Generative and Discriminative Models

Alan Yuille

October 9, 2021

Discriminate Methods versus Generative Modeling

- For the last ten years most of computer vision research has been driven by discriminative methods (like Deep Networks). This has lead to huge improvements as evaluated by standard performance measures on large annotated datasets.
- But there are clear limits to the current approaches. When Deep Networks were introduced performance improved by 10AP for some visual tasks. But now improvements are much smaller (despite orders of magnitude more researchers and computer power).

Discriminate Methods versus Generative Modeling

- Fundamental Problem. The datasets are not nearly big enough. Too few visual tasks are being addressed (because of the need for annotation). Much tougher testing is required (e.g., out-of-distribution testing). The computer vision community is stuck in a local minimum.
- High Tech companies are aware of this problem. They have gigantic datasets (much bigger than the academic community) so they know that current methods are good, but not good enough, to deal with the complexity of the real world. "Existing methods do not work on big datasets because there are too many corner cases" (Anonymous CEO).

Discriminate Learning: The setup used by Deep Nets

- We first describe the standard procedures for learning a classifier assuming balanced annotated datasets.
- The standard procedure assumes there is an unknown distribution $P(x, y)$ which generates both the training and the testing data. Our task is to find a classifier $y = f(x : \theta)$ from the training data and with a loss function $L(y, f(x, \theta))$.
- We assume training samples $\mathcal{X}_{Train} = \{(x_i, y_i) : i = 1, \ldots, N\}$ and testing samples $\mathcal{X}_{Test}\{(x_a, y_a) : a = 1, \ldots, M\}$. Both are random samples from $P(x, y)$ so it assumed that $\mathcal{X}_{Train}$ and $\mathcal{X}_{Test}$ are similar.

Discriminate Learning: The setup used by Deep Nets

- The classifier is learnt from the training set to determine $\hat{\theta} = \arg\min_\theta \sum_{(x,y)\in\mathcal{X}_{Train}} L(y, f(x,\theta))$. The classifer is tested on the test set $\sum_{(x,y)\in\mathcal{X}_{Test}} L(y, f(x,\hat{\theta}))$.
- If there is sufficient training data, in terms of the complexity of the classifiers, then good performance on the training set will imply good performance on the testing set (must check to avoid overfitting).
- This is a *discriminative approach* because it learns a classifier $y = f(x : \theta)$. If the loss function is the cross entropy loss, then it tries to learn the distribution $P(y|x)$. *This standard approach can fail badly if we perform out-of-distribution testing*, i.e. if the test data is not generated from $P(x, y)$.

Why do People use the Discriminative Approach?

- *Why do people use the discriminative approach?*
- Two main reasons:
- (I) We know how to do discriminative learning. There is a long history of successful discriminative methods with Deep Nets being the most recent (and most effective). These methods are very successful for certain types of visual tasks (given data, GPUs, etc). They are essentially regression methods (using Statistics terminology, where regression includes continuous and discrete variables).
- (II) The current gold standard for evaluating computer vision (and other machine learning algorithms) is based on finite-sized balanced annotated datasets, like ImageNet and Coco. Papers are grant proposals are accepted based on algorithms performance on these types of datasets. Dsicriminative/regression methods are well suited to this type of task.

Why do People use the Discriminative Approach?

- There is an alternative approach – Generative/Bayesian – which formulates vision as *analysis by synthesis*. Synthesis means that the generative models are trained to generate images from the underlying visual scene. Analysis means that we interpret images by finding the visual scene which is most likely to generate the observed images. (A modern formulation is inverse computer graphics).
- But analysis by synthesis is much harder to do than discriminative models. There are good algorithms for generating images (Computer Graphics and Style-GANs), which are promising but not yet good enough. But, even more challengingly, we need algorithms which can invert the generative process to analyze the images.
- This talk will describe the advantages of the Generative/Bayesian approach compared to Discriminative methods. This will use approximate generative models. It can be thought of as approximate analysis by synthesis.

What are the limitations of the discriminative approach? (I)

- *What are the limitations of the discriminative approach?*
- The datasets are not big enough. And may never be big enough.
- Consider object classification on ImageNet, This is one of the big success stories of computer vision. On this dataset, computer vision algorithms seem to outperform human observers. But this is not true if you study performance closely.

What are the limitations of the discriminative approach? (I)

- The main limitation is that the datasets are not large enough to be representative of the complexity of the real world. It is easy to show that a sofa detector trained on ImageNet fails to classify sofa's which are seen from viewpoints unrepresented in the training data. There is also context bias (e.g., in ImageNet the only objects in trees are birds). There are *rare events*, also known as *corner cases*, which are underepresented in the dataset (both in the training and testing).
- This means that $\mathcal{X}_{\text{Train}}$ and $\mathcal{X}_{\text{Test}}$ are *biased samples* from the real world distribution $P(x, y)$ so performance results on these datasets mail fail to generalize to other data that is sampled from $P(\vec{x}, y)$.

What are the limitations of the discriminative approach? (II)

- To make this more precise, suppose $y \in \mathcal{Y}$ (e.g., the set of object classes). For each object class y, we define $\mathcal{X}_y$ to be the set of all images in the real world that correspond to object class y. By contrast, our image examples in our training and testing datasets are $\mathcal{X}_{y,\mathrm{Train}}$ and $\mathcal{X}_{y,\mathrm{Test}}$. We assume that the training and testing datasets are balanced, which means that $\mathcal{X}_{y,\mathrm{Train}}$ and $\mathcal{X}_{y\mathrm{Test}}$ are similar (technically this means that these distributions are samples from the same distribution, but it may not be the real world distribution $P(x, y)$.).
- One simple type of Dataset bias arises if there is a large subregion $\mathcal{X}_{y_{\mathrm{bias}}} \in \mathcal{X}_y$ which does not overlap with $\mathcal{X}_{y,\mathrm{Train}}$ (and hence also from $\mathcal{X}_{y,\mathrm{Test}}$ because the training and test dataset is balanced). In this case, discriminative methods can give bad results if it is given images from $\mathcal{X}_{y_{\mathrm{bias}}}$. In practice, there are "missing subregions" for each object category, i.e. $\{\mathcal{X}_{y,\mathrm{bias}} : y \in \mathcal{Y}\}$.

What are the limitations of the discriminative approach? (II)

- This relates to the domain transfer problem. In this case, we have two different domains $\mathcal{X}_{1,\mathrm{Train}}, \mathcal{X}_{1,\mathrm{Test}}$ and $\mathcal{X}_{2,\mathrm{Train}}, \mathcal{X}_{2,\mathrm{Test}}$. But these are different so that algorithms trained on the first domain may not perform well on the second domain (and vice versa).
- Other biases can occur if the number of training examples $|\mathcal{X}_{y,\mathrm{train}}|$ differs between the object categories. They may be a lots of training data for some object classes y (i.e. $|\mathcal{X}_{y,\mathrm{train}}|$ is large) while for others classes the amount of training data is much less.

What are the limitations of the discriminative approach? (III)

- To get more intuitive understanding, realize that images are generated from the underlying three-dimensional world/environment. The image of an object is a function of different *environmental factors*: (i) geometric factors S, V (the shape of the object and the viewpoint), (ii) the T, M texture/material properties of the object, and (iii) the lighting L. (There are other factors, like occlusion or weather conditions, which we will discuss later). Formally $I = F(S, V, T, M, L)$.
- Dataset biases arise if objects are only seen from a limited range of these factors. E.g., the object is seen from a limited range of viewpoints, or a limited range of lighting conditions.

What are the limitations of the discriminative approach? (III)

- In ImageNet, objects are seen with background context. Ground truth for objects is specified by a bounding box surrounding the object which includes the foreground (the object) and the background context (the rest of the bounding box). Discriminative algorithms can exploit the background context (e.g. blue sky gives evidence that the object is an airplane or a bird).
- But this can lead to dataset biases which an algorithm can unfairly exploit. A deep net can incorrectly classify a penguin as a human if a TV is superimposed near the penguin (because a TV is background context that frequently occurs with humans but almost never with penguins).
- Note that these types of *environmental factors* are not modeled in deep networks. But they do appear in computer graphics models and some style-GANs models.

What are the limitations of the discriminative approach? (IV)

- We have illustrated the limitations for the task of object classification. But the same concerns arise for all visual tasks.
- The problems are also apparent for scene classification. Scenes consist of many objects arranged in backgrounds. The number of possibly ways to create scenes is combinatorial, which means that datasets which are representative must be truely gigantic.
- The problems are even clearer for action recognition. Studies suggest that many algorithms are largely relying on background context. E.g., "boxing" is classified by detecting the boxing ring, but boxers can box in the street (or in many locations) and, conversely, people in boxing rings may not be boxing (they could be playing poker as in the film "Lock, Stock, and Two Smoking Barrels"). In such situations, alternative measures like discriminating between very similar actions is a better performance measure.

What are the limitations of the discriminative approach? (IV)

- There are also many ways to defeat current deep networks by making small changes to images, which would not fool a human, but which cause deep networks to make serious mistakes.
- *More fundamental limitations of discriminative methods (at least of deep networks) is that they typically perform only a single task* (e.g., object detection, object classification, etc). By contrast, human can perform an enormous number of visual tasks (detect/classify objects, identify their parts and attributes, estimate their geometry and other environmental factors). Deep Nets do not represent these properties explicitly (if implicitly, then they are buried inside the features of the deep networks).

Out-of-Distribution

- An out-of-distribution task means that the algorithm is trained on date from distribution $P_1(\vec{x}, y)$ but tested on data from distribution $P_2(\vec{x}, y)$. There are other related tasks, such as deciding if a test image $\vec{x}$ is from $P_1(\vec{x}, y)$ or not.
- As stated, it is impossible to solve an out-of-distribution task without making additional assumptions.
- The simplest assumption is to find image features $\vec{f}(\vec{x})$ so that $P_1(y|\vec{f}(\vec{x})) \approx P_2(y|\vec{f}(\vec{x}))$ that they are likely to be invariant to details in the image which are invariant to the task. This has been successfully applied to some examples of domain adaption. But this is a very restrictive assumption.

Generative/Bayesian Perspective

- The Generative/Bayesian perspective is very different. It has, in theory, huge advantages compared to the discriminative approach. But it is harder and requires knowledge of generative/Bayesian methods which are not well-known to the computer vision community.
- A starting point for the Generative/Bayesian approach is the observation (a few slides ago) that image of object depend on environmental factors (shape/viewpoint, texture/material, lighting, etc). There is an external 3D world, the environment, which generate images which are inputs to AI vision systems (and the human visual system).
- This suggests that vision should be thought of as *modeling the environment*. This includes knowing that images consist of compositions of objects arranged on various background structure (e.g., roads, a grass lawn, a university lecture hall). Objects can be thought of as compositions of elementary parts (a horse consists of a head, torso, legs, and tail) which, in turn, can be expressed in terms of subparts (e.g., head contains eyes, nose, mouth, etc.). Actions can be thought of as actors (objects) interacting with each other obeying spatiotemporal relationships).

Generative/Bayesian Perspective

- This perspective suggests that all the variables (environmental factors, etc.) should be represented explicitly. This has many advantages, it enables us to build new objects from existing parts (and recognize that a banana consists of banana slices enveloped in a skin). It also allows us to perform multiple tasks in a consistent manner (their answers will be given by these internal representations. (Question: can we design a taxonomy of out-of-distribution tasks?). (Approximate analysis by synthesis – use features that are invariant to factors/details which we do not care about).
- Note: generalizing to unseen colors/textures/geometries requires the concepts of 3D models and the factorization of images in terms of geometry/viewpoint, texture/material, and lighting/illumination.

Toy Example: Contaminated Data Samples

- We start by describing a toy example of out-of-distribution testing and a Bayesian approach for dealing with it. We assume that some of the testing data is generated from the training distribution $P(\vec{x}, y)$ and the rest is generated by another distribution $Q(\vec{x}, y)$. This can be done by introducing a latent variable $z \in \{0, 1\}$ for each data sample, where $z = 1$ if the data is generated by $P(\vec{x}, y)$ and $z = 0$ if it is generated by $q(\vec{x}, y)$.
- So the test data is generated by $Pr((\vec{x}, y)|z)P(z)$ where $Pr((\vec{x}, y)|z) = \{P(\vec{x}, y\}^z\{Q(\vec{x}, y)\}^{1-z}$ and $P(z = 1) = 1 - \epsilon$ $P(z = 0) = \epsilon$. In other words, the test data is generated from $(1 - \epsilon)P(\vec{x}, y) + \epsilon Q(\vec{x}, y)$, where $Q(\vec{x}, y)$ is another distribution and ϵ is a constant.

Toy Example: Contaminated Data Samples

- If ϵ is small, then classifiers trained on data from $P(\vec{x}, y)$ may still perform well on data from $(1-\epsilon)P(\vec{x}, y) + \epsilon Q(\vec{x}, y)$. But performance will typically degrade badly if ϵ is large. We can also estimate the latent variable z to determine if the data is likely to come from $P(\vec{x}, y)$ or $Q(\vec{x}, y)$.
- Note: the discipline of Robust Statistics partially addresses this problem by showing that some distributions $P(\vec{x}, y)$ are *robust* in the sense that they are unaffected by small amounts of contamination (small ϵ). Gaussian distributions, for example, are known to be non-robust.

Toy Example: Contaminated Data Samples

- How to address this problem? One solution is to use Bayesian methods. Instead of learning the classifiers we learn generative probability distributions $P(\vec{x}|y)$ and $P(y)$ from the training set. We estimate $Q(\vec{x}, y) = Q(\vec{x}|y)Q(y)$ using other data (or modelling assumptions).
- Suppose the data is generated by $Pr((\vec{x}, y)|z)P(z)$. We can estimate z to determine of the data comes from $P(\vec{x}, y)$ or $Q(\vec{x}, y)$ (e.g., by comparing $\max_y Q(y|\vec{x})$ with $\max_y P(y|\vec{x})$ using a weighted threshold to allow for ϵ and model complexity). And then estimate $\hat{y}$ from $P(y|\vec{x})$ or $Q(y|\vec{x})$ as appropriate.

Toy Example: Contaminated Data Samples

- This is the Bayesian/Generative approach. It is conceptually simple but it requires us to learn probability distributions $P(\vec{x}, y)$ and $Q(\vec{x}, y)$ for generating images. But it is very difficult to learn probability distributions for images because the dimensionality of images is very high. It is much easier to learn discriminative distributions $P(y|x)$ because these are much lower-dimensional.
- Recently there is hope that we can make generative models of objects, and perhaps even scenes, using a combination of computer graphics, GANs, and other techniques.

Domain Generalization: Edge Detection Example

- We now consider domain generalization, which is a variant of out-of-distribution learning. This is a based on Konishi *et al.* "Statistical Edge Detection". TPAMI. 2003.
- We assume two different edge detection domains (the Sowerby and the South Florida datasets). These are specified by $P_1(\vec{x}, \vec{y})$ and $P_2(\vec{x}, \vec{y})$. These distributions are very different (Sowerby and South Florida contain outdoor and indoor images respectively).

Domain Generalization: Edge Detection Example

- Here $\vec{x} = \{x_a : a \in \mathcal{D}\}$ are image feature vectors (e.g., image derivatives, which are cues for edges) defined at positions a in the image lattice $\mathcal{D}$ and the variables $y_a \in \{0, 1\}$ denote whether there is an edge at a ($y_a = 1$) or not ($y_a = 0$).
- We re-express the distributions as $P_1(\vec{x}, \vec{y}) = P_1(\vec{x}|\vec{y})P_1(\vec{y})$ and $P_2(\vec{x}, \vec{y}) = P_2(\vec{x}|\vec{y})P_2(\vec{y})$. To simplify we assume that the distributions are factorizable, i.e. $P_i(\vec{x}|\vec{y}) = \prod_{a \in \mathcal{D}} P_i(x_a|y_a)$ for $i = 1, 2$. Similarly for the distributions $P_1(\vec{y}), P_2(\vec{y})$.
- If we know these distributions, then we can detect edges in the two domains by thresholding $\log \frac{P_1(x_a|y_a=1)}{P_1(x_a|y_a=0)}$ and $\log \frac{P_2(x_a|y_a=1)}{P_2(x_a|y_a=0)}$ respectively. This was the first effective statistical/probabilistic edge detector (Konishi *et al.* 2003) which even outperformed s discriminative method (Martin *et al.* 2004).

Domain Generalization: Edge Detection Example

- We now consider domain adaptation. We have annotated training data from the first domain, i.e., we can estimate $P_1(\vec{x}, y), P_1(\vec{x}|y), P_1(y)$, and we have unannotated data from the second domain, i.e., we know $P_2(\vec{x})$. If the domains are sufficiently different, i.e. $P_1(y|\vec{x}) \neq P_2(y|\vec{x})$, then a classifier trained on the first domain will not perform well on the second domain. (Edge detectors trained on South Florida perform badly when tested on Sowerby).
- For the second dataset, *we assume that the distributions of the feature vectors are similar at the edges*, i.e. $P_1(x_a|y_a = 1) = P_2(x_a|y_a = 1)$. This is reasonable because at the edges there is typically a big discontinuity in the image intensity (and can be relaxed). Hence we can use this to estimate $P_2(x_a|y_a = 1)$. Similarly we assume $P_1(\vec{y}) = P_2(\vec{y})$.

Domain Generalization: Edge Detection Example

- To estimate $P_2(x_a|y_a = 0)$, we exploit the fact that *most pixels in images are not edges* so we can approximate it, without needing annotations, by $P_2(x_a) = P_2(x_a|y_a = 0)P_2(y_a = 0) + P_2(x_a|y_a = 0)P_2(y_a = 1)$.
- In practice, Konishi *et al.* relaxed the assumption that $P_1(x_a|y_a = 1) = P_2(x_a|y_a = 1)$, and assumed instead that the magnitude of the filter responses between the two datasets differed by an unknown constant scaling factor which could be estimated by making assumptions about the form of the probability distributions.