" test.

Approximate randomization for NLP

- We can control for dependent variables by *stratifying* the output and only permuting within each stratum. (Yeh 2000, Noreen 1989)
- In this case, we'll stratify each o_A^i, o_B^i.
- Let $o_A = \{o_A^1, \cdots, o_A^n\}$ and $o_B = \{o_B^1, \cdots, o_B^n\}$ be the output of the two systems *on the same input*.
- Start with $X = o_A$ and $Y = o_B$.
- Repeat R times: randomly flip each o_A^i, o_B^j between X and Y with probability $\frac{1}{2}$. Calculate $t(X, Y)$.
- Let r be the number of times that $t(X, Y) \geq t(o_A, o_B)$.
- As $R \to \infty$, $\frac{r+1}{R+1}$ approaches the significance level.

Randomization vs. Bootstrap

Q: How do randomization tests compare with bootstrap resampling, in which data is drawn *with* replacement?

- For example, Koehn (2004) proposes "paired bootstrap resampling" for MT comparisons, which is almost identical to AR except for the replacement issue.

A: Bootstrap resampling contains an additional assumption, which is that the (original) sample is close to the population of all possible outputs.

- Randomization tests do not require this assumption and thus are better.
- Riezler and Maxwell (2005) also give anecdotal evidence that bootstrap resampling is more prone to type I errors than AR for SMT.

Comparing many systems

- So that's how we compare two systems.
- If we compare many systems, there's a danger we need to be aware of.
- In the binary comparison case, with threshold 0.05, validity tells us that we'll falsely reject the null hypothesis (make a type I error) 5% of the time.
- But if we do 20 comparisons, the chance of making a type I error can be as high as $1 - (1 - 0.05)^{20} = .64$.
- How do we prevent this?

Correcting for Multiple Tests

- The *Bonferonni correction* is the most well-known solution: simply divide the threshold by n. In the above case, $1 - (1 - \frac{0.05}{20})^{20} = 0.04884 \approx 0.05$.
  - But Bonferonni is widely considered overly conservative (i.e. sacrifices Type II error control for Type I) and not often used in practice.
- Another popular option is Fisher's Least Significant Difference (LSD) test. (But possibly too liberal.)
- Or, consider Tukey's Honestly Significant Difference (HSD) test. (But possibly too conservative.)

Which one should I use?

- Probably none of them.
- Only indisputably called for when:
  1. you're doing post-hoc (unplanned) comparisons; or
  2. you have a "global" null hypothesis ("if any one of these components is different from the others, then we've improved the state of the art").
- In other cases, you probably have sufficient philosophical currency to do nothing at all.
- But you should be aware of the issue, so you know what to say when Bob Moore yells at you at ACL.

Summary

- Use approximate randomization to compare two systems.
- Calculate confidence intervals, but don't read anything into overlap.
- If you're comparing lots of things, think about (but don't use) some form of correction.

References

- Wassily Hoeffding. 1952. The Large-Sample Power of Tests Based on Permutations of Observations. Annals of Mathematical Statistics, 23, 169–192.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation. Proceedings of EMNLP.
- Eric W. Noreen. 1989. Computer Intensive Methods for Testing Hypothesis. John Wiley & Sons.
- Stefan Rielzer and John T. Maxwell III. 2005. On Some Pitfalls in Automatic Evaluation and Significance Testing in MT. Proceedings of the ACL Workshop in Intrinsic and Extrinsic Evaluation Measures for MT.
- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. Proceedings of Coling 2000.