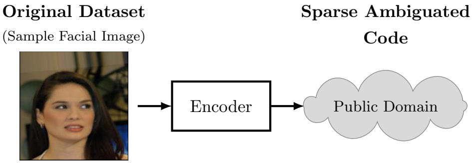
Privacy-Preserving Image Sharing via Sparsifying Layers on Convolutional Groups

Sohrab Ferdowsi, Behrooz Razeghi, Taras Holotyak,
Flavio P. Calmon, Slava Voloshynovskiy

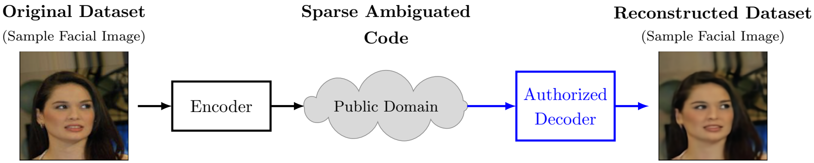
University of Geneva, Harvard University

ICASSP'20 (virtual presentation)
May 2020

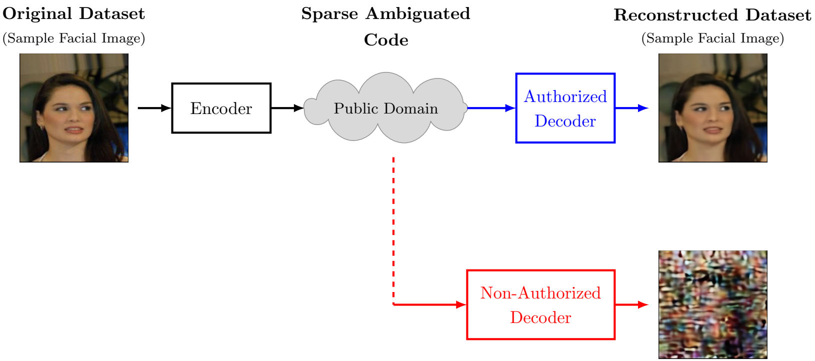
Motivation


Original Dataset
(Sample Facial Image)
Sparse Ambiguated Code
Encoder
Public Domain

Motivation


Original Dataset
(Sample Facial Image)
Encoder
Sparse Ambiguated Code
Public Domain
Authorized Decoder
Reconstructed Dataset
(Sample Facial Image)

Motivation


Original Dataset
(Sample Facial Image)
Encoder
Sparse Ambiguated Code
Public Domain
Authorized Decoder
Reconstructed Dataset
(Sample Facial Image)
Non-Authorized Decoder

Outline

Overview

Introduction

General questions:

- How to provide utility from data, while respecting the privacy of individuals?
- How to design a practical multi-party data-sharing mechanism by revealing the content only to authorized parties?

Common scenarios:

- A busy hospital records large volumes of privacy-sensitive data and should share it selectively across clinics, wards, laboratories, ...
- Data is proprietary and should be shared only to clients who pay for it.

Important challenges:

- Efficient data sharing may inadvertently reveal it to unsafe hands.
- Securing storage and communication is very expensive for large volumes of data.

General overview

Main solution: Instead of strict security, ambiguate the images

Classic solutions: image obfuscation

- Ambiguate (obfuscate) some parts of the images to make the private content non-discernible.
- Considering only single images and not the whole database.
- Ad-hoc manipulation and hence non-efficient
- Prone to many attacks.

Our solution: representation learning

- Judiciously add ambiguation noise to the representations.
- Consider the whole database and let the network learn what to ambiguate.
- Optimal trade-offs for utility-privacy.
- The disambiguation keys very compact yet highly effective.
- Many versatile multi-party scenarios possible.

Formal statement

Problem formulation:

- Data $\mathbf{x} \in \mathcal{X}$, as the pair $(\mathbf{u}_p, \mathbf{u}_s)$.
- Representation size of $\mathbf{u}_s$ be much smaller than that of $\mathbf{u}_p$.
- Guessing cost of the data only using the public portion, i.e., $\mathbf{x}|\mathbf{u}_s$ should be exponential, while its recovery provided both parts, i.e., $\mathbf{x}|\mathbf{u}_s, \mathbf{u}_p$ should be linear.

Scheme overview:

1. *Neural network training:* Objective is to learn compact codes.
2. *Data owner encoding and ambiguating:* Data owner sparsely encodes images. The support of these codes are then kept secure and shared with their corresponding authorized parties. The codes are then ambiguated and shared with public.
3. *Data users/parties disambiguating their content:* Trusted parties use ambiguation maps as the key to unlock their access-granted content from within the public ambiguated database.

Different phases

Method

- *Training Phase:*
  - Train L encoders $\mathbf{z}^{[l]} = \mathrm{Enc}^{[l]}(\mathbf{x}), \forall l \in [L]$.
  - Reconstruct as $\hat{\mathbf{x}}^{[l]} = \mathrm{Dec}^{[l]}[\mathbf{z}^{[l]}]$.
- *Sharing Phase:*
  - Secured part: $\mathbf{u}_s^{[l]} = \text{supp}\left(\mathbf{z}^{[l]}\right)$.
  - Shared part: $\mathbf{u}_p^{[l]} = A\left(\mathbf{z}^{[l]}\right) = \mathbf{z}^{[l]} \oplus \mathbf{n}_{\text{supp}}$.
  - Sparsity $\|\mathbf{u}_p^{[l]}\|_0 = k + k_n = k'$.
- *Reconstruction Phase:*
  - : Service provider reconstructs the data as: $\hat{\mathbf{x}}^{[l]} = \mathrm{Dec}^{[l]}(\mathbf{z}^{[l]} \mid \mathbf{u}_s^{[l]})$.
  - Storage cost of compressed image: $\log_2 \binom{m}{k}^L \simeq mL \times H_2\left(\frac{k}{m}\right)$.
  - Storage cost of the key: $\log_2 \binom{k'}{k}^L \simeq k'L \times H_2\left(\frac{k}{k'}\right)$.
  - Attacker should make $\binom{k'}{k}$ guesses.

Overview

Relation to image compression

- In order to be useful, the ambihuation keys should be much more compact than the original images.
- Tightly related to image compression, however:
  - No bit-manipulation scheme within classical codecs (JPEG, JPEG2000, ..).
  - Non adaptive to possibly very specialized domains.
  - An adversary can infer some content by analyzing the statistical behaviour of activities of similar images.

The need for learned compression:

- A learned approach tries to maximally spread the activations within all coefficients.
- This leaves no trace of the active content.
- Moreover, this provides better compression-fidelity trade-offs than the fixed codecs.
- The literature of "learned compression" recently being active.
- A technical difficulty exists, however: lack of means to impose sparsity.

Imposing sparisty on the activations of CNNs

- Sparsity well-motivated and present in signal processing, yet quite absent in deep learning.

Two main technical challenges:

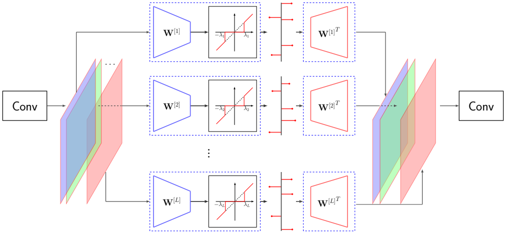
1. Usual non-linearities (ReLu, sigoid, tanh, ..) do not promote sparsity.
2. For large images, we should avoid large fully-connected layers that are prone to over-fitting.

Our main technical contributions:

1. To propose “top-k” non-linearity capable of achieving high sparsity levels with fast training.
2. To propose “grouped linear layers” to promote sparsity on very large images with few parameters.

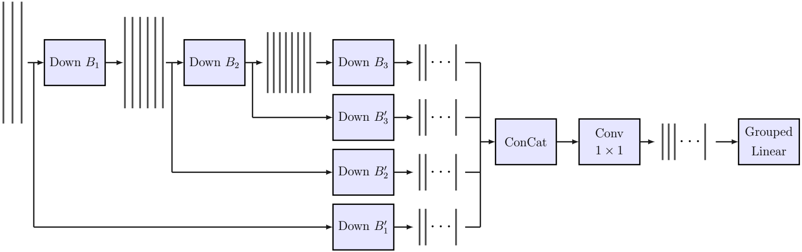
Grouped linear layers

- Convolutional feature maps cannot be directly sparsified. We need fully-connected layers.
- This requires huge matrix multiplication. To avoid this, we use them in "grouped" order.
- By-product: Grouped learning encourages separation of image attributes to different code-maps.



General network architecture

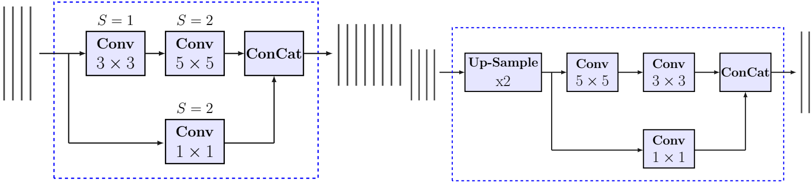
Encoder network



- Each network uses 6 conv blocks with down/up-sampling rations $[1, 2, 1, 2, 1, 2]$.
- 1×1 convolutions used to generate arbitrary number of code-maps.
- Grouped linear layer is used at the bottleneck between the encoder and decoder.
- Decoder network is symmetrical, but uses up-sampling blocks.

Convolutional blocks

- Down-sampling convolutional blocks use depth-wise separation and grouped convs for better efficiency.
- Skip-connections are concatenated (rather than addition) to further separate image attributes to code-maps.



Down Convolutional Block

Up Convolutional Block

Overview

Experimental setup

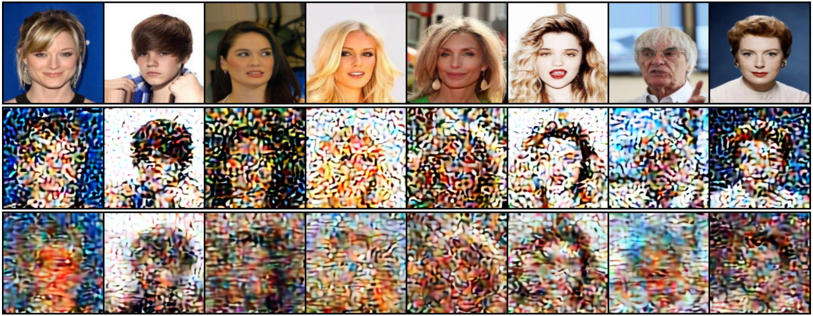
- Our implementation is online at https://github.com/sssohrab/sparsifying_groups_imAmbiguation
- CelebA database of $200K$ images of 128×128.
- 40 epochs, BCE loss, Adam in PyTorch.
- $L = 20$ code-maps of $m = 512$ dimensions, with $k = 128$ true and 128 fake non-zeros.

Quantitative performance:

	Recon. $(k = 64)$ rate $= 0.0845$	Recon. $(k = 128)$ rate $= 0.1690$	JPEG $\text{rate} = 0.1830$	ambiguated $(k' = 256)$	rand. guess $(k' = 256)$
PSNR	28.66	30.75	22.40	12.00	12.31
SSIM	0.92	0.95	0.76	0.24	0.25

Visual performance: authorized users

Visual performance: unauthorized users



Overview

Conclusions

- Proposed a privacy-aware and practical image-sharing scheme for large-scales.
- Learned compression as a solution to provide compact and un-correlated codes adapted to the data.
- Sparsity a useful concept to adopt also in deep learning with many potential applications.
- "top-k" non-linearity is capable of achieving highly sparse codes without slowing down training.
- Grouped linear layer promotes sparsity and feature independence without the hassles of fully-connected layers.
- Future work: to investigate different attacks to this system using adversarial training.