9-2022

# Distance based image classification: A solution to generative classification's conundrum?

Wen-yan LIN
*Singapore Management University*, daniellin@smu.edu.sg

Siying LIU

Bing Tian DAI
*Singapore Management University*, btdai@smu.edu.sg

Hongdong LI

**Noname manuscript No.**
(will be inserted by the editor)

# Distance Based Image Classification: A solution to generative classification's conundrum?

**Wen-Yan Lin · Siying Liu · Bing Tian Dai · Hongdong Li**

Received: date / Accepted: date

**Abstract** Most classifiers rely on discriminative boundaries that separate instances of each class from everything else. We argue that discriminative boundaries are counter-intuitive as they define semantics by what-they-are-not; and should be replaced by generative classifiers which define semantics by what-they-are. Unfortunately, generative classifiers are significantly less accurate. This may be caused by the tendency of generative models to focus on easy to model semantic generative factors and ignore non-semantic factors that are important but difficult to model. We propose a new generative model in which semantic factors are accommodated by shell theory's [25] hierarchical generative process and non-semantic factors by an instance specific noise term. We use the model to develop a classification scheme which suppresses the impact of noise while preserving semantic cues. The result is a surprisingly accurate generative classifier, that takes the form of a modified nearest-neighbor algorithm; we term it distance classification. Unlike discriminative classifiers, a distance classifier: defines semantics by what-they-are; is amenable to incremental updates; and scales well with the number of classes.

**Keywords** incremental learning · high dimensions · statistics · shell theory · generative classifiers · anomaly detection · nearest neighbor · distance

Wen-Yan Lin
Singapore Management University
Tel.: +65-6826-1345
E-mail: daniellin@smu.edu.sg

Siying Liu
Institute for Infocomm Research
Tel.: +65-6408-2018
E-mail: liusy1@i2r.a-star.edu.sg

Bing Tian Dai
Singapore Management University
Tel.: +65-6828-9603
E-mail: btdai@smu.edu.sg

Hongdong Li
Australia National University
Tel.: +61-2-6125-7708
E-mail: hongdong.li@anu.edu.au

## 1 Introduction

How do we know apples are apples? Pondering this question leads to a suspicion that discriminative classifiers, which lie at the heart of modern machine learning, may be conceptually flawed.

In discriminative classification, each semantic class is defined by a discriminative boundary which separates instances of itself from everything else. This is effective but rather odd, as semantics become defined by what-they-are-not. Thus, a discriminative classifier views an apple as a fruit that is not an orange, not a banana, and not a pear. We believe many of the eccentricities that trouble computer vision are caused by such, what-they-are-not semantic definitions.

Consider the task of updating a pre-trained discriminative classifier to incorporate a new class. The new class will change the what-they-are-not definitions of existing classes; thus, the update requires an expensive retraining of all classes. Failure to perform retraining will result in catastrophic forgetting [10], with old classes being progressively forgotten as new classes are learned. Other oddities include ill-conditioning in the presence of unknown semantic overlaps and poor scaling with number of classes, problems that are unknown to humans. Perhaps the solution lies in the other major learning paradigm, generative classification.

Generative classifiers treat instances (images in the context of computer vision) as generated outcomes and labels as symbolic representations of individual generative processes [25]. The mathematical linkage of instances to labels, through an assumed generative process, allows machine learning tasks to be reduced to mathematical problems with known solutions. This enforces a rigor, which naturally results in semantics being defined by what-they-are. Unfortunately, the results are often disappointing.

The generative paradigm's seminal work is Bayesian Classification [45,38,37], which elegantly reduces complex learning tasks to a series of statistical inferences. However, despite the guarantees of statistical optimality, Bayesian Classifiers are often significantly less accurate than their discriminative counterparts [30].

More recently, shell theory [24,25] employed the generative framework to show that in a high dimensional hierarchical generative processes, the task of classification can be reduced to a nearest-neighbor search. The proof is elegant. However, as demonstrated in Fig. 1, naive nearest-neighbor classifiers display major quirks that are incompatible with shell theory's assurances of optimality.

We believe the current problems stem from using unrealistic generative models. Most generative models treat instances of a semantic as i.i.d. outcomes of a common generative process; this implicitly assumes image appearance is solely determined by semantic factors. However, non-semantic factors, like background, lighting and object pose, also impact image formation. The challenge arises from the instance specific nature of these non-semantic factors. Incorporating them into the generative model, requires assuming each instance has an individualized generative process; this would sever the mathematical link between instances and labels. However, without instance specific factors, the model cannot be realistic. Our paper attempts to solve this long standing conundrum.

We propose a new generative model where semantic generative factors are accommodated by the hierarchical generative process of shell theory [25]; and non-semantic generative factors by an instance specific noise term. We use this model to develop a classification scheme that explicitly suppresses noise (non-semantic factors) while preserving distance based semantic cues. The result is a surprisingly accurate generative classifier, that takes the form of a modified nearest-neighbor algorithm; we term it distance classification.

In summary, our paper provides the following theoretical contributions:

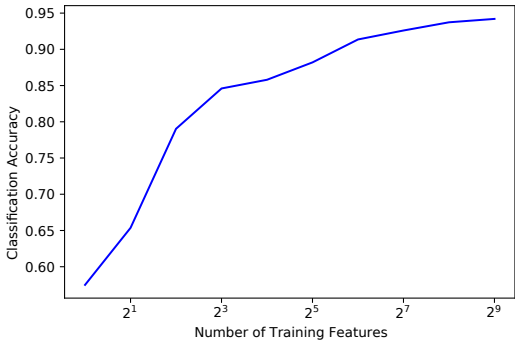
- **New perspective on classification:** We suggest the popular discriminative framework is conceptually flawed.
- **New approach to generative modeling:** We develop a generative model that accommodates, often ignored, non-semantic generative factors.
- **Interpretable distance:** We provide a (more) realistic statistical interpretation of the distances between high dimensional entities.

From a practical perspective, our contribution is the distance classification algorithm, which can be used to replace the final soft-max layer of a (pre-trained) neural network. This provides the following advantages:

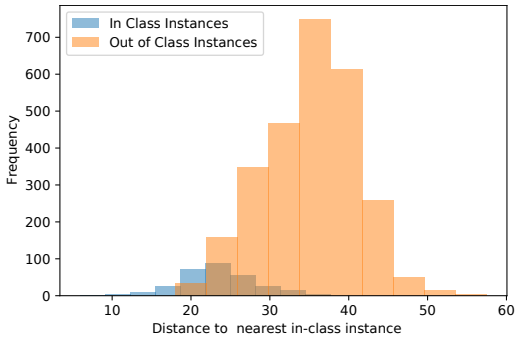
- **Accurate:** The soft-max layer can be replaced with any generative classifier. However, the distance classifier is uniquely accurate; thus, its use does not incur the usual loss of accuracy.
- **Incrementable:** Distance classifiers make it trivial to incrementally add classes to a pre-trained network. Relative to prior incremental learners, updates with distance classifiers have reduced time complexity and greater accuracy.
- **Agnostic to semantic overlaps:** Distance classifiers are agnostic to semantic overlaps, making afore mentioned incremental updates more robust.
- **Scalable inference:** A soft-max layer's inference complexity is $O(N)$, where $N$ is the number of classes. Distance classifiers can employ approximate nearest-neighbor searches, allowing for an inference complexity of $O(log(N))$. This makes it possible to accommodate a large number of classes.

### 1.1 Related Works

Distance classification lies at the intersection of a lively debate on high dimensional statistics, interpretable machine learning and incremental learning. This section provides an overview of each sub-field and distance classification's role in them.




**(a)** A naive nearest neighbor classification is quite accurate, affirming shell theory [25].




**(b)** Distance metrics are terrible at validating class membership, contradicting shell theory.

**Fig. 1:** A puzzling situation where distance metrics are good for classification but poor at validating class membership. Our paper suggests this phenomenon arises from noise induced by non-semantic generative factors and shows how it can be corrected.

**Interpretable Distance:** Distance is a common classification heuristic [32] but is seldom considered a formal classification constraint. This may be because we lack a consistent mathematical framework for interpreting distances. The problem is especially acute for distances between high dimensional entities.

Images and image features are high dimensional entities. According to traditional machine learning theory, they should be impacted by contrast-loss, which renders high dimensional distances meaningless [3]. Aggarwal *et al.* [3] is often cited to explain algorithm failures. However, Fig. 1 shows nearest-neighbor classification, while imperfect, is not a complete failure.

Shell theory [25] suggests Aggarwal *et al.*'s proof is flawed because it assumes all instances are independent, identically distributed outcomes of a single generative process, thus implying the presence of only one one class. Such a generative model would render classification ill-posed in both high and low dimensions; thus, it should not be used to prove that high dimensions are especially meaningless.

Shell theory further shows that in the presence of multiple generative processes, contrast-loss makes distances more meaningful, with the nearest-neighbor of each instance being its most closely related instance [25, 24]. However, shell theory is itself inadequate. Figure 1, shows mutual distance is a good classification metric but also a terrible metric for class validation, simultaneously affirming and contradicting shell theory.

Distance classification extends shell theory to incorporate non-semantic generative factors, creating a mathematical framework that explains the divergence between classification and validation scores. The framework also allows non-semantic factors to be suppressed, leading to a modified nearest-neighbor classifier which avoids such issues.

**Nearest Neighbor Classification:** Nearest neighbor classifiers assign a test instance the class of its nearest neighbor in a training set. The effectiveness of this algorithm is traditionally tied to having sufficient training samples to populate the sample space densely [4]. Computer vision problems typically involve high dimensional sample spaces, that are too large to populate densely. Thus, it is often recommend that data be projected to a low dimensional subspace before performing nearest neighbor classifications [1].

Distance classification provides an alternative perspective. Like shell theory, it argues that in high dimensions, two instances are unlikely to be coincidentally similar. Thus, if non-semantic generative factors are suppressed, the nearest neighbor of an instance will be its most closely related instance. From this perspective, dimensionality reduction is unnecessary; and a nearest neighbor classifier with noise cancellation, can be effective with only a few training instances. This makes the nearest neighbor algorithm much more practical, as shown in Fig. 8.

**Incremental Learning:** Incremental learning seeks to mimic the human ability to add new semantic concepts to an existing pool of knowledge. A naive solution is to fully retrain classifiers each time a semantic is added. However, retraining is prohibitively slow. However, as explained earlier, catastrophic forgetting [10] occurs if old classes are not retrained.

Currently incremental solutions can divided into two main categories. We term the first category weakly incremental learning. This allows retraining with past data [36, 8, 41, 14, 34] but restricts it to ensure efficiency. Weakly incremental learning is effective but updates incur significant computational cost. The second category is strictly incremental learning. Examples include having model parameters dedicated to different batches

of semantics [34,44,35] or training a series of one-class learners [39,9]. Strictly incremental learners can be updated quickly but are usually less accurate.

As distance classifiers represent classes by what-they-are, they can be incrementally updated by appending the appropriate class representation. The result is an algorithm for strict incremental learning, that is significantly more accurate than weak incremental learners.

**Shell Theory [24,25]:** Shell theory introduces a statistical framework for analyzing high dimensional distance. However, the analysis only matches empirical results for normalized data. Distance classification generalizes shell theory to accommodate instance specific noise, allowing for the interpretation of both normalized and unnormalized data. From the perspective of distance classification, shell theory is a special case where data is noise free; and normalization is one of a number of different mechanics to accommodate noise.

As shell theory is only effective on normalized data, it cannot be employed in incremental learning, as it would involve previously learned classes being re-normalized each time a new class is encountered. Since distance classifiers are less dependent on normalization, they are more effective at incremental learning.

### 1.2 Paper Organization

The formulation begins in Sec. 2, with a brief explanation of shell theory. Section 3 and Sec. 4 introduce noise and noise cancellation respectively. Finally, Sec. 5, Sec. 6 and Sec. 7, discuss algorithm design and provide experimental validation.

## 2 Introducing Shell Theory

Statistical machine learning is typically centered on a stochastic model of data generation. These models posit that instances of a class share a common generative process and are thus concentrated in a unique region of the statistical sample space. This makes it possible to statistically infer an instance's semantic class using sample densities estimated from training data.

Such density based models are effective on simple machine learning tasks but cannot be directly applied to complex tasks like image classification. This is because statistical sample spaces grow exponentially with the number of dimensions and image data is notably high dimensional. As a result, image sample spaces are often too large to populate densely.[1]

The traditional solution is to assume data can be captured by a low dimensional projection. This idea connects to the vast literature on low-rank methods [29, 7,2,22] and has been applied in image recognition contexts [40,7]. However, the low-dimensional projections learned from one dataset often generalizes poorly in the presence of bias [26,22]; and is not guaranteed to be appropriate for unseen classes which an incremental learner will encounter.

Shell theory [24,25] is based on a different hypothesis regarding the nature of data. In shell theory, the differences between complex semantic concepts (like "cat" and "dog") may be low dimensional but the concepts themselves are high dimensional [25]. i.e. no single attribute makes a cat, a cat; rather, the concept of "cat" only emerges when multiple attributes are observed simultaneously. Thus, if we are to represent semantics in terms of what-they-are, the problem of high dimensional statistics needs to be addressed.

Shell theory [24,25] suggests that the solution is to perform statistics in terms of distance, rather than density. Shell theory suggests that if each class is described by a long list of attributes, it will be highly improbable that the description is coincidentally satisfied by an instance from outside the class. Shell theory then shows that this intuition can be formalized in the form of a generative model in which high dimensional distances act as proxies for statistical dependencies in the generative process and vice-versa. The results is a distance based statistical framework, which avoids the need to project data onto a low-dimensional subspaces; and accommodates unseen classes.

### 2.1 Hierarchical Generative Process

Apart from its focus on high dimensions, shell theory's other major difference from classical machine learning is its proposed, hierarchical generative model.

In most statistical machine learning literature, data generation is modeled using a set of stochastic generative processes. For example, a tiger image might be modeled as an outcome of some tiger generator. This is simple and intuitive but also incomplete, as the tiger generator's origins are left unexplained.

Shell theory [25] addresses this issue by proposing a hierarchical generative processes which is designed to mimic natural generative processes. In a hierarchical

[1] Image sample spaces may be so large that most potential images never exist. If so, it is possible that even a dataset containing all current images, will fail to densely populate an image sample space.

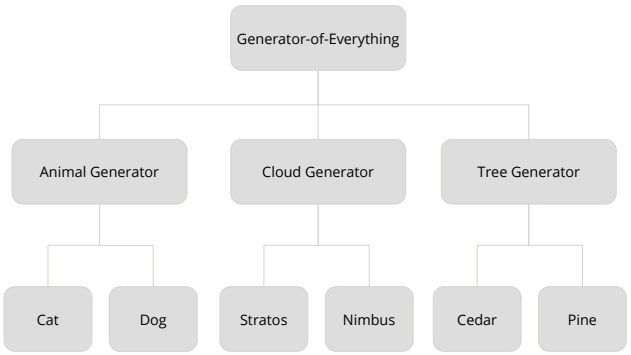



**Fig. 2:** Shell theory assumes that semantics are symbolic representations of a hierarchy of generative processes.

generative processes, the tiger generator is an outcome of an ancestral feline generator, which in turn is the outcome of an ancestral animal generator, etc. The hierarchy is rooted in a hypothesized generator-of-everything illustrated in Fig. 2.

Interestingly, if generative processes are high dimensional and statistically independent given their most recent common ancestor, shell theory predicts that an instance's nearest-neighbor is almost-surely its most closely related instance. If true, many classification tasks could be reduced to a nearest-neighbor discovery problem. This section summarizes shell theory's proof, which we extend (in the following sections) to accommodate real image data.

### 2.2 High Dimensional Distances

Following the notational conventions of statistics, random variables are denoted with upper case letters like $A$. Random variables act as "place-holders" for stochastic outcomes that have yet to be undetermined. If the outcome of a stochastic process is known, it is termed an instance. Instances and denoted with lower case letters like $a$. The vector concatenation of two or more random variables is termed a random vector; random vectors and ordinary vectors are denoted with boldfaced types like $\mathbf{A}$ and $\mathbf{a}$.

Let $\mathbf{A}$ and $\mathbf{B}$ represent the outcomes of two independent generative processes. As their number of dimensions, $k$, becomes large, from [24]:

$$\frac{\|\mathbf{A}-\mathbf{B}\|^2}{k} \overset{a.s.}{=} \frac{\|\boldsymbol{\mu}_\mathbf{A}-\boldsymbol{\mu}_\mathbf{B}\|^2 + v_\mathbf{A} + v_\mathbf{B}}{k}, \tag{1}$$

where $\overset{a.s.}{=}$ denotes almost-surely-equal. $\boldsymbol{\mu}$, $v$ denote mean and total variance respectively, their subscripts indicating the random variable of origin.

Equation (1) uses $k$ as a common denominator to ensure individual terms remain finite as $k$ tends to infinity. For readability, we simplify Eq. (1) to:

$$\|\mathbf{A}-\mathbf{B}\|^2 \overset{a.s.}{=} \|\boldsymbol{\mu}_\mathbf{A}-\boldsymbol{\mu}_\mathbf{B}\|^2 + v_\mathbf{A} + v_\mathbf{B}. \tag{2}$$

Let $\mathbf{c}$ be a constant vector, chosen in a manner that is independent of $\mathbf{A}$. We can set $\mathbf{B}$ in Eq. (2) to have a mean of $\mathbf{c}$ and variance of zero, thus:

$$\|\mathbf{A}-\mathbf{c}\|^2 \overset{a.s.}{=} \|\boldsymbol{\mu}_\mathbf{A}-\mathbf{c}\|^2 + v_\mathbf{A}, \tag{3}$$

Equation (2) and Eq. (3) represent the core distance relationships at the heart of our paper.

We use $\mathcal{A}$ to denote the set of generators in a hierarchical generative process,

$$\begin{aligned}\mathcal{A} = \{&\text{generator-of-everything}, \ldots,\\ &\text{animal generator}, \ldots,\\ &\text{cloud generator}, \ldots,\\ &\text{cat generator}, \ldots,\\ &\text{dog generator}, \ldots,\\ &\text{tree generator}, \ldots\};\end{aligned}$$

and $\mathbf{x}$ to denote an instance created by $\mathcal{A}$.

In a hierarchical generative process, $\mathbf{x}$ is simultaneously the outcome of multiple generators. For example, an individual tiger is simultaneously the outcome of the tiger, feline and animal generators. For any $\mathbf{x}$, the non-trivial ancestral generators (generators with finite difference in either mean or total variance) are represented by the ordered set:

$$\mathcal{A}_\mathbf{x} = \{\mathbf{A}_i \mid i \in \mathbb{Z}_0^+\}, \quad \mathcal{A}_\mathbf{x} \subset \mathcal{A},$$

where $\mathbf{A}_i$ is ancestral to $\mathbf{A}_{i+1}$. From shell theory [25], we know that non-trivial descendant generators, almost-surely have lower total variance than their parents. Thus,

$$a.s. \quad v_{\mathbf{A}_j} > v_{\mathbf{A}_i}, \quad \forall\, \mathbf{A}_i, \mathbf{A}_j \in \mathcal{A}_\mathbf{x},\, i > j; \tag{4}$$

where $a.s.$ denotes almost-surely constraints between instantiated variables.

Let $\mathbf{y}$ denote a different instance. $\mathbf{x}$ and $\mathbf{y}$ will share a most recent common ancestral generator, $\mathbf{A}_c \in \mathcal{A}_\mathbf{x}$. As the descendant generators are assumed to be independent of each other, $\mathbf{x}$ and $\mathbf{y}$ are independent outcomes of $\mathbf{A}_c$. Hence, from Eq. (2):

$$a.s. \quad \|\mathbf{x}-\mathbf{y}\|^2 = 2v_{\mathbf{A}_c}, \quad \mathbf{A}_c \in \mathcal{A}_\mathbf{x}. \tag{5}$$

Equation (5) implies that the distance between $\mathbf{x}$ and $\mathbf{y}$ is the square root of the total variance of their most recently shared ancestor. Equation (4) implies that the total variance decreases as we descend the hierarchy. Thus, the closer $\mathbf{x}$ is to $\mathbf{y}$, the more closely

related they are (share a more recent common ancestor) and vice-versa. If true, image classification could be reduced to a problem of nearest-neighbor discovery. Unfortunately, as we show in Sec. 9.1, in practice, the distance between instances is corrupted by noise. To accommodate this, we introduce a more robust, mean based distance.

### 2.3 Describing Generators by their Mean

Let $\mathbf{x}$ be an instance and $\mathbb{M}$ a set of generator means, of which at least one belongs to an ancestral generator of $\mathbf{x}$. $\boldsymbol{\mu}_c$ denotes the mean in $\mathbb{M}$ that is most closely related to $\mathbf{x}$. For generator-means to act as "descriptions" of their respective generators, $\mathbf{x}$ must be nearer to $\boldsymbol{\mu}_c$ than any other mean in $\mathbb{M}$, i.e.

$$\|\mathbf{x}-\boldsymbol{\mu}_c\|^2 < \|\mathbf{x}-\boldsymbol{\mu}\|^2, \quad \forall\, \boldsymbol{\mu}\in\mathbb{M},\, \boldsymbol{\mu}\neq\boldsymbol{\mu}_c. \tag{6}$$

This section formally derives Eq. (6).

Let $\mathbf{A}_i \in \mathcal{A}_{\mathbf{x}}$ be the generator associated with $\boldsymbol{\mu}_c$ (i.e. $\boldsymbol{\mu}_{\mathbf{A}_i} = \boldsymbol{\mu}_c$). As $\mathbf{x}$ is an outcome of $\mathbf{A}_i$, from Eq. (3):

$$a.s. \quad \|\mathbf{x}-\boldsymbol{\mu}_c\|^2 = v_{\mathbf{A}_i}. \tag{7}$$

Let us consider a different mean, $\boldsymbol{\mu}\in\mathbb{M}$, $\boldsymbol{\mu}\neq\boldsymbol{\mu}_c$. $\mathbf{A}_j \in \mathcal{A}_{\mathbf{x}}$, is used to denote the most recent ancestral generator of $\mathbf{x}$ and $\boldsymbol{\mu}$. From Eq. (3):

$$a.s. \quad \|\mathbf{x}-\boldsymbol{\mu}\|^2 = v_{\mathbf{A}_j} + \|\boldsymbol{\mu}_{\mathbf{A}_j}-\boldsymbol{\mu}\|^2. \tag{8}$$

Observe that because $\boldsymbol{\mu}_c$ is more closely related to $\mathbf{x}$ than $\boldsymbol{\mu}$, therefore, $i > j$. Thus, from Eq. (4), $v_{\mathbf{A}_i} < v_{\mathbf{A}_j}$. As such, Eq. (7) and Eq. (8) can be combined to yield:

$$a.s. \quad \|\mathbf{x}-\boldsymbol{\mu}_c\|^2 < \|\mathbf{x}-\boldsymbol{\mu}\|^2, \quad \forall\, \boldsymbol{\mu}\in\mathbb{M},\, \boldsymbol{\mu}\neq\boldsymbol{\mu}_c. \tag{9}$$

This formally derives Eq. (6). Less formally, Eq. (9) proves that **the nearest generator-mean to a given feature, is the generator-mean most closely related to that feature**.

What about the most closely related feature to a given generator-mean? Let $\mathbf{y}$ be an instance that is not descended from $\mathbf{A}_i$. If so, $\mathbf{y}$ must be related to $\boldsymbol{\mu}_c$ (recall that $\boldsymbol{\mu}_c$ is the mean of $\mathbf{A}_i$) by a closest common generator, $\mathbf{A}_k \in \mathcal{A}_{\mathbf{x}}$, $i > k$. Thus, from Eq. (3) and Eq. (4), the distance of $\mathbf{y}$ to $\boldsymbol{\mu}_c$ is almost-surely greater than that of $\mathbf{x}$ to $\boldsymbol{\mu}_c$:

$$\begin{aligned} &a.s. \quad \|\mathbf{y}-\boldsymbol{\mu}_c\|^2 = v_{\mathbf{A}_k} + \|\boldsymbol{\mu}_{\mathbf{A}_k}-\boldsymbol{\mu}_c\|^2 > v_{\mathbf{A}_i} \\ &\Rightarrow \quad \|\mathbf{y}-\boldsymbol{\mu}_c\|^2 > \|\mathbf{x}-\boldsymbol{\mu}_c\|^2 = v_{\mathbf{A}_i}. \end{aligned} \tag{10}$$

Equation (10) proves that **the nearest feature to a generator-mean, is also the feature most closely related to that generator-mean**.

Equation (9) and Eq. (10) suggest the mean of a generative process can be used to identify instances of that process, turning them into "descriptions" of the process.

## 3 Introducing Noise

Shell theory makes nearest-neighbor / nearest-mean classifiers appear attractive. However, the practical performance of such distance based classifiers is often disappoint. We hypothesize that this gap between theory and reality occurs because, like most generative models, shell theory implicitly assumes image formation is only impacted by semantics. However, in practice, background, object pose and lighting, play a large role in determining the appearance of the final image but are not present in the generative model.

Our solution is to group the non-semantic generative factors into an instance specific noise term. Thus, if $\mathbf{A}$ is an idealized generator, its noisy counterpart is $\mathbf{A}(t)$:

$$\mathbf{A}(t) = \mathbf{A} + \mathbf{N}(t), \quad \mathbf{A}\in\mathcal{A}. \tag{11}$$

Here, $t$ represents the instance's index and $\mathbf{N}(t)$ the instance specific perturbations.

We assume that noise elements are: independent of one another, independent of the ideal generators, and unbiased. More formally, denoting the $i^{th}$ element of $\mathbf{N}(t)$ as $\mathbf{N}(t)[i]$ and statistical independence as $\perp\!\!\!\perp$,

$$\begin{aligned} &E(\mathbf{N}(t)) = \mathbf{0}, && \forall\, t\,; \\ &\mathbf{N}(t) \perp\!\!\!\perp \mathbf{A}, && \forall t,\, \mathbf{A}\in\mathcal{A}\,; \\ &\mathbf{N}(t) \perp\!\!\!\perp \mathbf{N}(t'), && \forall\, t\neq t'\,; \\ &\mathbf{N}(t)[i] \perp\!\!\!\perp \mathbf{N}(t)[j], && \forall\, i\neq j\,. \end{aligned} \tag{12}$$

Let $\mathbf{c}$ be a constant reference vector, chosen in a manner which is independent of $\mathbf{A}(t)$. The distance of $\mathbf{A}(t)$ from $\mathbf{c}$ can be expressed in terms of an idealized distance with a noise induced perturbation, using Eq. (3) and Eq. (11):

$$\begin{aligned} &\|\mathbf{A}(t)-\mathbf{c}\|^2 \\ =&\|\mathbf{A}+\mathbf{N}(t)-\mathbf{c}\|^2 \\ =&\|\mathbf{A}-\mathbf{c}\|^2 + 2\times\langle\mathbf{N}(t),\ \mathbf{A}-\mathbf{c}\rangle + \|\mathbf{N}(t)\|^2 \\ =&\|\mathbf{A}-\mathbf{c}\|^2 + \Gamma(t,\mathbf{c}) + \|\mathbf{N}(t)\|^2. \end{aligned} \tag{13}$$

When the the number of dimensions, $k$, is large, Eq. (12) implies that the middle gamma term tends to zero:

$$\begin{aligned} \Gamma(t,\mathbf{c}) =& 2\times\langle\mathbf{N}(t),\ \mathbf{A}(t)-\mathbf{c}\rangle \\ \approx& 2\times k\times E\left(\mathbf{N}(t)[i]\right)\times E\left(\mathbf{A}(t)[i]-\mathbf{c}[i]\right) \\ =& 0. \end{aligned} \tag{14}$$

Let us study the impact of noise on idealized distances of Sec. 2.

### 3.1 Impact of Noise on Distance

Let $\mathbf{x}(t)$ denote a noisy feature, $\mathbf{x}_t$ its ideal counterpart and $\mathbf{c}$ a reference vector, chosen in a manner that is independent of one of the generators of $\mathbf{x}_t$ (recall that an instance can have multiple generators). Equation (13) implies that the noisy distance, $\|\mathbf{x}(t)-\mathbf{c}\|$ is related to the ideal distance $\|\mathbf{x}_t-\mathbf{c}\|$ by:

$$\begin{aligned} a.s.\quad \|\mathbf{x}(t)-\mathbf{c}\|^2 =& \|\mathbf{x}_t-\mathbf{c}\|^2+\gamma(t,\mathbf{c})+\|\mathbf{n}(t)\|^2 \\ \approx& \|\mathbf{x}_t-\mathbf{c}\|^2+\|\mathbf{n}(t)\|^2; \end{aligned} \tag{15}$$

where $\mathbf{n}(t)$, $\gamma(t,\mathbf{c})$ are instantiations of $\mathbf{N}(t)$, $\Gamma(t,\mathbf{c})$ and Eq. (14) indicates that $\gamma(t,\mathbf{c})$ is small.

Let $\boldsymbol{\mu}$ be a generator-mean, Eq. (15) implies:

$$\begin{aligned} a.s.\quad \|\mathbf{x}(t)-\boldsymbol{\mu}\|^2 =& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2+\gamma(t,\boldsymbol{\mu})+\|\mathbf{n}(t)\|^2 \\ \approx& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2+\|\mathbf{n}(t)\|^2. \end{aligned} \tag{16}$$

Equation (16) is true because for any given $\mathbf{x}_t$ and $\boldsymbol{\mu}$, there will exist a closest common ancestral generator from which both can be considered independent descendants. Thus, $\boldsymbol{\mu}$ fulfills the requirement for $\mathbf{c}$ in Eq. (15).

Finally, the distance between two noisy features, $\mathbf{x}(t)$, $\mathbf{x}(t')$, will be:

$$\begin{aligned} a.s.\quad & \|\mathbf{x}(t)-\mathbf{x}(t')\|^2 \\ =& \|\mathbf{x}_t-\mathbf{x}(t')\|^2+\gamma(t,\mathbf{x}(t'))+\|\mathbf{n}(t)\|^2 \\ =& \|\mathbf{x}_t-\mathbf{x}_{t'}\|^2+\gamma(t,\mathbf{x}(t'))+\gamma(t',\mathbf{x}_t) \\ & +\|\mathbf{n}(t)\|^2+\|\mathbf{n}(t')\|^2 \\ \approx& \|\mathbf{x}_t-\mathbf{x}_{t'}\|^2+\|\mathbf{n}(t)\|^2+\|\mathbf{n}(t')\|^2. \end{aligned} \tag{17}$$

Equation (17) comes about because $\mathbf{x}_t$ and $\mathbf{x}_{t'}$ will have a common ancestral generator. As both $\mathbf{x}_t$ and $\mathbf{x}_{t'}$ are independent outcomes of their common ancestor, we can alternately treat each instance as $\mathbf{c}$ in Eq. (15), yielding Eq. (17).

Equation (15), Eq. (16) and Eq. (17) show that noise induced perturbations can be decomposed into two terms:

- *Gamma perturbations*: $\gamma(t,\_)$, that depend on both the instance in question, $t$, and the reference vector. $\gamma(t,\_)$ tends to be small;
- *Noise magnitude perturbations*: $\|\mathbf{n}(t)\|^2$ are constant for any given instance, $t$.

Of the two terms, it is most important to cancel out the dominant, noise magnitude perturbation. Fortunately, it is constant for any given instance, $t$, making it possible to remove the noise induced perturbations through carefully selected distance comparisons.

### 3.2 Noise Invariant Nearest Mean Classifiers

Before explaining our solution for noise cancellation, we illustrate the implicit noise cancellation embedded in a nearest mean classifier.

Let $\mathbb{M}$ be a set of generator-means[2].

$\mathbf{x}(t)$ is a noisy test feature. From Eq. (16), the distance of $\mathbf{x}(t)$ to a generator-means is:

$$\begin{aligned} a.s.\quad & \|\mathbf{x}(t)-\boldsymbol{\mu}\|^2 \\ =& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2+\gamma(t,\boldsymbol{\mu})+\|\mathbf{n}(t)\|^2 \\ \approx& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2+\|\mathbf{n}(t)\|^2, \qquad \forall\boldsymbol{\mu}\in\mathbb{M}. \end{aligned} \tag{18}$$

Thus, if $\boldsymbol{\mu}_c$ is the generator-mean in $\mathbb{M}$ most closely related to $\mathbf{x}_t$, from Eq. (9) and Eq. (18):

$$\begin{aligned} a.s.\quad & \|\mathbf{x}(t)-\boldsymbol{\mu}\|^2-\|\mathbf{x}(t)-\boldsymbol{\mu}_c\|^2 \\ =& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2-\|\mathbf{x}_t-\boldsymbol{\mu}_c\|^2+\gamma(t,\boldsymbol{\mu})-\gamma(t,\boldsymbol{\mu}_c) \\ & +\|\mathbf{n}(t)\|^2-\|\mathbf{n}(t)\|^2 \\ =& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2-\|\mathbf{x}_t-\boldsymbol{\mu}_c\|^2+\gamma(t,\boldsymbol{\mu})-\gamma(t,\boldsymbol{\mu}_c) \\ \approx& \|\mathbf{x}_t-\boldsymbol{\mu}\|^2-\|\mathbf{x}_t-\boldsymbol{\mu}_c\|^2>0, \\ & \qquad \forall\boldsymbol{\mu}\in\mathbb{M},\ \boldsymbol{\mu}\neq\boldsymbol{\mu}_c. \end{aligned} \tag{19}$$

Equation (19) shows that the nearest generator-mean to $\mathbf{x}(t)$ is also the generator-mean most closely related to $\mathbf{x}(t)$, a property that is happily noise invariant.

The noise invariance in Eq. (19) arises from the act of comparing $\mathbf{x}(t)$ to a number of different means. The dominant, noise magnitude perturbation is constant for a given $t$, and is cancelled by the comparisons.

Noise cancellation becomes even stronger if gamma perturbations vary smoothly with $\boldsymbol{\mu}$. If so, the already weak gamma perturbations would be further weakened by the partial self cancellation in Eq. (19). The self cancellation effect is strongest for the most important comparisons, those that take place between similar means.

Unfortunately, noise cancellation is not to be taken for granted. Let us demonstrate it by reversing the classification problem. Instead of seeking the generator-mean most closely related to a feature, we seek the feature most closely related to a generator-mean.

Let us assume $\boldsymbol{\mu}_c$ is more closely related to $\mathbf{x}_t$ than $\mathbf{x}_{t'}$. From Eq. (10),

$$a.s.\quad \|\mathbf{x}_{t'}-\boldsymbol{\mu}_c\|^2-\|\mathbf{x}_t-\boldsymbol{\mu}_c\|^2>0. \tag{20}$$

This allows the distance of an instance from a given generator-mean, to act as an indicator of the closeness

[2] A generator-mean can be estimated by averaging a large number of its generated instances. As noise has a mean of zero, this estimate is unbiased.

of their relationship. In the presence of noise, Eq. (20) becomes:

$$\begin{aligned} a.s. \quad & \|\mathbf{x}(t') - \boldsymbol{\mu}_c\|^2 - \|\mathbf{x}(t) - \boldsymbol{\mu}_c\|^2 \\ =& \|\mathbf{x}_{t'} - \boldsymbol{\mu}_c\|^2 + \gamma(t', \boldsymbol{\mu}_c) + \|\mathbf{n}(t')\|^2 \\ & - \|\mathbf{x}_t - \boldsymbol{\mu}_c\|^2 - \gamma(t, \boldsymbol{\mu}_c) - \|\mathbf{n}(t)\|^2 \\ \approx& \|\mathbf{x}_{t'} - \boldsymbol{\mu}_c\|^2 - \|\mathbf{x}_t - \boldsymbol{\mu}_c\|^2 + \|\mathbf{n}(t')\|^2 - \|\mathbf{n}(t)\|^2. \end{aligned} \tag{21}$$

This may be greater than, less than, or equal to zero, depending on the noise magnitude terms, $\|\mathbf{n}(t')\|^2 - \|\mathbf{n}(t)\|^2$, which are not small.

Unlike the ideal case in Eq. (20), in the presence of noise, Eq. (21) shows that the feature nearest to a generator-mean, is not necessarily the feature most closely related to the generator-mean. Thus, the validity of assigning a feature to a generator (label validation), cannot be measured by the raw distance between the feature and the generator-mean.

In summary, the raw distance of a feature to a generator-mean is a good classification measure but a terrible validation measure. This is empirically verified in Fig. 3. To our knowledge, we are the first paper to report this peculiar divergence in classification and validation performance.

## 4 Noise Cancellation for Label Validation

If the distance of a feature to a generator-mean is to be directly interpretable, the "signal-to-noise-ratio" needs to be modified. This section proposes three different solutions. Section 4.1 uses a reference vector based noise cancellation to explicitly estimate noise cancelled distances. Section 4.2 shows that certain ratios are both intuitive and noise cancelling. Section 4.3 uses normalization to enhance underlying distance based constraints, creating a pre-processor for downstream algorithms.

### 4.1 Reference Vector Based Noise Cancellation

Reference vector based noise cancellation, uses a comparison to an appropriately chosen reference vector for noise cancellation.

Let $\mathbf{E} \in \mathcal{A}$ be the generator-of-everything and $\mathbf{z}$ an independently selected reference vector; from Eq. (3):

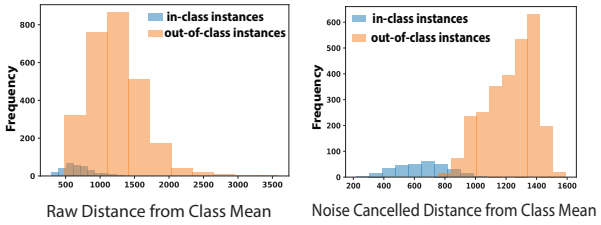
$$\|\mathbf{E} - \mathbf{z}\|^2 \overset{a.s.}{=} \|\boldsymbol{\mu}_\mathbf{E} - \mathbf{z}\|^2 + v_\mathbf{E} = c_\mathbf{z}, \tag{22}$$

where $c_\mathbf{z} = \|\boldsymbol{\mu}_\mathbf{E} - \mathbf{z}\|^2 + v_\mathbf{E}$ is a constant. If $\mathbf{x}_t$ is an ideal feature, it is necessarily an outcome of the generator-of-everything; thus, Eq. (22) implies the distance of $\mathbf{x}_t$ from $\mathbf{z}$ is:

$$a.s. \quad \|\mathbf{x}_t - \mathbf{z}\|^2 = c_\mathbf{z}, \quad \forall t. \tag{23}$$

| | Raw distances | Noise cancellation with reference vector (random vector) | (mean) |
|---|---|---|---|
| nearest-mean classification (accuracy) | 0.945 | 0.945 | 0.945 |
| label validation (auroc) | 0.874 | 0.965 | 0.992 |

**(a)** Raw distance to class-means can be used for classification measure but not for validating class membership. Section 3.2 explains the phenomenon analytically. Section 4 corrects the problem through noise cancellation.




**(b)** Histogram of distance from airplane's class mean, before and after noise cancellation. Noise cancelled distances can be a validation measure for class membership.

**Fig. 3:** Experiments on STL-10 [11]. Images are represented with ResNet-50 features [15].

$\mathbf{x}(t)$ is the noisy counterpart of $\mathbf{x}_t$, thus from Eq. (15):

$$\begin{aligned} a.s. \quad \|\mathbf{x}(t) - \mathbf{z}\|^2 =& \|\mathbf{x}_t - \mathbf{z}\|^2 + \gamma(t, \mathbf{z}) + \|\mathbf{n}(t)\|^2 \\ =& c_\mathbf{z} + \gamma(t, \mathbf{z}) + \|\mathbf{n}(t)\|^2 \\ \approx& c_\mathbf{z} + \|\mathbf{n}(t)\|^2, \qquad \forall t. \end{aligned} \tag{24}$$

From Eq. (16), the distance of $\mathbf{x}(t)$ from a generator-mean is:

$$\begin{aligned} a.s. \quad & \|\mathbf{x}(t) - \boldsymbol{\mu}\|^2 \\ =& \|\mathbf{x}_t - \boldsymbol{\mu}\|^2 + \gamma(t, \boldsymbol{\mu}) + \|\mathbf{n}(t)\|^2 \\ \approx& \|\mathbf{x}_t - \boldsymbol{\mu}\|^2 + \|\mathbf{n}(t)\|^2, \qquad \forall t, \forall \boldsymbol{\mu} \in \mathbb{M}, \end{aligned} \tag{25}$$

where $\mathbb{M}$ is a set of generator-means.

By subtracting Eq. (24), from Eq. (25) we can eliminate the primary noise perturbation, $\|\mathbf{n}(t)\|^2$:

$$\begin{aligned} & d^2(\mathbf{x}(t), \boldsymbol{\mu}, \mathbf{z}) \\ =& \|\mathbf{x}(t) - \boldsymbol{\mu}\|^2 - \|\mathbf{x}(t) - \mathbf{z}\|^2 \\ =& \|\mathbf{x}_t - \boldsymbol{\mu}\|^2 + \gamma(t, \boldsymbol{\mu}) - \gamma(t, \mathbf{z}) - c_\mathbf{z} \\ \approx& \|\mathbf{x}_t - \boldsymbol{\mu}\|^2 - c_\mathbf{z}, \qquad \forall t, \forall \boldsymbol{\mu} \in \mathbb{M}. \end{aligned} \tag{26}$$

Thus, the noise cancelled distance, $d^2(\mathbf{x}(t), \boldsymbol{\mu}, \mathbf{z})$, is a noise invariant estimate of $\mathbf{x}(t)$'s distance to $\boldsymbol{\mu}$, with a constant offset of -$c_\mathbf{z}$.

From Eq. (10) and Eq. (26), we know that if $\boldsymbol{\mu}_c \in \mathbb{M}$ is more closely related to $\mathbf{x}_t$ than $\mathbf{x}_{t'}$:

$$\begin{aligned} & \|\mathbf{x}_{t'} - \boldsymbol{\mu}_c\|^2 > \|\mathbf{x}_t - \boldsymbol{\mu}_c\|^2, \\ \Rightarrow \quad & d^2(\mathbf{x}(t'), \boldsymbol{\mu}_c, \mathbf{z}) \gtrapprox d^2(\mathbf{x}(t), \boldsymbol{\mu}_c, \mathbf{z}). \end{aligned} \tag{27}$$

Equation (27) implies that the noise cancelled distance, $d^2(\mathbf{x}(t), \boldsymbol{\mu}_c, \mathbf{z})$, is an indicator for how closely related $\mathbf{x}(t)$ is to $\boldsymbol{\mu}_c$.

**Common Mean as Reference Vector:** Let us assume that the generator-means in $\mathbb{M}$ are stochastic outcome of a most recent common ancestor $\mathbf{E}$[3]. $\mathbf{m}$ is used to denote the mean of $\mathbf{E}$. Noise cancellation is enhanced if $\mathbf{m}$ is the reference vector in Eq. (26) (i.e. $\mathbf{z} = \mathbf{m}$). The explanation is as follows.

If gamma perturbations vary smoothly with choice of reference vector, $\mathbf{z}$, noise cancellation in Eq. (26) will be most effective when $\mathbf{z}$ is as close as possible to individual means in $\mathbb{M}$, while being chosen in a manner that is independent of $\mathbf{E}$. $\mathbf{m}$ satisfies both properties, as explained below.

First, the mean of $\mathbb{M}$ is the vector which minimizes the average distance to each element in $\mathbb{M}$. From the towering property of conditional expectation, the expected mean of $\mathbb{M}$ is the mean of the overall generator, $\mathbf{E}$, which is $\mathbf{m}$. Second, $\mathbf{m}$ is generated during the creation of generator $\mathbf{E}$. As $\mathbf{m}$ is created prior to any instances of $\mathbf{E}$, its creation is independent of $\mathbf{E}$.

From Eq. (3), we know all instances descended from $\mathbf{E}$ will be at a constant distance from $\mathbf{m}$. $c_{\mathbf{m}}$ is used to denote the square of this distance. Thus,

$$a.s. \quad \|\mathbf{x}_t - \mathbf{m}\|^2 = c_{\mathbf{m}}, \quad \forall t \in \mathcal{T}, \tag{28}$$

where the set $\mathcal{T}$ contains all indices of features generated by $\mathbf{E}$. Similar to Eq. (26), the noise cancelled distance with $\mathbf{m}$ as reference vector is:

$$\begin{aligned} & d^2(\mathbf{x}(t), \boldsymbol{\mu}, \mathbf{m}) \\ =& \|\mathbf{x}(t) - \boldsymbol{\mu}\|^2 - \|\mathbf{x}(t) - \mathbf{m}\|^2 \\ =& \|\mathbf{x}_t - \boldsymbol{\mu}\|^2 + \gamma(t, \boldsymbol{\mu}) - \gamma(t, \mathbf{m}) - c_{\mathbf{m}} \\ \approx& \|\mathbf{x}_t - \boldsymbol{\mu}\|^2 - c_{\mathbf{m}}, \qquad \forall t \in \mathcal{T}, \forall \boldsymbol{\mu} \in \mathbb{M}. \end{aligned} \tag{29}$$

The approximation in Eq. (29) is better than that in Eq. (26), because it enhances the partial self cancellation of gamma perturbations, $\gamma(t, \boldsymbol{\mu}) - \gamma(t, \mathbf{m})$. The impact on validation accuracy is illustrated in Fig. 3.

### 4.2 Ratio Based Noise Cancellation

Distance ratios are often more interpretable than raw distances. A famous example is SIFT's ratio test [27], where the best feature match is validated by its ratio to the second best feature match.

Interestingly, the ratio of $\mathbf{x}(t)$'s distance to means $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$, is noise invariant when the distances $\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2$, $\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2$ are similar. This allows classification thresholds to be defined in terms of appropriately chosen distance ratios.

Let $\mathbf{x}(t)$ be a feature and $\mathbf{x}_t$ its ideal counterpart; $\boldsymbol{\mu}_i, \boldsymbol{\mu}_j \in \mathbb{M}$ be a pair of means whose distances to $\mathbf{x}(t)$ are similar; i.e.

$$\Delta_{ij} = \|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2 - \|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2,$$

is small relative to either $\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2$ or $\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2$. Noise perturbations,

$$\gamma(t, \boldsymbol{\mu}_i) + \|\mathbf{n}(t)\|^2, \qquad \gamma(t, \boldsymbol{\mu}_j) + \|\mathbf{n}(t)\|^2,$$

are also assumed to be small relative to either $\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2$ or $\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2$.

The ratio of $\mathbf{x}(t)$'s distance to $\boldsymbol{\mu}_i$, $\boldsymbol{\mu}_j$ is:

$$\begin{aligned} & \frac{\|\mathbf{x}(t) - \boldsymbol{\mu}_i\|^2}{\|\mathbf{x}(t) - \boldsymbol{\mu}_j\|^2} \\ =& \frac{\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2 + \gamma(t, \boldsymbol{\mu}_i) + \|\mathbf{n}(t)\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2 + \gamma(t, \boldsymbol{\mu}_j) + \|\mathbf{n}(t)\|^2} \\ \approx& \frac{\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2} \times \left(1 + \frac{\gamma(t, \boldsymbol{\mu}_i) + \|\mathbf{n}(t)\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2}\right) \\ & \quad \times \left(1 - \frac{\gamma(t, \boldsymbol{\mu}_j) + \|\mathbf{n}(t)\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2}\right) \\ \approx& \frac{\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2} \\ & \quad \times \left(1 + \frac{\gamma(t, \boldsymbol{\mu}_i) + \|\mathbf{n}(t)\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2 + \Delta_{ij}} - \frac{\gamma(t, \boldsymbol{\mu}_j) + \|\mathbf{n}(t)\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2}\right) \\ \approx& \frac{\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2} \left(1 + \frac{\gamma(t, \boldsymbol{\mu}_i) - \gamma(t, \boldsymbol{\mu}_j)}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2}\right). \end{aligned} \tag{30}$$

Observe that the first order terms of Eq. (30) contain only gamma terms, that are both small and self cancelling. Thus, the ratio

$$\frac{\|\mathbf{x}(t) - \boldsymbol{\mu}_i\|^2}{\|\mathbf{x}(t) - \boldsymbol{\mu}_j\|^2} \approx \frac{\|\mathbf{x}_t - \boldsymbol{\mu}_i\|^2}{\|\mathbf{x}_t - \boldsymbol{\mu}_j\|^2}, \tag{31}$$

is noise invariant.

The strength of using distance ratios is they can be adapted for fine-grained comparison. Their drawback is that noise cancellation is only valid for similar distance; this restricts ratio usage to local comparisons.

### 4.3 Shell Normalization

Shell theory [25] suggests a normalization step is required to enhance distance based classification constraints. The procedure is as follows: Let $\mathbf{m}$ be the

[3] Unlike in Eq. (22), in this context, $\mathbf{E}$ need not be the generator-of-everything. For example, if the dataset consists of different cat species, $\mathbf{E}$ would be the feline generator.

mean of the most recent ancestral generator of a set of features. $\mathbf{m}$ is subtracted from every feature and the resultant vector is then divided by its magnitude, such that it becomes a unit-vector:

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \mathbf{m}}{\|\mathbf{x} - \mathbf{m}\|}, \tag{32}$$

where $\hat{\mathbf{x}}$ denotes the normalized version of $\mathbf{x}$.

From the current paper's perspective, normalization is a form of noise cancellation, with the associated rescaling reducing the impact of noise while also enhancing "shell distinctiveness", as explained in shell theory [25]).

Normalization's advantage is it creates "better" features. Thus, normalization can boost the accuracy of downstream algorithms, like clustering, without requiring explicit modification of the algorithm. However, normalization's feature transformation can also be a disadvantage. For example, in incremental learning, the desired normalization changes as classes are added. Ensuring the normalization is uptodate would require re-normalizing instance of previously learned classes, as new classes are introduced, significantly slowing the incremental update. In this scenario, ratio based noise cancellation from Sec. 4.2 would be more appropriate.

## 5 Training a Distance Classifier

The previous sections have focused on assigning each feature to its most closely related generative process. However, classification tasks are formulated in terms of labels, not generative processes. Thus, a mapping from labels to generative processes, needs to be established.

The naive solution is to assume each label corresponds to a unique generative process and represent each semantic by the mean of its features. Empirical results in Fig. 3 and shell theory [25] show this is effective. However, it may not be ideal.

Let us consider the label, "church". This label can be applied to both the interior or facade of a christian religious building. However, church interiors are likely to be more closely related to the interiors of other buildings, than they are to church facades. Likewise, church facades are likely to be more closely related to the facades of other buildings, than to the church interiors. This suggests labels are symbolic representations of an ensemble of individual generative processes, not all of which are closely related. This is illustrated in Fig. 4.

This perspective simultaneously explains the complex semantic overlaps common in our language; and allows that the semantic mean can be a good identifier for instances of a semantic. After all, an instance that is unusually close to one of the ensemble of generative means, is likely to also be unusually close to the overall semantic mean.

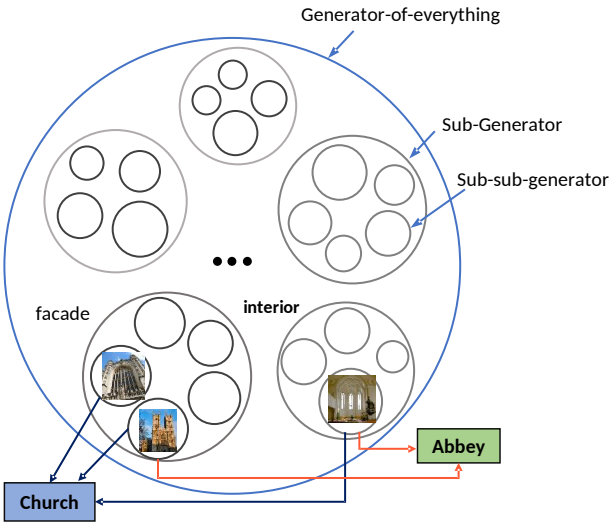



**Fig. 4:** Labels are symbolic representations of an ensemble of generative processes, not all of which are closely related. Each label should be represented a corresponding ensemble of means, which can be estimated through K-Means clustering [19].

Thus, although labels can be represented with a single mean, it is better to represent them with an ensemble of means, that correspond as closely as possible to the individual generator-means. In practice, these means can be estimated through K-Means clustering [19] of a label's features:

- Prior to clustering, all features of a class are normalized with respect to the semantic mean, providing implicit noise cancellation (Sec. 4.3).
- The number of clusters is set at $K = 30$; this (large) number should suffice ensure each unrelated process is represented by an individual cluster.
- To accommodate K-means clustering's randomness, the clustering procedure is repeated $r = 10$ times.
- Generator-means are estimated by taking the mean of the un-normalized features in each cluster. The set of estimated generator-means, is used as a representation of the semantic.

The process is summarized in Algorithm 1 and represents a Distance Classifier's training step. The semantic label of a test feature is inferred from the test feature's distance to the stored generator-means. Two different forms of inference are discussed, Distance Top-N in Sec. 6 and Distance (VS), (VW) in Sec. 7.

### 5.1 Design Choices

Readers may have noticed the proposed classifier uses a very large representation size. This design choice was made because employing multiple cluster-means improves accuracy but does not detract from scalability. We explain the design choice below.

Figure 5 shows the accuracy benefits that accrue to using a large number of clusters as well as how adding repetitions gives an additional performance boost. To maximize accuracy, we represent each class with 300 cluster-means (based on 30 clusters and 10 repetitions).

Such a large representation sizes are a-typical because traditional classifier perform inference with a soft-max layer, which has an inference complexity of $O(N)$, where $N$ is the representation size. In contrast, a distance classifier can use approximate nearest neighbor discovery, which has an inference complexity of $O(log(N))$, making representation sizes less of a concern as shown in Fig. 6.

In practice, we find that inference speed is not a bottleneck in most current datasets, as shown in Table 4; and thus, not a current concern. However, the lower complexity of distance classification opens up the possibility of to very large inference engines encompassing millions of classes.

## 6 Distance Top-N Classifier (Inference)

Top-N classification is the task of finding the $N$ best labels for a given feature, $\mathbf{x}(t)$.

Let $\mathbb{M}_i$ denote the set of generator-means used to "describe" label-$i$. $\mathbb{M}_i$ can be estimated using Algorithm 1 in Sec. 5. $\mathfrak{M}$ is the set of all estimated generator means and $\mathbb{L}$ is the set of possible labels.

$$\mathbb{M}_i \subset \mathfrak{M}; \qquad \mathbb{M}_i \cap \mathbb{M}_j = \varnothing, \quad \forall\, i \neq j \in \mathbb{L}.$$

The squared distance of $\mathbf{x}(t)$ from label-$i$ is the smallest squared distance from $\mathbf{x}(t)$ to any generator-mean associated with label-$i$:

$$min_i(t) = \min_{\boldsymbol{\mu} \in \mathbb{M}_i} \{\|(\mathbf{x}(t) - \boldsymbol{\mu}\|^2\}. \tag{33}$$

From Eq. (19), we know that $\mathbf{x}(t)$ is most closely related to the nearest generator mean (and thus its semantic label). Therefore, $\mathbf{x}(t)$'s top label is:

$$y_1(t) = \underset{i \in \mathbb{L}}{\operatorname{argmin}}\, min_i(t). \tag{34}$$

The top $N^{th}$ label, $y_N(t)$, is estimated with the same algorithm, modified to exclude the previous top $N$-1 labels from consideration.

**Algorithm 1: Learning Label-i**

**Input:**
$K = 30$; # number of clusters
$R = 10$; # number of repetitions
$\mathcal{X}_i$. # set of features from class $i$

**Initialization:**
$r = 0$;
$\overline{\mathbf{x}}_i = \text{mean}(\mathcal{X}_i)$;
$\widetilde{\mathcal{X}_i} = \left\{ \hat{\mathbf{x}}(t) = \frac{\mathbf{x}(t) - \overline{\mathbf{x}}_i}{\|\mathbf{x}(t) - \overline{\mathbf{x}}_i\|} \;\middle|\; \mathbf{x}(t) \in \mathcal{X}_i \right\}$;
$\mathbb{M}_i = \{\}$.

**Estimate generator-means:**
**while** $r < R$ : **do**
  # $\mathcal{C}_k$ is the set of indices belonging to cluster $k$ #
  $\{\mathcal{C}_k\}$ = K-means-cluster($\widetilde{\mathcal{X}_i}, K$)
  **for** $k < K$ : **do**
    $\mathbf{m}_k = \text{mean}(\{\mathbf{x}(t) \,|\, t \in \mathcal{C}_k\})$
    $\mathbb{M}_i = \mathbb{M}_i \cup \{\mathbf{m}_k\}$
  $r = r + 1$
**return** $\mathbb{M}_i$.

# an optional step for label centric validation; #
# not used in top-N classification.#
**Additional Input:** $\mathbf{m}$. # common mean of data
From Eq. (40), estimate histogram, $h_i(d, \mathbf{m})$
**return** $h_i(d, \mathbf{m})$.

### 6.1 Small Datasets

We begin the evaluation on benchmarks familiar to many researchers. They are:

- MNIST [21]: the classic dataset of handwritten numbers ranging from zero to nine.
- Fashion-MNIST [42]: consists of an assortment of clothing silhouettes grouped into 10 different labels.
- STL-10 [11]: contains images from 10 commonly occurring labels. For example, bird, cat and truck.
- Internet STL-10 [25]: shares the same labels as STL-10; however, training data is crawled from the internet and contains many outliers; testing is conducted on the original STL-10.
- SUN-Small [46]: comprises of the first five labels of SUN-397 [46]. These are: abbey, alley, airport terminal, amusement park and aquarium.
- ASSIRA [13]: contains two, highly ambiguous labels, dog and cat.

MNIST and Fashion-MNIST images are represented by pixel rasterization. This is possible because their objects of interest are simple, appropriately scaled and centered in each image. For the other datasets, images are represented by ResNet-50 [15] features, whose weights were pre-trained on imageNet [12]. The deep-learned feature representation is necessary because objects of interest are complex and unaligned, making pixel based image comparison meaningless.

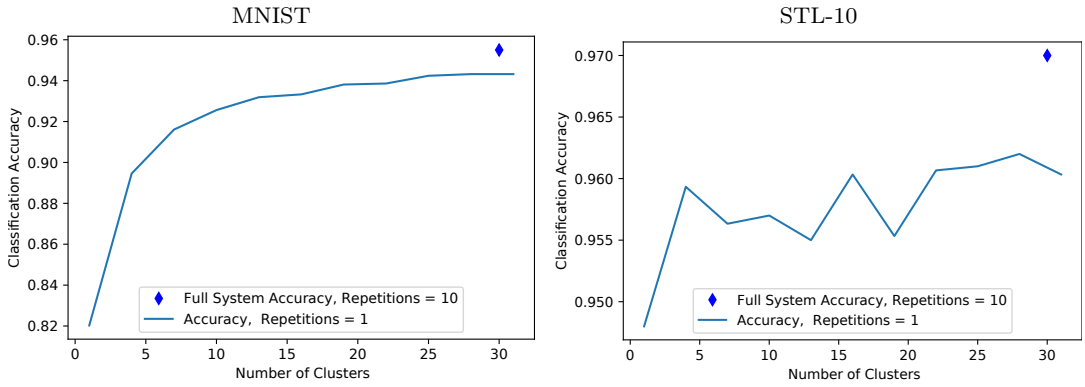



**Fig. 5:** Using more clusters allows for higher accuracy at the expense of larger representation sizes. Large representation sizes are less of a problem for distance classifiers because, if necessary, they can take advantage of approximate nearest-neighbor algorithms, as shown in Fig. 6.

| | | *Average classification accuracy on each dataset* | | | | | |
|---|---|---|---|---|---|---|---|
| | Algorithm | MNIST | Fashion-MNIST | STL-10 | Internet STL-10 | SUN-Small | ASSIRA |
| Discriminatively Trained Classifiers | SVM-linear [16] | 0.912 | 0.810 | 0.944 | 0.880 | 0.926 | **0.988** |
| | SVM-kernel [33] | **0.966** | **0.884** | **0.969** | **0.915** | **0.939** | 0.987 |
| | Soft-Max Layer | 0.897 | 0.810 | 0.967 | 0.897 | 0.926 | **0.988** |
| Generative Naive Bayes Classifiers | GaussNB [45] | 0.556 | 0.586 | 0.926 | 0.864 | 0.877 | 0.981 |
| | MulitNomialNB [38] | **0.837** | **0.655** | **0.941** | **0.895** | **0.913** | **0.983** |
| | ComplementNB [37] | 0.729 | 0.606 | 0.936 | 0.846 | 0.895 | 0.983 |
| Weakly Incremental Learners | OCSVM-N [9,25] | 0.739 | 0.617 | 0.924 | 0.862 | 0.865 | 0.984 |
| | Shell-N [25] | **0.824** | **0.681** | **0.945** | **0.901** | **0.903** | **0.985** |
| Strictly Incremental Learners | OCSVM-R [9] | 0.100 | 0.101 | 0.735 | 0.820 | 0.656 | 0.958 |
| | Shell-R | 0.820 | 0.676 | 0.930 | 0.902 | 0.911 | 0.981 |
| | Mahalanobis [23] | 0.753 | 0.677 | 0.922 | 0.453 | 0.770 | 0.979 |
| | Distance (top-N) | **0.955** | **0.825** | **0.970** | **0.907** | **0.929** | **0.986** |

**Table 1:** Classification accuracy on traditional benchmarks. Distance (top-N) is a strict incremental learner with an accuracy that is more commonly associated with discriminative classifiers.

Classification algorithms are trained on a subset of each dataset and evaluated on an unseen test set. Classification accuracy is reported in Table 1. Algorithms are divided into three categories: discriminative classifiers, generative classifiers and incremental learners. The highest accuracy for each category is bolded.

Discriminative classifiers are represented by linear-SVM [16], kernel-SVM [33] and soft-max classifiers. Unsurprisingly, discriminative classifiers are the best performing category of classifiers.

Generative classifiers are represented by variants of the popular Naive Bayes framework [45,38,37]. They employ joint training to learn the identifying statistics of each semantic. As expected, the generative classifiers are significantly less accurate than discriminative classifiers.

The final set of algorithms are incremental learners which learn each semantic independently. Incremental learning algorithms are further divided into two groups. The first comprises of algorithms that use shell normalization [25] (see Sec. 4.3 for details). Shell normalization significantly enhances accuracy [25]; however, it requires the dataset mean, which is information external to a semantic. We regard algorithms that use shell normalization, to have weakly fulfilled the incremental learning requirements. The second group of incremental learners employ no normalization and are deemed strict incremental learners. As expected, incremental learn-

ers have generally low accuracy, with strict incremental learners being the most inaccurate.[4]

Distance (top-N) inverts the performance hierarchy. Although Distance (top-N) is a strict incremental learner, its accuracy approaches the upper ranks of discriminative classifiers. This is an exciting prospect because Distance (top-N)'s provides significant advantages in training complexity and ease of update.

### 6.2 Training Complexity

Traditional machine learning uses the training step to discover classification constraints. In distance based classification, the constraints are analytically derived prior to training; this significantly reduces training complexity.

Let $N$ denote the number of semantic labels, the training complexity of Distance (Top-N) is $O(N)$ or $O(Nlog(N))$. The $O(N)$ complexity comes about because Distance Classifiers learn each semantic independently. A complexity of $O(Nlog(N))$ is incurred if an additional search tree is constructed [17,18]. Distance (Top-N) has a test complexity of $O(log(N))$ if a search tree is used and $O(N)$ otherwise. Distance Classification's very low training and test complexity makes it scalable to large datasets. In contrast, the training and test complexity of linear-SVM, one of the fastest traditional classifiers, is $O(N^2)$ and $O(N)$ respectively.

Figure 6 shows real-world training and test times on SUN-397 [46]. Distance (Top-N)'s training is clearly faster than linear-SVM's, with the gap growing rapidly as the number of classes increase. Distance (Top-N)'s absolute test time is slower than linear-SVM's; however, Distance (Top-N)'s test time grows more slowly.

### 6.3 Incrementally Learned Large Datasets

The strictly incremental, one-class learners in Sec. 6.1 are almost never employed on large scale datasets because their performance drops rapidly as the number of classes increase. Indeed, the consensus seems to be that strictly incremental learning is impractical and most large scale incremental learners use some replay of past data in their update steps [36,8,41,14,34]. In this paper's definition, such algorithms are weakly incremental.

To ensure efficient updates, weakly incremental try to minimize replay. However, even if one instance per semantic is selected for replay, the update complexity of a weakly incremental learner is at least $O(N)$, where $N$ is the number of previously learned semantics. To manage computational cost, these learners employ simpler neural-networks, like ResNet-18 [15], trading accuracy for speed.

Distance (top-N) provides a strictly incremental learner which does not require replay. As each semantic is learned independently, the update complexity of Distance (top-N) is $O(1)$, or $O(log(N))$ if a tree is created for fast search. The increased update speed makes it practical to use better features, such as ResNet-50 [15] (instead of ResNet-18), which in turn improves accuracy.

The disadvantage of Distance (top-N) is it lacks a mechanism for feature refinement. Instead, Distance (top-N) relies on noise cancellation to accommodate less than ideal features. Nonetheless, the overall system is faster (as shown in Sec. 6.4) and more accurate (as shown in Table 2) than traditional incremental learners. Below, we discuss the results in detail.

*Baseline algorithms [41, 14, 34]:* These are recent, top performing, large scale incremental learners. Baseline algorithms enjoy the benefit of replay, which is not used in distance classification. To further ensure a fair comparison, baseline algorithms are initialized with pre-trained imageNet weights, a setting which we find maximizes their performance.

*ImageNet-1000 [12]:* This is the traditional benchmark for classification algorithms. Distance classification's top-5 accuracy on imageNet is 88%, which is quite close to the 94.7% accuracy of jointly-trained ResNet-50 [15], the upper bound for an incremental learner's performance [41]. ImageNet's test set has many erroneous labels, some of which have been corrected [5]. Evaluated on the new test labels, distance classification's top-5 accuracy improves to 91.7%, while the accuracy of ResNet-50 declines slightly [5], further narrowing the performance gap.

As the ResNet-50 features used in distance classification were learned using imageNet, distance classification's performance on imageNet should not be considered a measure of it's generalizability. Instead, the imageNet evaluation should be interpreted as a test for distance classifier's ability to replace ResNet-50's final soft-max layer. From this perspective, the results on imageNet are promising. To evaluate general classification accuracy, we turn to the next three datasets.

*SUN-397 [43], Food-101 [6], Flower-102 [31]:* These datasets consist of buildings, food and flowers respectively. As these semantic concepts are very different

[4] Our evaluation focuses only on the top one-class learning algorithms for these datasets. Comparisons with other one-class learning algorithms can be found in [25].

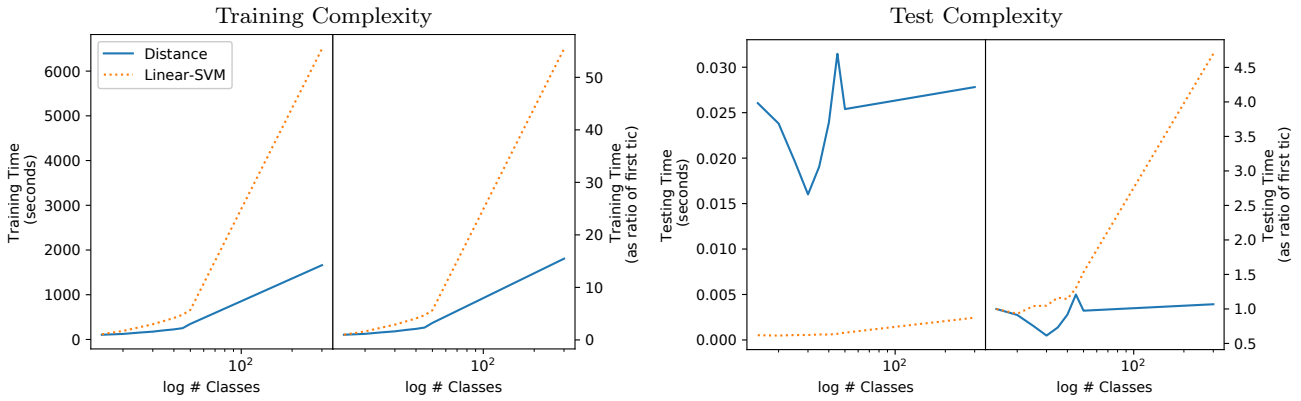



**Fig. 6:** Comparing training and testing speeds with increasing number of classes. Computational time is measured in seconds and as a ratio of the computational time at the first tic mark on the graph (15 classes). Distance Classifiers have a lower **training complexity** than linear-SVM, $O(Nlog(N))$ vs $O(N^2)$, where $N$ is the number of labels. Distance Classifiers also have a lower **test complexity**, $O(log(N))$ vs $O(N)$. Although the Distance Classifier's initial test time is higher than linear-SVM; the test time grows very slowly with increasing number of classes.

from the animal centric semantics of imageNet, the classes can be considered "unseen" by ResNet. These datasets are very challenging and even state-of-the-art incremental learners exhibit significant performance variations, with the top performer on one dataset, sometimes failing completely on a different dataset. In contrast, Distance (Top-N) is consistently the best algorithm or a close second.

### 6.4 Speed Comparisons

Distance (Top-N) can also be trained efficiently, as shown in Table 3. The performance gain being especially noticeable on large datasets like SUN-397. Further, although Distance (Top-N) uses a more complex representation than the soft-max layer of other incremental leaning, its inference time remains low, as shown in Table 4. If additional inference speed is required, the approximate nearest neighbor algorithm, discussed in Sec. 6.2, can be employed.

## 7 Label Centric Validation (Inference)

The top-N classification discussed in Sec. 6 suffices if each feature has one and only one label. However, labels often overlap one another and features may derive from labels outside the training set. Thus, rather than finding the best label for a given feature, label centric validation seeks to estimate the likelihood of a feature belonging to a given label.

### 7.1 Distance Label Validation

As explained in Sec. 3.2 and Fig. 3, in the presence of noise, the raw distance of a feature to a set of means, is a good classification metric but a terrible validation metric. We propose two metrics for label centric validation; the first is based on distance ratios, which are shown in Sec. 4.2 to be noise invariant; the second is based on noise cancelled distance of features to means, derived in Sec. 4.1.

*Ratio Based Validation:* Ratio based validation is based on the intuition that a features should be unusually close to the label-$i$ before it is validated by label-$i$.

We use the same notation as in Sec. 6, $\mathfrak{M}$ is the set of all estimated generator-means (Sec. 5, Algorithm 1) and $\mathbb{M}_i \subset \mathfrak{M}$ is the set of generator means associated with label-$i$.

For convenience, we define the following entities:

$$\begin{aligned} min_i(t) &= \min_{\boldsymbol{\mu}\in\mathbb{M}_i}\{\|(\mathbf{x}(t)-\boldsymbol{\mu}\|^2\}, \\ min\_(t) &= \min_{\boldsymbol{\mu}\in\mathfrak{M}}\{\|(\mathbf{x}(t)-\boldsymbol{\mu}\|^2\}, \\ max\_(t) &= \max_{\boldsymbol{\mu}\in\mathfrak{M}}\{\|(\mathbf{x}(t)-\boldsymbol{\mu}\|^2\}. \end{aligned} \tag{35}$$

$min_i(t)$ is $\mathbf{x}(t)$'s squared distance to the closest generator-mean in label-$i$, a definition repeated from Eq. (33). $min\_(t)$ is $\mathbf{x}(t)$'s squared distance to the closest generator-mean in $\mathfrak{M}$. $max\_(t)$ is $\mathbf{x}(t)$'s squared distance to the further generator-mean in $\mathfrak{M}$.

The small-ratio scores $\mathbf{x}(t)$ by how close it is to label-$i$, relative to $\mathbf{x}(t)$'s closest mean:

$$s_i(t) = \frac{min_i(t)}{min\_(t)}; \tag{36}$$

***Top-5 accuracy (%)***

| | Increment: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Final | Decline |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ImageNet-1000** 100 class / inc | BiC [41] | 94.1 | 92.5 | 89.6 | 89.1 | 85.7 | 83.2 | 80.2 | 77.5 | 75.0 | 73.2 | 20.9 |
| | REMIND [14] | 94.7 | 87.1 | 83.5 | 80.1 | 77.1 | 75.3 | 74.1 | 72.9 | 72.4 | 71.2 | 23.5 |
| | iTAML [34] | 91.5 | 89.0 | 85.7 | 84.0 | 80.1 | 76.7 | 70.2 | 71.0 | 67.9 | 63.2 | 28.3 |
| | Distance | 94.0 | 94.0 | 93.2 | 92.7 | 92.8 | 92.1 | 90.8 | 89.5 | 88.2 | **88.0** | 6.0 |
| | Distance* (relabeled) | 95.6 | 95.6 | 95.1 | 94.7 | 94.1 | 93.2 | 92.4 | 91.8 | 91.3 | 91.7 | 3.9 |
| **SUN-397** 40 class / inc | BiC [41] | 84.1 | 76.7 | 61.7 | 50.4 | 48.3 | 47.2 | 34.2 | 30.3 | 24.3 | 22.6 | 61.5 |
| | REMIND [14] | 92.9 | 88.7 | 85.9 | 83.0 | 81.2 | 77.8 | 77.9 | 74.8 | 73.8 | 72.6 | 20.3 |
| | iTAML [34] | 95.8 | 90.1 | 86.9 | 81.9 | 78.6 | 75.7 | 75.3 | 73.3 | 71.9 | 68.9 | 26.9 |
| | Distance | 99.4 | 97.8 | 95.8 | 94.4 | 92.9 | 92.0 | 90.2 | 88.8 | 88.1 | **87.5** | 11.9 |
| **Food-101** 10 class / inc | BiC [41] | 88.2 | 82.0 | 80.3 | 77.9 | 72.4 | 73.5 | 69.1 | 68.3. | 65.6 | 61.5 | 26.7 |
| | REMIND [14] | 92.3 | 91.1 | 87.6 | 85.2 | 83.6 | 81.0 | 81.1 | 79.4 | 78.2 | 75.9 | 16.4 |
| | iTAML [34] | 98.6 | 97.0 | 95.2 | 94.3 | 92.8 | 91.1 | 89.4 | 87.5 | 86.6 | **84.7** | 13.9 |
| | Distance | 99.0 | 93.0 | 91.0 | 88.8 | 87.7 | 85.8 | 85.3 | 84.5 | 82.9 | 81.1 | 17.9 |
| | | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | Final | | Decline |
| **Flower-102** 6 class / inc | BiC [41] | 92.1 | 65.4 | 53.1 | 39.6 | 32.0 | 38.7 | 31.0 | 30.1 | 35.9 | | 56.2 |
| | REMIND [14] | 100 | 99.0 | 99.4 | 98.8 | 98.7 | 99.1 | 98.9 | 97.4 | 96.8 | | 3.2 |
| | iTAML [34] | 100 | 94.4 | 92.8 | 88.8 | 90.0 | 88.4 | 85.6 | 83.2 | 81.0 | | 19.0 |
| | Distance | 100 | 100 | 100 | 99.5 | 98.7 | 99.0 | 98.1 | 97.3 | **97.1** | | 2.9 |

***Top-1 accuracy (%)***

| | Increment: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Final | Decline |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ImageNet-1000** 100 class / inc | Distance | 76.7 | 76.7 | 74.3 | 73.9 | 74.9 | 73.8 | 71.9 | 70.3 | 68.3 | 67.7 | 9.0 |
| | Distance* (relabeled) | 81.7 | 81.2 | 79.0 | 78.8 | 78.4 | 77.0 | 75.8 | 74.8 | 73.9 | 74.2 | 7.5 |
| **SUN-397** 40 class / inc | REMIND [14] | 71.8 | 66.9 | 61.6 | 57.0 | 54.6 | 49.6 | 51.8 | 47.7 | 47.7 | 46.7 | 25.1 |
| | iTAML [34] | 65.7 | 53.1 | 44.1 | 36.3 | 34.1 | 32.0 | 28.9 | 25.4 | 23.5 | 21.6 | 44.1 |
| | Distance | 86.6 | 81.1 | 74.7 | 71.4 | 68.3 | 66.4 | 64.0 | 61.8 | 61.2 | **60.6** | 26.0 |
| **Food-101** 10 class / inc | REMIND [14] | 66.3 | 64.1 | 61.9 | 58.9 | 57.6 | 58.1 | 55.4 | 54.1 | 50.9 | 50.6 | 15.7 |
| | iTAML [34] | 87.5 | 82.1 | 77.8 | 74.5 | 72.5 | 67.8 | 62.6 | 59.6 | 58.6 | **57.9** | 29.6 |
| | Distance | 74.9 | 65.3 | 65.0 | 63.7 | 61.8 | 59.8 | 60.0 | 59.3 | 57.0 | 55.2 | 19.7 |
| | | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | Final | | Decline |
| **Flower-102** 6 class / inc | REMIND [14] | 100 | 92.1 | 93.6 | 91.7 | 88.6 | 92.3 | 90.8 | 87.7 | 85.3 | | 14.7 |
| | iTAML [34] | 88.1 | 66.4 | 66.5 | 61.5 | 57.9 | 53.7 | 48.2 | 43.4 | 41.0 | | 47.1 |
| | Distance | 100 | 96.2 | 96.5 | 94.6 | 92.8 | 92.5 | 90.7 | 88.6 | **87.6** | | 12.4 |

**Table 2:** Top-5 and Top-1 accuracy (%) of incremental learners. Decline is the accuracy difference between the first and last increment. ImageNet results are duplicated from the original papers. Distance classification's Top-1 accuracy is 11.3% better than REMIND, its closest competitor.

the smallest (and best) possible value for $s_i(t)$ is 1.

The large-ratio scores $\mathbf{x}(t)$ by how close it is to label-$i$ relative to $\mathbf{x}(t)$'s furthest mean:

$$l_i(t) = 1 + \frac{min_i(t)}{max\_(t)}; \tag{37}$$

the smallest (and best) possible value for $l_i(t)$ is 1.

Section 4.2 shows the noise invariance of ratios is limited to similar distances. To have a metric which spans the range of possible distances, the final ratio metric is a composite of both small and large ratios:

$$r_i(t) = s_i(t) \times l_i(t). \tag{38}$$

*Distance Based Validation:* Equation (29) shows that the noise cancelled distance of a feature to a generator-mean is a measure of how closely they are related. Distance based validation algorithmizes this property.

Let $\mathbf{m}$ denote the reference vector used in noise cancellation. As explained in Sec. 4.1, an ideal choice of $\mathbf{m}$ is the mean of the dataset. If this cannot be determined, $\mathbf{m}$ can be set to the mean of a random image collection like Flickr11k [20].

| Dataset | Method | Num. labels learned before last increment | Num. labels to learn in last increment | Training time for last increment (seconds) | Training time per label (seconds) |
|---|---|---|---|---|---|
| **SUN-397** | BiC [41] | 360 | 37 | 66002 | 1784 |
| | REMIND [14] | 360 | 37 | 2452 | 66 |
| | iTAML [34] | 360 | 37 | 73601 | 1989 |
| | Distance | 360 | 37 | **453** | **12.2** |
| **Food-101** | BiC [41] | 90 | 11 | 1524 | 139 |
| | REMIND [14] | 90 | 11 | 1090 | 99 |
| | iTAML [34] | 90 | 11 | 1553 | 141 |
| | Distance | 90 | 11 | **498** | **45.3** |
| **Flower-102** | BiC [41] | 90 | 12 | 627 | 52 |
| | REMIND [14] | 90 | 12 | **19** | **1.6** |
| | iTAML [34] | 90 | 12 | 1248 | 104 |
| | Distance | 90 | 12 | 38 | 3.5 |

**Table 3:** Time to incremental update a representation. Distance classification has a complexity advantage over other algorithms and thus, is notably faster on large datasets like SUN-397.

| | SUN-397 | Food-101 | Flower-102 |
|---|---|---|---|
| Distance Time to classify an image (seconds) | 0.014 | 0.007 | 0.002 |

**Table 4:** Time taken for distance classifier to classify an image through exhaustive testing of all means; the GPU used is a RTX 2080Ti. As classification times are consistently below 0.1 seconds, testing time is not a bottleneck.

From Sec. 4.1, the noise cancelled distance from label-i is denoted:

$$\widetilde{min}_i(t, \mathbf{m}) = \min_{\boldsymbol{\mu} \in \mathbb{M}_i} d^2(\mathbf{x}(t), \boldsymbol{\mu}, \mathbf{m}). \tag{39}$$

To convert distance into a likelihood, we use a (inverse) percentile likelihood function:

$$h_i(d, \mathbf{m}) = \frac{\left| \{t \,|\, t \in \mathcal{T}_i,\, \widetilde{min}_i(t, \mathbf{m}) > d\} \right|}{|\mathcal{T}_i|}. \tag{40}$$

$|\,.\,|$ denotes set cardinality (number of members in a set), $\mathcal{T}_i$ is the set of indices belonging to label-$i$ and $d$ is a squared distance. This makes $h_i(d, \mathbf{m})$, the percentage of instances of label-$i$ whose noise cancelled distance to label-$i$ exceeds $d$. The (inverse) likelihood of $\mathbf{x}(t)$'s membership with label-$i$ is

$$a_i(t, \mathbf{m}) = h_i(\widetilde{min}_i(t, \mathbf{m}), \mathbf{m}), \tag{41}$$

where $a_i(t, \mathbf{m})$ varies between 0 (best) and 1 (worst).

*Final Validation Metric:* The final validation metric fuses the ratio and distance metrics from Eq. (38) and Eq. (41):

$$v_i(t, \mathbf{m}) = 1 - a_i(t, \mathbf{m}) \times r_i(t), \tag{42}$$

where the maximum (and best) score is 1. Empirically, we find $v_i(t, \mathbf{m}) > 0.9$ is a good threshold for accepting $\mathbf{x}(t)$ as a member of label-$i$. The validation process is summarized in Algorithm 2.

Note that Eq. (42) assumes knowledge of the dataset mean, $\mathbf{m}$. This information is external to an individual semantic; thus, algorithms based on Eq. (42) are only weak one-class learners.

A strict incremental learner is possible, if only ratio validation is used:

$$v_i(t) = 1 - r_i(t). \tag{43}$$

While slightly inferior to Eq. (42), this is still a very good validation function.

### 7.2 Experiments

Label centric validation seeks to estimate the likelihood of a feature belonging to a given semantic. If the likelihood estimate is good, there will be a threshold (discoverable from training data) for determining label membership. Rather than evaluating algorithms at specific thresholds, the convention is to evaluate whether a good threshold is possible. One such evaluation metric is AUPRC (area under the precision-recall curve). A high AUPRC implies the existence of a threshold where precision and recall are both high. The maximum (and best) AUPRC is 1.

Algorithms are evaluated on the same datasets used in Sec. 6.1, AUPRC scores are reported in Table 5. Distance Classification is represented by two algorithms: Distance (VS) is a distance classifier trained in a strictly incremental manner; label likelihoods are estimated using Eq. (43). Distance (VW) only weakly satisfies the

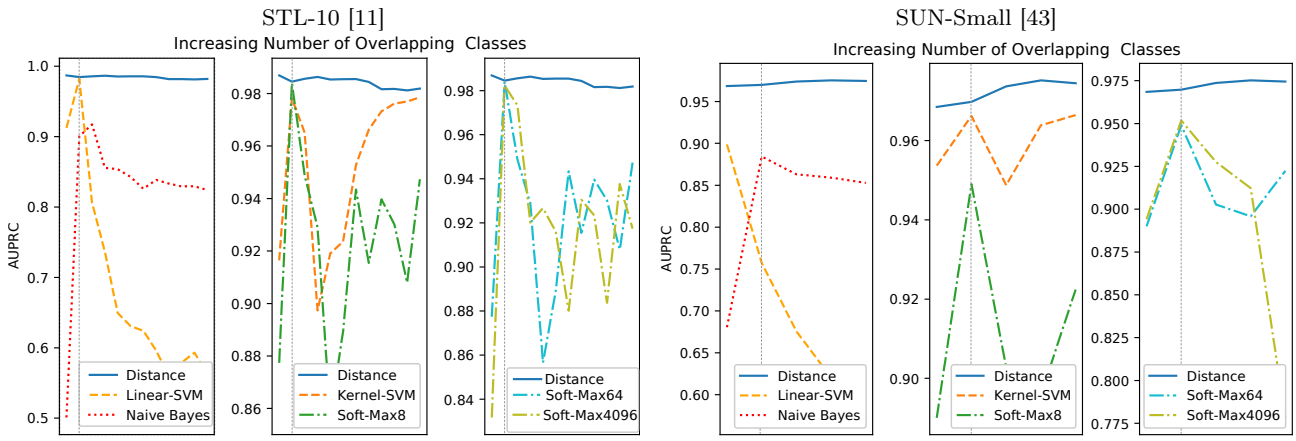



**Fig. 7:** AUPRC as unknown label overlaps increase; soft-max-X refers to training with batch-size X. Traditional classifiers behave erratically when label overlaps are unknown; in contrast, Distance Classifiers are agnostic to overlaps.

**Algorithm 2: Distance Validation**

**Training:**
# $\mathfrak{M}$ is the set of all learned generator-means#
# $\mathfrak{H}$ is the set of all learned percentile functions#
$\mathfrak{M} = \{\}$,
$\mathfrak{H} = \{\}$,
**for** *each class, $i$ :* **do**
  Estimate $\mathbb{M}_i$, $h_i(d, \mathbf{m})$ with Algorithm 1;
  $\mathfrak{M} = \mathfrak{M} \cup \mathbb{M}_i$;
  $\mathfrak{H} = \mathfrak{H} \cup \{h_i(d)\}$;
**return** $\mathfrak{M}$, $\mathfrak{H}$

**Testing:**
# $t$ is the index of the test instance#
# $\mathbf{y}$ is a one-hot array representing $t$'s label membership#
# $\tau = 0.9$ is the acceptance threshold #
**for** *each class, $i$ :* **do**
  Estimate $a_i(t, \mathbf{m})$, using Eq. (41);
  Estimate $r_i(t)$, using Eq. (38);
  $v_i(t, \mathbf{m}) = 1 - a_i(t, \mathbf{m}) \times r_i(t)$.
  **if** $v_i(t) > \tau$ **then**
    $\mathbf{y}[i]$ = True
  **else**
    $\mathbf{y}[i]$ = False
**return** $\mathbf{y}$

incremental learning requirements, label likelihoods are estimated using Eq. (42). Both Distance (VS) and Distance (VW) top their respective categories, with AUPRCs approaching that of discriminative classifiers.

The AUPRC metric used in Table 5 is much stricter than the traditional AUROC used to evaluate likelihood estimates. If measured in terms of AUROC scores, Distance (VW)'s performance on MNIST and Fashion-MNIST are 0.995 and 0.963 respectively. This contrasts with its AUPRC scores of 0.960 and 0.770 respectively. [5]

*Novelty Detection:* Classifiers will eventually encounter novel instances from semantics outside of their training data; in such cases, it is helpful if novel instances could be identified by their consistently low likelihood score across all training labels. To evaluate classifiers on this metric, test data is augmented with an equal number of images, randomly sampled from Flickr11k [20]. An algorithm's ability to identify novel Flickr11k images is quantified using an AUPRC score, that is recorded in Table 6. The experiments show distance classifiers are surprisingly good at identifying novel instances, a task that is difficult for even NAT [28], one of the leading neural networks for STL-10.

Distance classification's strength at novelty detection may be due to its unique approach to learning. Unlike other top performing classifiers, distance classifiers are not trained to minimize a classification loss. This may make them less prone to over-fitting and allow for better generalization on unseen classes.

*Unknown Semantic Overlap:* Another long-standing classification problem is semantic overlap. If semantic overlaps are not identified prior to training, traditional classifiers can be ill-conditioned, leading to erratic performance. To investigate this problem, we design the following experiment:

– A set of $N$, non-overlapping labels are chosen.
– New labels are created by combining pairs of original labels; there are ${}^{N}\mathrm{C}_2$ new labels.

[5] Our evaluation focuses only on the top one-class learning algorithms for these datasets. Comparisons with other one-class learning algorithms can be found in [25].

| | *Average Area Under Precision-Recall Curve* | | | | | |
|---|---|---|---|---|---|---|
| | Algorithm | MNIST | Fashion-MNIST | STL-10 | Internet STL-10 | SUN-Small | ASSIRA |
| Discriminatively Trained Classifiers | SVM-linear [16] | 0.912 | 0.842 | 0.960 | 0.947 | 0.963 | **0.999** |
| | SVM-kernel [33] | **0.978** | **0.903** | 0.980 | 0.933 | 0.967 | **0.999** |
| | Soft-Max Layer | 0.896 | 0.809 | **0.995** | **0.963** | **0.978** | **0.999** |
| Jointly Learned Naive Bayes Classifiers | GaussNB [45] | 0.630 | **0.543** | 0.926 | 0.849 | 0.865 | 0.981 |
| | MulitNomialNB [38] | **0.746** | 0.509 | **0.962** | 0.915 | 0.935 | **0.987** |
| | ComplementNB [37] | 0.679 | 0.503 | 0.985 | **0.942** | **0.957** | **0.987** |
| Weakly Incremental Learners | OCSVM-N [9, 25] | 0.855 | 0.673 | 0.960 | 0.916 | 0.957 | **0.999** |
| | Shell-N [25] | 0.845 | 0.662 | 0.952 | **0.951** | 0.947 | **0.999** |
| | Distance (VW) | **0.960** | **0.770** | **0.984** | **0.951** | **0.969** | **0.999** |
| Strictly Incremental Learners | OCSVM-R [9] | 0.105 | 0.103 | 0.675 | 0.567 | 0.599 | 0.862 |
| | Shell-R [25] | 0.628 | 0.550 | 0.664 | 0.570 | 0.601 | 0.858 |
| | Mahalanobis [23] | 0.736 | 0.600 | 0.770 | 0.494 | 0.657 | 0.943 |
| | Distance (VS) | **0.980** | **0.852** | **0.984** | **0.931** | **0.941** | **0.985** |

**Table 5:** Label centric validation evaluated by AUPRC. Distance Classifiers are comparable to traditional discriminative classifier.

| | *AUPRC* | | | |
|---|---|---|---|---|
| Algorithm | STL-10 | Internet STL-10 | SUN -Small | ASSIRA |
| NAT [28] | 0.875 | - | - | - |
| SVM-linear [16] | 0.898 | 0.882 | 0.818 | 0.846 |
| SVM-kernel [33] | **0.978** | 0.885 | 0.826 | **0.991** |
| Soft-Max Layer | 0.922 | 0.861 | 0.824 | 0.944 |
| MulitNomialNB [38] | 0.613 | 0.626 | 0.597 | 0.564 |
| Distance (VW) | 0.970 | **0.922** | **0.852** | 0.979 |

**Table 6:** Comparing classifier's ability to identify novel instances. Distance classifiers are very effective; their AUPRC scores are consistently the best or a close second best.

– Algorithms are evaluated on their ability to learn progressively larger training sets. The first training set is a subset of the original labels; training sets are progressively expanded until they encompass all the original labels, this point is marked with a vertical dotted line in Fig. 7. The newly created, overlapping labels, are then added to the training set.
– An algorithm's AUPRC scores on each training set is plotted in Fig. 7.

Figure 7 shows that unlike traditional classifiers, distance classifiers are agnostic to semantic overlaps. This stems from distance classifier's ability to represent each semantic independently. As semantics are not defined by their relation to each other, the presence of absence of semantic overlap is no longer cause for concern.

## 8 Conclusion

This paper provides a statistical framework which reduces the problem of image classification to a modified nearest-neighbor algorithm that we term the distance classification. Distance classifiers are sufficiently accurate to act as a viable alternative for to a (trained) neural network's soft-max layer. This increases the generalizability of a pre-trained network as it can be: incrementally updated rapidly and accurately; accommodate huge number of classes; and posses the ability to recognize anomalies events. [6]

**Acknowledgements:** We would like to thank Ng Hongwei of Blackmagic Design for many hours of fruitful discussions; and the Lee Kong Chian foundation for supporting our work.

## References

1. `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`. Accessed: 2022-05-15.
2. Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.
3. Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.
4. Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In

[6] Code is available at: https://www.kind-of-works.com/

*International conference on database theory*, pages 217–235. Springer, 1999.
5. Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.
6. Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Eur. Conf. Comput. Vis.*, pages 446–461. Springer, 2014.
7. Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.
8. Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Eur. Conf. Comput. Vis.*, pages 233–248, 2018.
9. Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *IEEE Int. Conf. Image Process.*, pages 34–37. Citeseer, 2001.
10. Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.
11. Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.
12. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255. Ieee, 2009.
13. Jeremy Elson, John (JD) Douceur, Jon Howell, and Jared Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007.
14. Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *Eur. Conf. Comput. Vis.*, 2019.
15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.
16. Marti A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, July 1998.
17. Ville Hyvönen, Teemu Pitkänen, Sotiris Tasoulis, Elias Jääsaari, Risto Tuomainen, Liang Wang, Jukka Corander, and Teemu Roos. Fast nearest neighbor search through sparse random projections and voting. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 881–888. IEEE, 2016.
18. Elias Jääsaari, Ville Hyvönen, and Teemu Roos. Efficient autotuning of hyperparameters in approximate nearest neighbor search. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, page In press. Springer, 2019.
19. Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, (7):881–892, 2002.
20. Yin-Hsi Kuo, Hsuan-Tien Lin, Wen-Huang Cheng, Yi-Hsuan Yang, and Winston H Hsu. Unsupervised auxiliary visual words discovery for large-scale image object retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 905–912, 2011.
21. Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
22. Antoine Ledent, Rodrigo Alves, Yunwen Lei, and Marius Kloft. Fine-grained generalization analysis of inductive matrix completion. *Advances in Neural Information Processing Systems*, 34:25540–25552, 2021.
23. Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Adv. Neural Inform. Process. Syst.*, pages 7167–7177, 2018.
24. Wen-Yan Lin, Siying Liu, Jian-Huang Lai, and Yasuyuki Matsushita. Dimensionality's blessing: Clustering images by underlying distribution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5784–5793, 2018.
25. Wen-Yan Lin, Siying Liu, Changhao Ren, Ngai-Man Cheung, Hongdong Li, and Yasuyuki Matsushita. Shell theory: A statistical model of reality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.
26. Wen-Yan Lin, Zhonghang Liu, and Siying Liu. Locally varying distance transform for unsupervised visual anomaly detection. *ECCV*, 2022.
27. David G Lowe. Object recognition from local scale-invariant features. In *Int. Conf. Comput. Vis.*, volume 2, pages 1150–1157. Ieee, 1999.
28. Zhichao Lu, Gautam Sreekumar, Erik Goodman, Wolfgang Banzhaf, Kalyanmoy Deb, and Vishnu Naresh Boddeti. Neural architecture transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
29. Panos P Markopoulos, Sandipan Kundu, Shubham Chamadia, and Dimitris A Pados. Efficient l1-norm principal-component analysis via bit flipping. *IEEE Transactions on Signal Processing*, 65(16):4252–4264, 2017.
30. Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Adv. Neural Inform. Process. Syst.*, pages 841–848, 2002.
31. M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 2, pages 1447–1454. IEEE, 2006.
32. Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.
33. John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.
34. Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13588–13597, 2020.
35. Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. In *Adv. Neural Inform. Process. Syst.*, pages 7647–7657, 2019.
36. Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2001–2010, 2017.
37. Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.

38. Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.
39. Gido M. van de Ven, Zhe Li, and Andreas S. Tolias. Class-incremental learning with generative classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3611–3620, June 2021.
40. Lun Wu, Arvind Ganesh, Boxin Shi, Yasuyuki Matsushita, Yongtian Wang, and Yi Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Asian Conference on Computer Vision*, pages 703–717. Springer, 2010.
41. Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 374–382, 2019.
42. Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
43. Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3485–3492. IEEE, 2010.
44. Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *Int. Conf. Learn. Represent.*, 2017.
45. Harry Zhang. Exploring conditions for the optimality of naive bayes. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02):183–198, 2005.
46. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Adv. Neural Inform. Process. Syst.*, pages 487–495, 2014.


## 9 Appendix

The main body focuses on developing the best possible distance classifier. In the appendix, we relate our proposed solution to some popular nearest-neighbor alternatives.

### 9.1 Nearest-Neighbor Classification

Finally, no discussion of distance based classification can be complete without mentioning the nearest-neighbor classifier. This section introduces reference based noise cancelled nearest-neighbor classifier. Figure 8 shows that (similar to our other distance based classifier), noise cancellation greatly improves the nearest neighbor classifier's validation capability. The derivation is provided below.

Let $\mathbf{E}$ be the ideal, common generator of all instances in a dataset and $\mathbf{m}$ be its mean. As $\mathbf{m}$ is constant relative to the process of generating individual instances, from Eq. (3) the distance of all ideal instances

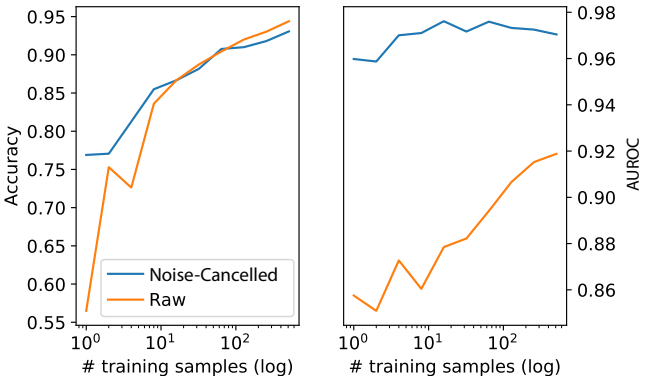



**Fig. 8:** Nearest-neighbor classification on STL-10 [11] with raw and reference based noise cancelled distances. Noise cancelled distances provide significantly better validation, as shown by the AUROC score.

to $\mathbf{m}$ will be a constant, which we denote as $c_{\mathbf{m}}$. i.e. , if $\mathbf{x}_t$, $\mathbf{x}_{t'}$ are two ideal instances

$$a.s. \quad \|\mathbf{x}_t - \mathbf{m}\| = \|\mathbf{x}_{t'} - \mathbf{m}\| = c_{\mathbf{m}}. \tag{44}$$

From Eq. (16), the distance of the noisy features $\mathbf{x}(t)$, $\mathbf{x}(t')$ from $\mathbf{m}$ is:

$$\begin{aligned} a.s. \quad & \|\mathbf{x}(t) - \mathbf{m}\|^2 \\ & \approx \|\mathbf{x}_t - \mathbf{m}\|^2 + \|\mathbf{n}(t)\|^2 = c_{\mathbf{m}} + \|\mathbf{n}(t)\|^2, \\ a.s. \quad & \|\mathbf{x}(t') - \mathbf{m}\|^2 \\ & \approx \|\mathbf{x}_{t'} - \mathbf{m}\|^2 + \|\mathbf{n}(t')\|^2 = c_{\mathbf{m}} + \|\mathbf{n}(t')\|^2. \end{aligned} \tag{45}$$

From Eq. (17), the distance of $\mathbf{x}(t)$ from $\mathbf{x}(t')$ is

$$\|\mathbf{x}(t) - \mathbf{x}(t')\|^2 \approx \|\mathbf{x}_t - \mathbf{x}_{t'}\|^2 + \|\mathbf{n}(t)\|^2 + \|\mathbf{n}(t')\|^2. \tag{46}$$

Thus, combining Eq. (45) and Eq. (46), a noise cancelled nearest-neighbor distance be can be defined as:

$$\begin{aligned} & f^2(\mathbf{x}(t), \mathbf{x}(t'), \mathbf{m}) \\ =& \|\mathbf{x}(t) - \mathbf{x}(t')\|^2 - \|\mathbf{x}(t) - \mathbf{m}\|^2 - \|\mathbf{x}(t') - \mathbf{m}\|^2 \\ \approx& \|\mathbf{x}_t - \mathbf{x}_{t'}\|^2 - 2c_{\mathbf{m}}, \end{aligned} \tag{47}$$

where $f^2(\mathbf{x}(t), \mathbf{x}(t'), \mathbf{m})$ approximates the ideal squared distance with a constant offset, $-2c_{\mathbf{m}}$.

In terms of practical effectiveness, reference based noise canceled nearest-neighbor, is similar to that of standard normalization. However, the ability to achieve a normalization like effect with a different algorithm, helps validate our theory. It also provides a chance to reconsider, the classic algorithm of undergraduate textbooks, from a new perspective.

### 9.2 Distance vs Nearest-Neighbor Classification

Finally, it would be instructive to compare our distance classifier with a traditional nearest neighbor classification algorithm. For this task, we consider both euclidean and cosine distance classifiers with and without normalization / centering. They against our distance classifier in Table 7, which reports both classification accuracy and validation AUPRC.

As predicted, all the nearest-neighbor classification accuracies are respectable, with only minor variations across algorithms. However, there are large differences in AUPRC. For Euclidean nearest-neighbor, normalization significantly improves AUPRC, which shell theory and Sec. 4.3 explain in terms of a noise cancellation effect. A similar improvement occurs when applying the cosine distance to centered data-points. However, we do not offer an analytical explanation because cosine distances are not translational invariant; and thus, cannot be trivially analyzed using shell theory.

Although normalization significantly improves validation AUPRC of traditional nearest-neighbor algorithms, the scores remain significantly below that of our distance classifier. If normalization cannot be employed (such as in the context of strictly incremental learning), our distance classifier will have significantly high validation AUPRC than either of the traditional nearest-neighbor techniques.

| | *Classification Accuracy* | | | | | | |
|---|---|---|---|---|---|---|---|
| Strictly Incremental | | MNIST | Fashion-MNIST | STL-10 | Internet STL-10 | SUN-Small | ASSIRA |
| No | Cos. NN (raw) | 0.968 | **0.852** | 0.952 | 0.887 | 0.906 | 0.982 |
| | Euc. NN (raw) | **0.969** | 0.850 | 0.955 | 0.865 | 0.885 | **0.986** |
| | Dist. Classifier (raw) | 0.955 | 0.825 | **0.970** | **0.907** | **0.929** | **0.986** |
| Yes | Cos. NN (centered) | **0.969** | **0.858** | 0.955 | 0.888 | 0.903 | 0.979 |
| | Euc. NN (normalized) | **0.969** | 0.858 | 0.955 | 0.888 | 0.898 | 0.979 |
| | Dist. Classifier (raw) | 0.955 | 0.825 | **0.970** | **0.907** | **0.929** | **0.986** |
| | *Validation AUPRC* | | | | | | |
| Strictly Incremental | | MNIST | Fashion-MNIST | STL-10 | Internet STL-10 | SUN-Small | ASSIRA |
| No | Cos. NN (raw) | 0.767 | 0.624 | 0.912 | 0.851 | 0.764 | 0.976 |
| | Euc. NN (raw) | 0.784 | 0.673 | 0.764 | 0.589 | 0.562 | 0.915 |
| | Dist. Classifier (VS) | **0.980** | **0.852** | **0.984** | **0.931** | **0.941** | **0.985** |
| Yes | Cos. NN (centered) | 0.923 | 0.667 | 0.915 | 0.855 | 0.825 | 0.958 |
| | Euc. NN (normalized) | 0.923 | 0.667 | 0.922 | 0.855 | 0.805 | 0.958 |
| | Dist. Classifier (VW) | **0.960** | **0.770** | **0.984** | **0.951** | **0.969** | **0.999** |

**Table 7:** Comparing distance classification with traditional nearest-neighbors. Accuracy scores are similar; however, distance classification has higher validation AUPRC. This is especially notable in the full incremental case, where the classifier must be trained without knowledge of the true dataset mean.