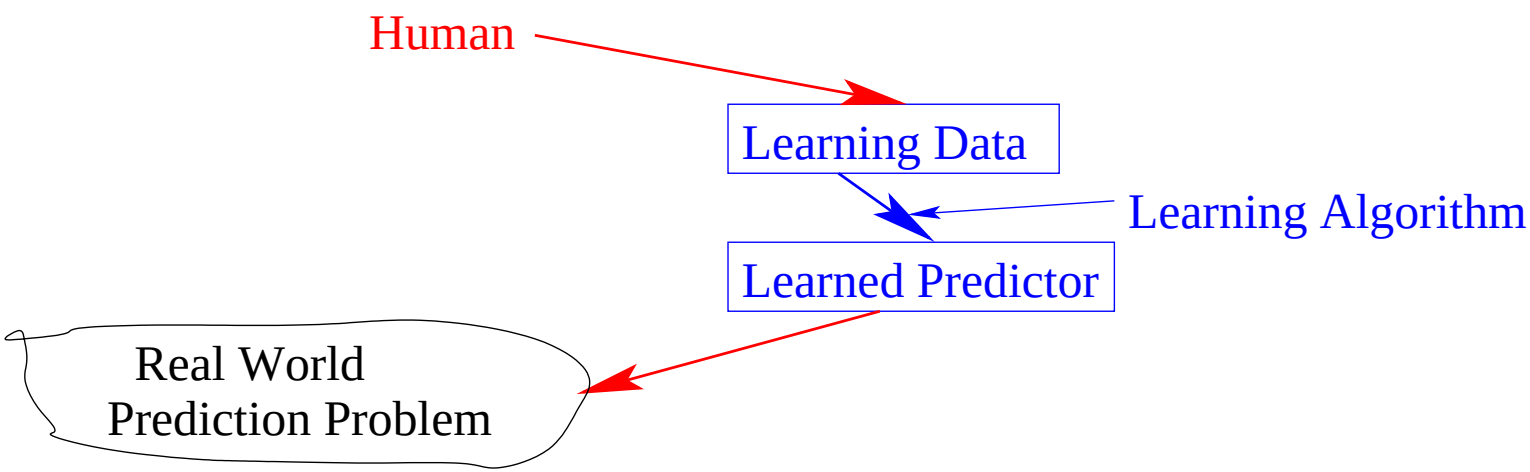
What is Learning Problem Design?

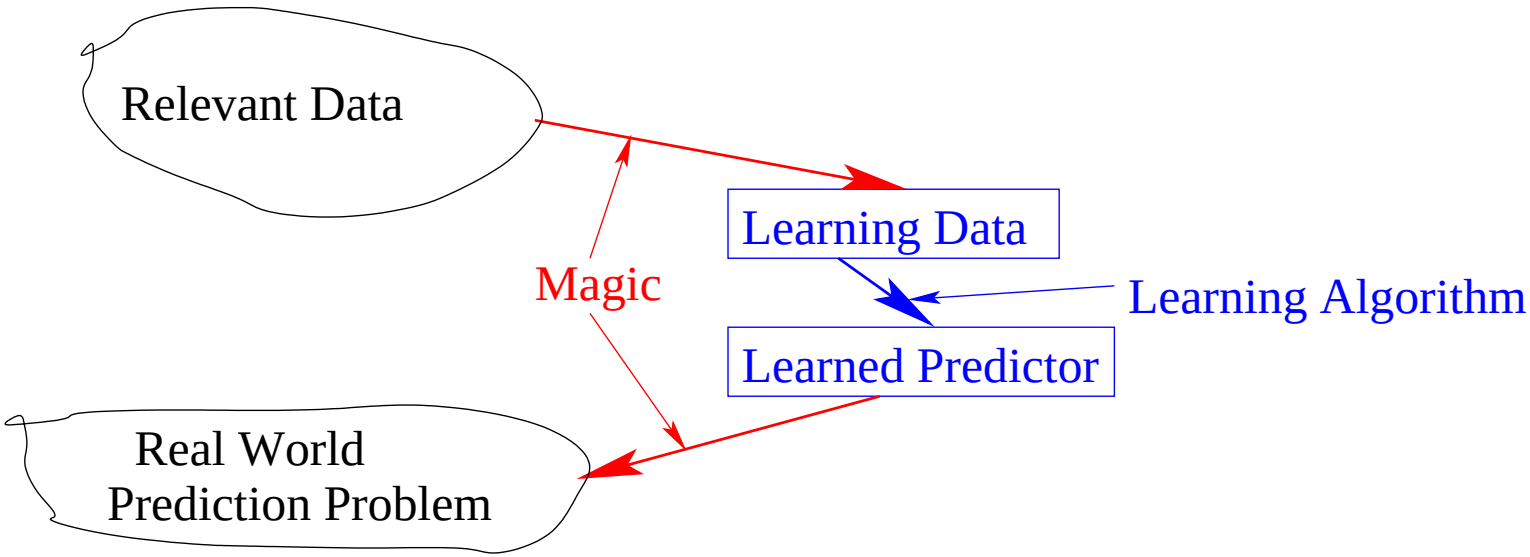
Learning Problem Design Workshop, NIPS 2007

John Langford

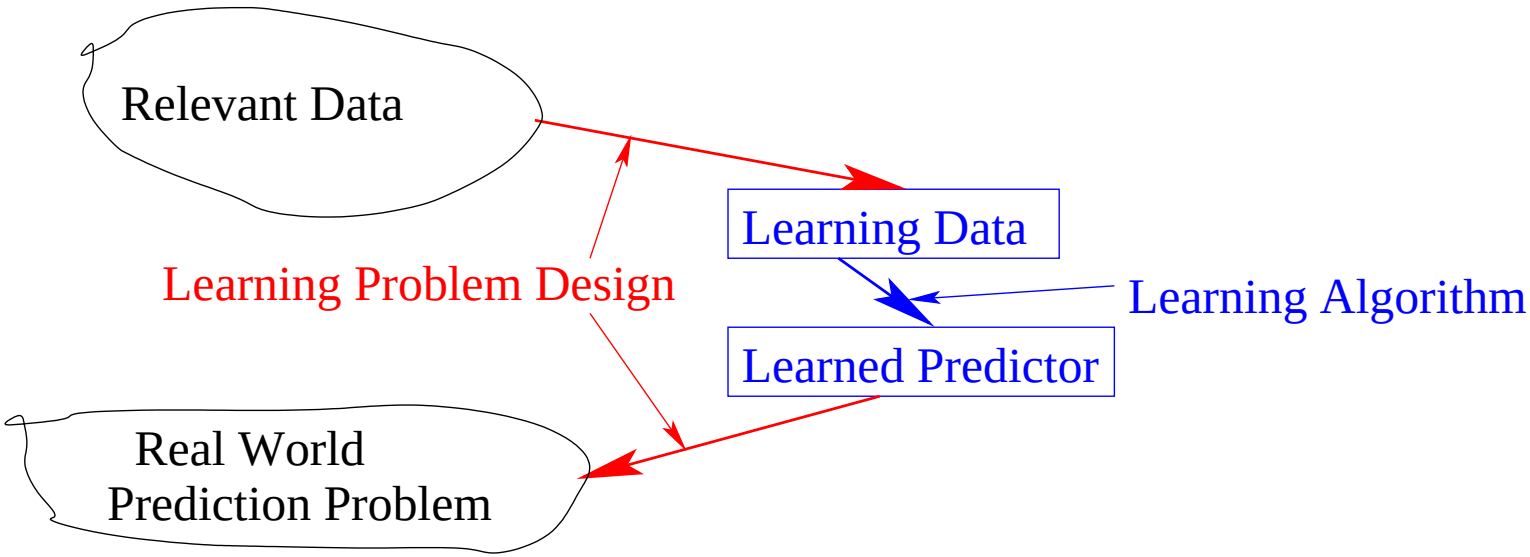
How Learning Works



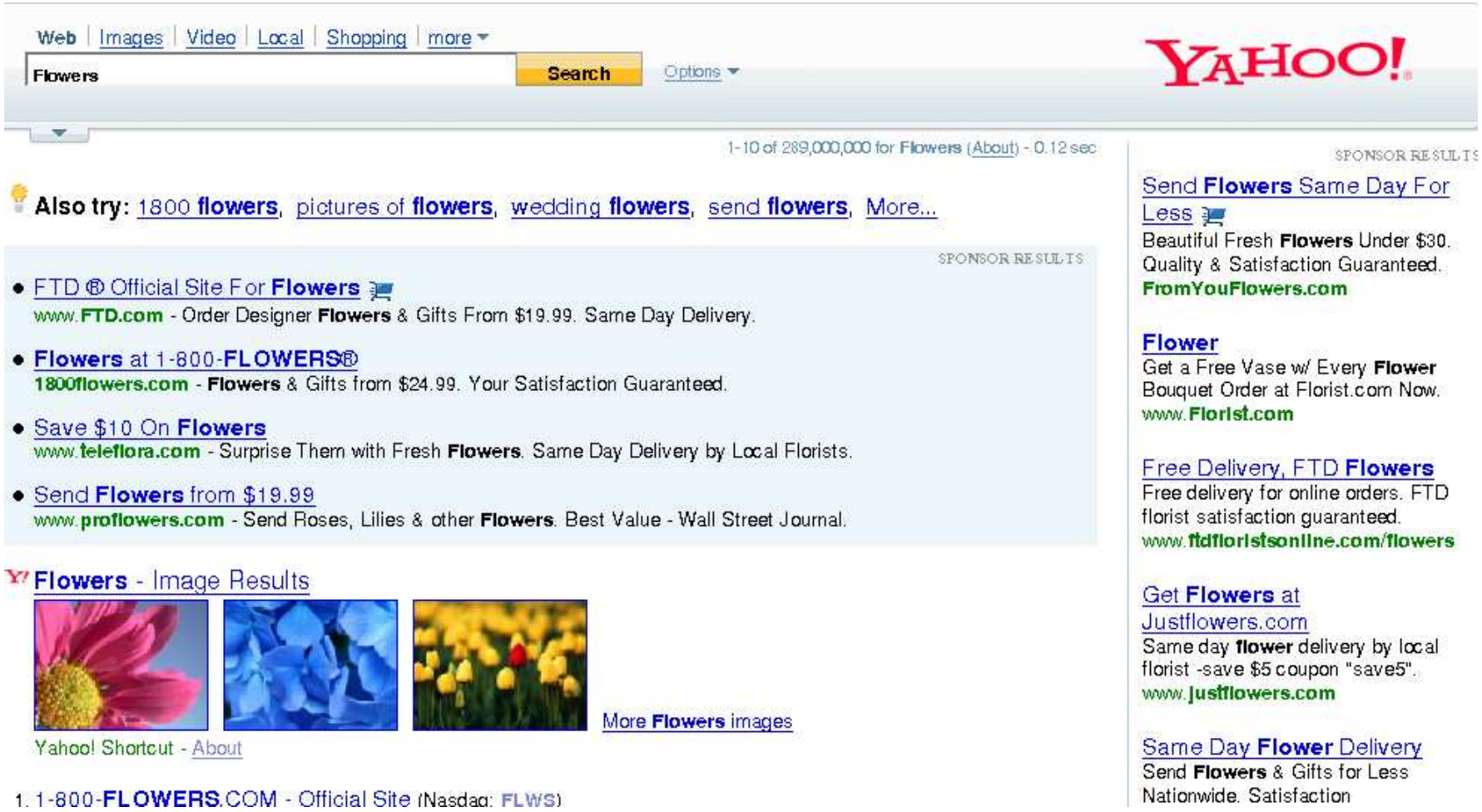
How Learning Sometimes Works



How we would like learning to Work



Example: Ad Display Policy



You might think this is a ranking or multiclass prediction problem, but...

It's *not* supervised learning. You do not know what would have happened in the past if different ads had been displayed $\Rightarrow$ you can't evaluate a new policy given data from an arbitrary old policy.

1. What should you predict?

2. And how do you use it to solve the real problem?

Example: Spam Filtration

This looks like a standard binary classification problem but...

1. There is a shared sense of what spam is, but it's not completely shared. Users are sometimes even adversarial.

2. What is the label of an email? The act of shifting it into a folder? Or the act of leaving it in a folder? Or something else?

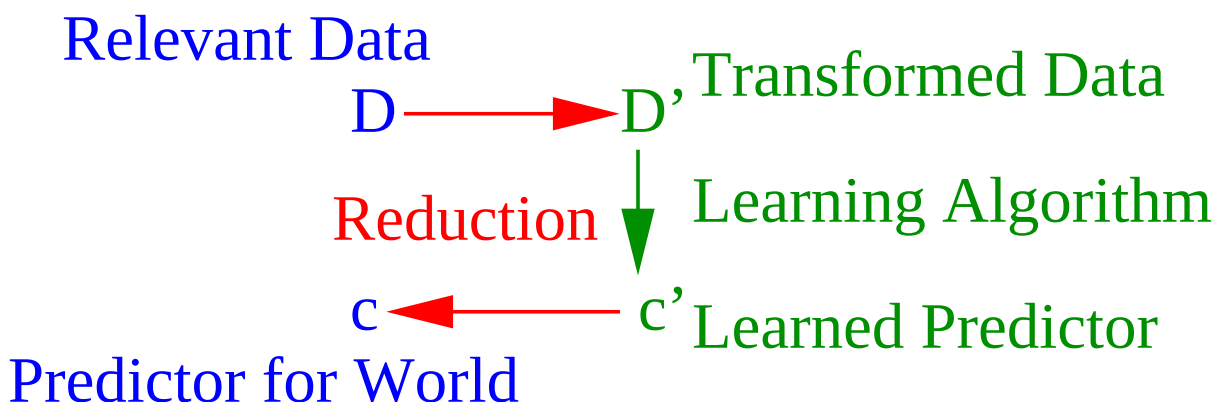
Questions We ask for Learning Problem Design

1. What are relevant data sources?

2. What are good methods for using relevant data sources?

3. How can we best design learning problems taking advantage of relevant data?

The Role of Reductions



Optimize Reduction so that

$$\text{Regret}(c, D) \equiv \text{Loss}(c, D) - \min_h \text{Loss}(h, D)$$

tightly bounded by

$$\text{Regret}(c', D') \equiv \text{Err}(c', D') - \min_{h'} \text{Err}(h', D')$$

Why Reductions are not a complete answer

1. It isn't clear we can find a reduction for every relevant data source.

2. Reductions assume away the difficulty of the learning step.