Generative Adversarial Minority Oversampling

Presenter: Renato Nascimento

Sankha Subhra Mullick
Shounak Datta
Swagatam Das
Outline

• Introduction

• Generative Adversarial Networks

• Related Works

• Proposed Method

• Experiments and Results
Introduction

• Class imbalance
  • Real-life situations – distribution of examples is skewed – some classes appear more frequently

• Majority class
• Minority class

• Most learning algorithms produce inductive bias favoring the majority class
Introduction

- Class imbalance
- AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions

Sizes action class in the dataset, with colors indicating action types
Introduction

• Class imbalance - Real world applications:

Airbus Ship Detection

Satellite Image Classification and Segmentation with Transfer Learning  
R. G. Nascimento, F. Viana, AIAA Scitech 2020

Wind turbine inspections

Images out of wind turbine gearbox borescope inspections  
[https://onyxinsight.com/2017/03/15/damage-wind-turbine-gearbox-i-looking]
Introduction

• Handling class imbalance
  • Cost tuning (higher costs to minority)
  • Under sampling
  • Oversampling techniques
    • can not be directly applied to end-to-end deep learning systems

  • Majority class
  • Minority class
  • Artificial minority points

• Generative Adversarial Oversampling (end-to-end technique)
Generative Adversarial Networks

Generative Adversarial Nets, Goodfellow et al. in 2014

Two player adversarial game

source: developers.google.com
Generative Adversarial Networks

MNIST

Sample

Generator

Discriminator

Real

Fake

Sample

adapted from: developers.google.com
Generative Adversarial Networks

adapted from: developers.google.com
Generative Adversarial Networks

MNIST

Sample

Real

Generator

Sample

Real

adapted from: developers.google.com
Generative Adversarial Networks

Minimax Loss

\[ \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D \left(x^{(i)}\right) + \log \left(1 - D \left(G \left(z^{(i)}\right)\right)\right) \right] \]

![Diagram of Generative Adversarial Networks](source: developers.google.com)
Generative Adversarial Networks

Minimax Loss

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D \left(G(z^{(i)})\right)\right)$$

Random Input $z^{(i)}$ 

minibatch of $m$ noise samples

source: developers.google.com

Gradient descending
Related Works

• SMOTE: Synthetic Minority Over-sampling Technique (2002)

\[ \mathbf{x}^R \text{ is randomly chosen among the k-nearest neighbors of } \mathbf{x} \]

source: imbalanced-learn.readthedocs.io
Related Works

• Effective data generation for imbalanced learning using conditional generative adversarial networks (cGAN) (2017)

\[
\min_G \max_D V(D, G) = E_D + E_G \\
E_D = E_{x,y \sim p_{data}}(x, y) [\log D(x, y)] \\
E_G = E_{z \sim p_z(z), y \sim p(y)} [\log (1 - D(G(z, y), y))]
\]

Extension of GAN (including additional space Y)

Fig. source: Mohammad Ali Bagheri
Proposed Method

• Three-player adversarial game
  • convex generator $G$
  • classifier network $M$
  • a discriminator $D$

Generator $G$ attempts to fool both $M$ and $D$
Convex Generator

Generate the new points only as convex combinations of the existing points from the minority class in question

\[ G(z|\hat{y}) = g_i(t(z|\hat{y}))^TX_i \]

Cross entropy loss

\[
\min_{G} \max_{M} J(G, M) = \sum_{i \in C} J_i, \quad J_i = (J_{i1} + J_{i2} + J_{i3} + J_{i4})
\]

\[
J_{i1} = P_i \mathbb{E}_{x \sim p^d_i}[\log M_i(x)],
\]

\[
J_{i2} = \sum_{j \in C \setminus \{i\}} P_j \mathbb{E}_{x \sim p^d_j}[\log(1 - M_i(x))],
\]

\[
J_{i3} = (P_c - P_i) \mathbb{E}_{G(z|\hat{y}) \sim p^g_i}[\log M_i(G(z|\hat{y}))], \text{ and,}
\]

\[
J_{i4} = \sum_{j \in C \setminus \{i\}} (P_c - P_j) \mathbb{E}_{G(z|\hat{y}) \sim p^g_j}[\log(1 - M_i(G(z|\hat{j})))]
\]

\[ p^d_i, p^g_i - \text{real and generated class conditional probability distributions of the } i\text{-th class} \]
Convex Generator

\[ G(z|i) = g_i(t(z|i))^T X_i \]

\[ \min_G \max_M J(G, M) = \sum_{i \in C} J_i, \]

Generates points within the convex hull of the samples from the minority

But points may still be placed at locations within the convex hull which do not correspond to the distribution of the intended class
Additional Discriminator

Discriminator ensures that the generated points do not fall outside the actual distribution of the intended minority class.

Three-player minimax game

$$\min_G \max_M \max_D Q(G, M, D) = \sum_{i \in C} Q_i,$$

$$Q_i = (J_{i1} + J_{i2} + J_{i3} + J_{i4} + Q_{i1} + Q_{i2})$$

Cross entropy loss

$$Q_{i1} = P_i \mathbb{E}_{x \sim p^g_i} [\log D(x | i)],$$

and,

$$Q_{i2} = (P_c - P_i) \mathbb{E}_{G(z | i) \sim p^g_i} [\log(1 - D(G(z | i) | i))]$$
Least-Square Loss

\[ L_{i1} = P_i \mathbb{E}_{x \sim p_i^d} [(1 - M_i(x))^2], \]

\[ L_{i2} = \sum_{j \in C \setminus \{i\}} P_j \mathbb{E}_{x \sim p_j^d} [(M_i(x))^2], \]

\[ L_{i3} = (P_c - P_i) \mathbb{E}_{G(z|i) \sim p_i^q} [(1 - M_i(G(z|i)))^2], \]

\[ L_{i4} = \sum_{j \in C \setminus \{i\}} (P_c - P_j) \mathbb{E}_{G(z|j) \sim p_j^q} [(M_i(G(z|j)))^2], \]

\[ L_{i5} = P_i \mathbb{E}_{x \sim p_i^d} [(1 - D(x|i))^2], \]

\[ L_{i6} = (P_c - P_i) \mathbb{E}_{G(z|i) \sim p_i^q} [(D(G(z|i)))^2], \]

\[ L_{i7} = \mathbb{E}_{G(z|i) \sim p_i^q} [(M_i(G(z|i)))^2], \]

\[ L_{i8} = \sum_{j \in C \setminus \{i,c\}} \mathbb{E}_{G(z|j) \sim p_j^q} [(1 - M_i(G(z|j)))^2], \]

\[ L_{i9} = \mathbb{E}_{G(z|i) \sim p_i^q} [(1 - D(G(z|i)))^2]. \]
Datasets

• MNIST and Fashion-MNIST
  • 28×28 grayscale, 10 classes

Evaluation Metrics
Average Class Specific Accuracy (ACSA)
Geometric Mean (GM)

Subset
Imbalance Ratio: 100
(largest/smallest)
Experiments and Results

MNIST and Fashion-MNIST
## Experiments and Results

### MNIST and Fashion-MNIST

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MNIST</th>
<th></th>
<th>Fashion-MNIST</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CE ACSA</td>
<td>GM</td>
<td>CE ACSA</td>
<td>GM</td>
</tr>
<tr>
<td>Baseline CN</td>
<td>0.88±0.01</td>
<td>0.87±0.02</td>
<td>0.85±0.01</td>
<td>0.86±0.01</td>
</tr>
<tr>
<td>SMOTE+CN</td>
<td>0.88±0.02</td>
<td>0.87±0.03</td>
<td>0.89±0.01</td>
<td>0.88±0.01</td>
</tr>
<tr>
<td>Oversample+CN</td>
<td>-</td>
<td>-</td>
<td>0.81±0.01</td>
<td>0.79±0.01</td>
</tr>
<tr>
<td>Augment+CN</td>
<td>-</td>
<td>-</td>
<td>0.82±0.01</td>
<td>0.78±0.01</td>
</tr>
<tr>
<td>DOS</td>
<td>-</td>
<td>-</td>
<td>0.82±0.01</td>
<td>0.79±0.01</td>
</tr>
<tr>
<td>cGAN / cDCGAN+CN</td>
<td>0.88±0.01</td>
<td>0.87±0.01</td>
<td>0.88±0.01</td>
<td>0.81±0.02</td>
</tr>
<tr>
<td>cG+CN</td>
<td>0.86±0.03</td>
<td>0.85±0.02</td>
<td>0.85±0.03</td>
<td>0.79±0.02</td>
</tr>
<tr>
<td>cG+D+CN</td>
<td>0.85±0.02</td>
<td>0.85±0.01</td>
<td>0.82±0.02</td>
<td>0.79±0.01</td>
</tr>
<tr>
<td>GAMO\D (Ours)</td>
<td>0.87±0.01</td>
<td>0.86±0.01</td>
<td>0.81±0.01</td>
<td>0.80±0.01</td>
</tr>
<tr>
<td>GAMO (Ours)</td>
<td>0.89±0.01</td>
<td>0.88±0.01</td>
<td>0.91±0.01</td>
<td>0.90±0.01</td>
</tr>
</tbody>
</table>

Average Class Specific Accuracy (ACSA)
Geometric Mean (GM)
Datasets

- Cifar10 and SVHN
  - 32×32 RGB, 10 classes

**Subset**
Imbalance Ratio: 56.25 (largest/smallest)

**Evaluation Metrics**
- Average Class Specific Accuracy (ACSA)
- Geometric Mean (GM)

Cifar10 Sample

SVHN Sample

Label Index
## Experiments and Results

CIFAR10 and SVHN – Subset Imbalance Ratio of 56.25 (largest to smallest class)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CIFAR10</th>
<th></th>
<th>SVHN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACSA</td>
<td>GM</td>
<td>ACSA</td>
<td>GM</td>
</tr>
<tr>
<td>Baseline CN</td>
<td>0.45±0.01</td>
<td>0.37±0.01</td>
<td>0.74±0.01</td>
<td>0.73±0.01</td>
</tr>
<tr>
<td>SMOTE+CN</td>
<td>0.46±0.02</td>
<td>0.4±0.02</td>
<td>0.75±0.01</td>
<td>0.73±0.02</td>
</tr>
<tr>
<td>Oversample+CN</td>
<td>0.44±0.02</td>
<td>0.37±0.03</td>
<td>0.74±0.02</td>
<td>0.73±0.02</td>
</tr>
<tr>
<td>Augment+CN</td>
<td>0.47±0.01</td>
<td>0.39±0.02</td>
<td>0.69±0.01</td>
<td>0.63±0.01</td>
</tr>
<tr>
<td>cDCGAN+CN</td>
<td>0.42±0.02</td>
<td>0.32±0.03</td>
<td>0.69±0.01</td>
<td>0.66±0.02</td>
</tr>
<tr>
<td>DOS</td>
<td>0.46±0.02</td>
<td>0.37±0.01</td>
<td>0.71±0.02</td>
<td>0.68±0.01</td>
</tr>
<tr>
<td>GAMO \ D (Ours)</td>
<td>0.47±0.01</td>
<td>0.40±0.01</td>
<td>0.75±0.01</td>
<td>0.75±0.02</td>
</tr>
<tr>
<td>GAMO (Ours)</td>
<td><strong>0.49±0.01</strong></td>
<td><strong>0.43±0.02</strong></td>
<td><strong>0.76±0.01</strong></td>
<td><strong>0.75±0.02</strong></td>
</tr>
</tbody>
</table>

Average Class Specific Accuracy (ACSA)
Geometric Mean (GM)
Datasets - higher resolution

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th>Classes</th>
<th>Small subset</th>
<th>Large subset</th>
<th>Imbalance Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>CelebA</td>
<td>64\times64 RGB</td>
<td>5</td>
<td>15000, 1500, 750, 300, and 150</td>
<td>28000, 4000, 3000, 1500, and 750</td>
<td>100, 37.33</td>
</tr>
<tr>
<td>LSUN</td>
<td>224\times224 RGB</td>
<td>5</td>
<td>15000, 1500, 750, 300, and 150</td>
<td>50000, 5000, 3000, 1500, and 750</td>
<td>100, 66.67</td>
</tr>
</tbody>
</table>
# Experiments and Results

**CelebA (64x64):** Small set (IR 100), Large set (IR 37.33) and **LSUN (224x224):** Small set (IR 100), Large set (IR 66.67)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CelebA-Small</th>
<th></th>
<th></th>
<th>CelebA-Large</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>During Training</td>
<td>During Testing</td>
<td></td>
<td>During Training</td>
<td>During Testing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACSA</td>
<td>GM</td>
<td>ACSA</td>
<td>GM</td>
<td>ACSA</td>
<td>GM</td>
</tr>
<tr>
<td>Baseline CN</td>
<td>0.91±0.01</td>
<td>0.91±0.01</td>
<td>0.59±0.01</td>
<td>0.45±0.04</td>
<td>0.93±0.01</td>
<td>0.92±0.01</td>
</tr>
<tr>
<td>SMOTE+CN</td>
<td>0.99±0.01</td>
<td>0.99±0.01</td>
<td>0.62±0.02</td>
<td>0.48±0.03</td>
<td>0.99±0.01</td>
<td>0.99±0.01</td>
</tr>
<tr>
<td>Oversample+CN</td>
<td>0.99±0.01</td>
<td>0.99±0.01</td>
<td>0.59±0.03</td>
<td>0.39±0.04</td>
<td>0.98±0.01</td>
<td>0.98±0.02</td>
</tr>
<tr>
<td>Augment+CN</td>
<td>0.74±0.06</td>
<td>0.70±0.09</td>
<td>0.62±0.05</td>
<td>0.47±0.08</td>
<td>0.82±0.01</td>
<td>0.79±0.01</td>
</tr>
<tr>
<td>cDCGAN+CN</td>
<td>0.86±0.01</td>
<td>0.84±0.01</td>
<td>0.59±0.01</td>
<td>0.36±0.02</td>
<td>0.87±0.01</td>
<td>0.86±0.01</td>
</tr>
<tr>
<td>DOS</td>
<td>0.82±0.03</td>
<td>0.80±0.02</td>
<td>0.61±0.01</td>
<td>0.48±0.02</td>
<td>0.84±0.01</td>
<td>0.83±0.02</td>
</tr>
<tr>
<td>GAMO (Ours)</td>
<td>0.92±0.01</td>
<td>0.91±0.01</td>
<td><strong>0.66±0.01</strong></td>
<td><strong>0.54±0.02</strong></td>
<td>0.91±0.01</td>
<td><strong>0.75±0.01</strong></td>
</tr>
</tbody>
</table>

|                        | LSUN-Small    |                  |                  | LSUN-Large    |                  |                  |
|                        | During Training | During Testing |                  | During Training | During Testing |                  |
|                        | ACSA          | GM               | ACSA             | GM            | ACSA             | GM               |
| Baseline CN            | 0.90±0.01     | 0.89±0.01        | 0.50±0.01        | 0.28±0.05     | 0.87±0.01        | 0.87±0.01        |
| SMOTE+CN               | 0.99±0.02     | 0.99±0.02        | 0.50±0.01        | 0.40±0.02     | 0.98±0.01        | 0.98±0.01        |
| Oversample+CN          | 0.99±0.01     | 0.99±0.01        | 0.52±0.01        | 0.43±0.02     | 0.98±0.01        | 0.98±0.01        |
| Augment+CN             | 0.67±0.06     | 0.64±0.09        | 0.54±0.03        | 0.45±0.07     | 0.70±0.03        | 0.65±0.03        |
| cDCGAN+CN              | 0.80±0.02     | 0.79±0.02        | 0.53±0.02        | 0.43±0.03     | 0.81±0.02        | 0.80±0.02        |
| DOS                    | 0.78±0.03     | 0.76±0.02        | 0.54±0.02        | 0.44±0.02     | 0.79±0.02        | 0.77±0.02        |
| GAMO (Ours)            | 0.93±0.01     | 0.93±0.01        | **0.57±0.01**    | **0.50±0.02** | 0.80±0.01        | **0.70±0.02**    |
Some application may require that actual samples be generated by oversampling to form an artificially balanced dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GAMO2pix (Ours)</th>
<th>cDCGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fashion-MNIST</td>
<td>0.75±0.03</td>
<td>5.57±0.03</td>
</tr>
<tr>
<td>SVHN</td>
<td>0.17±0.02</td>
<td>0.59±0.04</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>1.59±0.03</td>
<td>2.96±0.03</td>
</tr>
<tr>
<td>CelebA-Small</td>
<td>11.13±0.04</td>
<td>15.12±0.05</td>
</tr>
</tbody>
</table>
Questions?

Generative Adversarial Minority Oversampling

Sankha Subhra Mullick
Shounak Datta
Swagatam Das

github.com/SankhaSubhra/GAMO

ICCV 2019