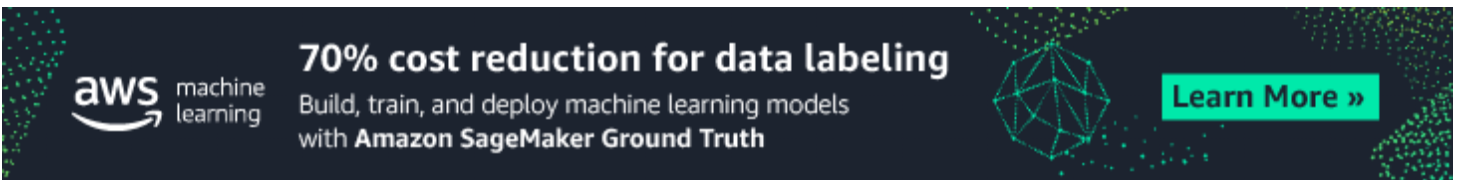


Generative vs. discriminative

Asked 8 years, 8 months ago Active 1 month ago Viewed 71k times



I know that generative means "based on $P(x, y)$ " and discriminative means "based on $P(y|x)$," but I'm confused on several points:

159

- Wikipedia (+ many other hits on the web) classify things like SVMs and decision trees as being discriminative. But these don't even have probabilistic interpretations. What does discriminative mean here? Has discriminative just come to mean anything that isn't generative?
- Naive Bayes (NB) is generative because it captures $P(x|y)$ and $P(y)$, and thus you have $P(x, y)$ (as well as $P(y|x)$). Isn't it trivial to make, say, logistic regression (the poster boy of discriminative models) "generative" by simply computing $P(x)$ in a similar fashion (same independence assumption as NB, such that $P(x) = P(x_0)P(x_1) \dots P(x_d)$, where the MLE for $P(x_i)$ are just frequencies)?
- I know that discriminative models tend to outperform generative ones. What's the practical use of working with generative models? Being able to generate/simulate data is cited, but when does this come up? I personally only have experience with regression, classification, collab. filtering over structured data, so are the uses irrelevant to me here? The "missing data" argument ($P(x_i|y)$ for missing x_i) seems to only give you an edge with training data (when you actually know y and don't need to marginalize over $P(y)$ to get the relatively dumb $P(x_i)$ which you could've estimated directly anyway), and even then imputation is much more flexible (can predict based not just on y but other x_i 's as well).
- What's with the completely contradictory quotes from Wikipedia? "generative models are typically more flexible than discriminative models in expressing dependencies in complex learning tasks" vs. "discriminative models can generally express more complex relationships between the observed and target variables"

[Related question](#) that got me thinking about this.

machine-learning

generative-models

edited Apr 13 '17 at 12:44



Community ♦

1

asked Jun 27 '11 at 20:40



Yang

2,521

3

17

18

5 (+1) I think the terminology is confusing, and I think of a model of $P(y|x)$ as being generative for the

conditional distribution in contrast to SVM, say, that only models aspects of this conditional distribution for a purely discriminative purpose. As your second bullet point shows, a model of $P(y|x)$ can be augmented to become a fully generative model. The distinction is relevant from a learning point of view because we can either model $P(x, y)$ and compute $P(y|x)$ or we can model $P(y|x)$. – NRH Jun 28 '11 at 6:34

- 2 I don't think that your statement about SVMs is true: "But these don't even have probabilistic interpretations." Anything with a loss function can clearly be interpreted in a probabilistic sense by finding the MAP configuration of a suitable PDF. – gmatt Oct 8 '15 at 0:49 ✎

3 Answers

The **fundamental difference** between discriminative models and generative models is:

- **Discriminative models** learn the (hard or soft) **boundary** between classes
- **Generative models** model the **distribution** of individual classes

To answer your direct questions:

- SVMs and decision trees are discriminative because they learn explicit boundaries between classes. SVM is a maximal margin classifier, meaning that it learns a decision boundary that maximizes the distance between samples of the two classes, given a kernel. The distance between a sample and the learned decision boundary can be used to make the SVM a "soft" classifier. DTs learn the decision boundary by recursively partitioning the space in a manner that maximizes the information gain (or another criterion).
- It is possible to make a generative form of logistic regression in this manner. Note that you are not using the full generative model to make classification decisions, though.
- There are a number of advantages generative models may offer, depending on the application. Say you are dealing with non-stationary distributions, where the online test data may be generated by different underlying distributions than the training data. It is typically more straightforward to detect distribution changes and update a generative model accordingly than do this for a decision boundary in an SVM, especially if the online updates need to be unsupervised. Discriminative models also do not generally function for outlier detection, though generative models generally do. What's best for a specific application should, of course, be evaluated based on the application.
- (This quote is convoluted, but this is what I think it's trying to say) Generative models are typically specified as probabilistic graphical models, which offer rich representations of the independence relations in the dataset. Discriminative models do not offer such clear representations of relations between features and classes in the dataset. Instead of using resources to fully model each class, they focus on richly modeling the boundary between classes. Given the same amount of capacity (say, bits in a computer program executing the model), a discriminative model thus may yield more complex representations of this boundary than a generative model.

answered Jun 27 '11 at 22:40



benhamner

2,453 1 16 14

A trivial comment: $\binom{k}{2}$ boundaries between k clusters get harder and harder to plot / understand / measure as k increases. Thus e.g. k -clustering, viewed as a discriminative model, gets noisier and ad hoc. (Example:

Building an ML Anomaly Detection System?

Get The Ultimate Guide

anodot

(hamner's answer is great, so just cross-posting my answer from MetaOptimize for completeness.)

69

I think of generative algorithms as providing a **model of how the data is actually generated** (I think of them as giving you a model of both $P(X|Y)$ and $P(Y)$, rather than of $P(X, Y)$, though I guess it's equivalent), and discriminative algorithms as simply providing **classification splits** (and not necessarily in a probabilistic manner).



Compare, for instance, Gaussian mixture models and k-mean clustering. In the former, we have a nice probabilistic model for how points are generated (pick a component with some probability, and then emit a point by sampling from the component's Gaussian distribution), but there's nothing we can really say about the latter.

Note that generative algorithms have discriminative properties, since you can get $P(Y|X)$ once you have $P(X|Y)$ and $P(Y)$ (by Bayes' Theorem), though discriminative algorithms don't really have generative properties.

1: Discriminative algorithms allow you to classify points, without providing a model of how the points are actually generated. So these could be either:

- **probabilistic** algorithms try to learn $P(Y|X)$ (e.g., logistic regression);
- or **non-probabilistic** algorithms that try to learn the mappings directly from the points to the classes (e.g., perceptron and SVMs simply give you a separating hyperplane, but no model of generating new points).

So yes, discriminative classifiers are any classifiers that aren't generative.

Another way of thinking about this is that **generative algorithms make some kind of structure assumptions on your model**, but discriminative algorithms make fewer assumptions. For example, Naive Bayes assumes conditional independence of your features, while logistic regression (the discriminative "counterpart" of Naive Bayes) does not.

2: Yes, Naive Bayes is generative because it captures $P(X|Y)$ and $P(Y)$. For example, if we know that $P(Y = \text{English}) = 0.7$ and $P(Y = \text{French}) = 0.3$, along with English and French word probabilities, then we can now generate a new document by first choosing the language of the document (English with probability 0.7, French with probability 0.3), and then generating words according to the chosen language's word probabilities.

Yes, I guess you could make logistic regression generative in that fashion, but it's only because you're adding something to logistic regression that's not already there. That is, when you're performing a Naive Bayes classification, you're directly computing $P(Y|X) \propto P(X|Y)P(Y)$ (the terms on the right, $P(X|Y)$ and $P(Y)$, are what allow you to generate a new document); but when you're computing $P(Y|X)$ in logistic regression, you're not computing these two things, you're just applying a logistic function to a dot product.

3: Generative models often outperform discriminative models on smaller datasets because their **generative assumptions place some structure on your model that prevent overfitting**. For example, let's consider Naive Bayes vs. Logistic Regression. The Naive Bayes assumption is of course rarely satisfied, so logistic regression will tend to outperform Naive Bayes as your dataset grows (since it can capture dependencies that Naive Bayes can't). But when you only have a small data set, logistic regression might pick up on spurious patterns that don't really exist, so the Naive Bayes acts as a kind of regularizer on your model that prevents overfitting. There's a [paper](#) by Andrew Ng and Michael Jordan on discriminative vs. generative classifiers that talks about this more.

4: I think what it means is that generative models can actually learn the underlying structure of the data if you specify your model correctly and the model actually holds, but discriminative models can outperform in case your generative assumptions are not satisfied (since discriminative algorithms are less tied to a particular structure, and the real world is messy and assumptions are rarely perfectly satisfied anyways). (I'd probably just ignore these quotes if they're confusing.)

edited Jan 23 '17 at 1:33



PALEN

133 5

answered Jun 28 '11 at 21:43



raegtin

8,050 10 44 50



1



As an additional point to the above answers, when the aim of algorithm is only to classify, then discriminative approach may be better and less expensive than generative approach, as following the generative approach to model input distribution can result in requiring too much training data to model complexities in distribution which are unimportant for computing the posteriors required for decision making.



answered Jan 11 at 15:26



HARLEEN HANSPAL

11 1