



Towards Self-Adaptive Metric Learning On the Fly

村本尚生

筑波大学 Y-lab



書誌情報

Yang Gao, Yi-Fan Li, Swarup Chandra, Latifur Khan, and Bhavani Thuraisingham. 2019. Towards Self-Adaptive Metric Learning On the Fly. In Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13– 17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 11 pages.
<https://doi.org/10.1145/3308558.3313503>



概要

- 新しい Online Metric Learning フレームワーク, OAHU の提案.
- 実世界の画像分類, 顔認証, 画像検索タスクで最先端のベースラインと比較して, ベストパフォーマンスを出した.
- また, 入力の制約の質に対してよりロバスト性を得た.



研究領域 - Metric Learning

- **Similarity metrics** は，多くの大規模な実世界のアプリケーション (例：画像検索) のパフォーマンスを大幅に向上させる．
 - これらのアプリケーションは，モバイルアプリでのマルチメディア検索エンジンでの画像推薦などに広く使用される．
- 良質な画像 **similarity metrics** を学ぶためにスケーラブル・高性能な技術はますます重要に．



先行研究

- 制約 (constraints) から similarity metric を学習する, ニューラルネットワークを適用した様々な Online Metric Learning (OML) 手法が提案されている.
- これらの制約はユーザのクリックから収集・構築され画像間の類似・非類似を示している



先行研究の利点

- オフライン metric learning アプローチと比較して、OML アルゴリズムは、学習タスクの前に訓練データ全体を利用可能にする必要がないため、Web アプリケーションに適している。



先行研究の欠点 (1/2)

- 既存の **OML** アルゴリズムのほとんどは，凸最適化問題を解くことにより，事前に選択された線形 **metric** (マハラノビス距離など) を学習するよう設計されている.
- これらの単純な **metrics** は画像検索などの複雑なアプリケーションのインスタンス間の非線形セマンティックの類似性をキャプチャできない.



先行研究の欠点 (2/2)

- 既存の **OML** のパフォーマンスは，訓練データの制約に強く依存する.
- 既存の **OML** アプローチでは，実際のアプリケーションでの制約の使用率が低くなり，パフォーマンスが低下する可能性がある.



研究目的

- 制約を完全に利用し， ANN-based metric を学習，さらにモデルを動的に適応できる新しいフレームワークの提案.
- Adaptive-Bound Triplet Loss (ABTL) を提案.
- metric の重み・metric model のパラメータを更新するための新しい Adaptive Hedge Update (AHU) の導入.
- 提案手法を， Online metric learning with Adaptive Hedge Update (OAHU) と呼ぶ.



この論文の寄与

- 新しい OML フレームワーク OAHU の提案.
- 実世界の画像分類, 顔認証, 画像検索タスクで OAHU を経験的に評価し, その結果を既存の最先端の OML アルゴリズムと比較する.



OAHU Overview (1/3)

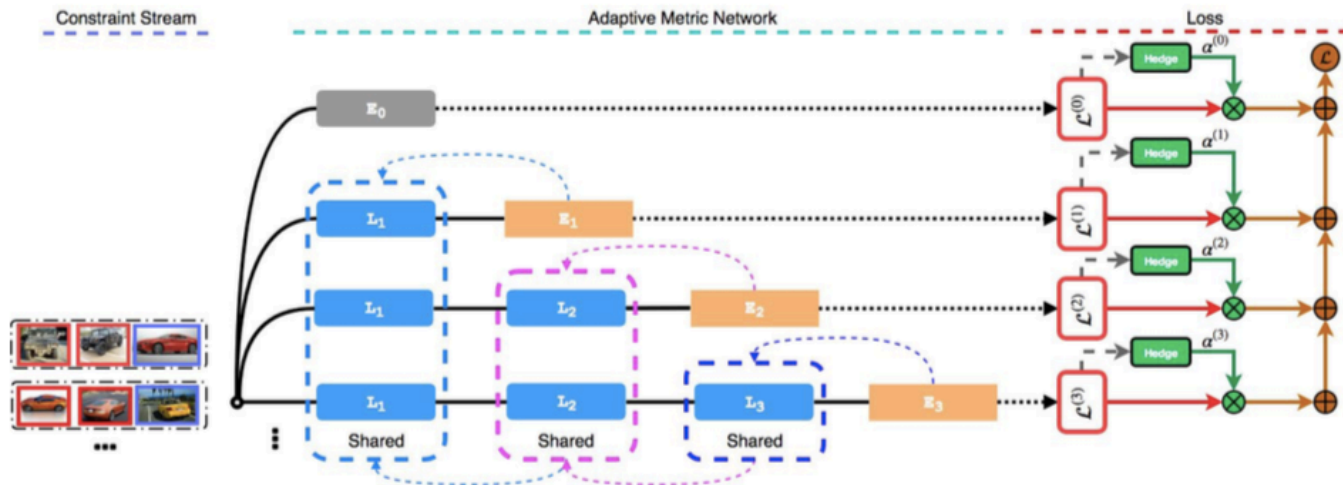


Figure 1: Overview of the OAHU. The chromatic dashed arrows represent update contributions and directions. Each $L_i \in \{L_1, L_2, \dots\}$ represents a linear transformation layer followed by a ReLU activation. $\{E_0, E_1, \dots\}$ are the embedding layers connected to corresponding input or hidden layers. Note that E_0 here represents a linear metric model, i.e., a linear transformation from the input feature space to the embedding space.



OAHU Overview (2/3)

- 入力の制約に基づく **metric function** の適切な複雑さを学ぶために効果的な深さを自動的に適応させる.
- L 個の隠れ層を持つニューラルネットを考えると, 入力層とそれぞれの隠れ層に独立した **embedding** 層 (類似インスタンスを近くに, 非類似は遠くに表す空間を表現) をつなげる.



OAHU Overview (3/3)

- 全ての triplets $(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)$ に対して, l_{th} metric model から metric embedding を取得.
- metric embedding に基づき, 評価された類似・非類似誤差による local 誤差 $L^{(l)}$ を生成する.
- Overall 誤差は,

$$\mathcal{L}_{\text{overall}}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) = \sum_{l=0}^L \alpha^{(l)} \cdot \mathcal{L}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)$$



Adaptive-Bound Triplet Loss (1/3)

- 一般的な triplet loss :

$$L(x_t, x_t^+, x_t^-) = \max\{0, b + d(x_t, x_t^+) - d(x_t, x_t^-)\}$$

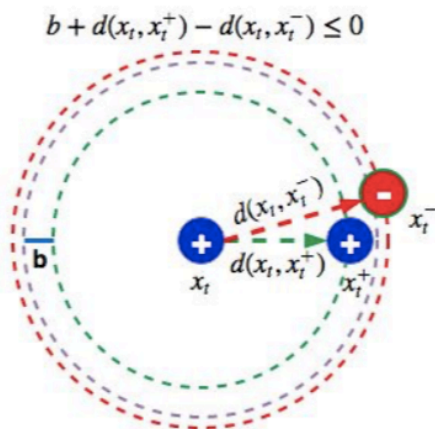


Figure 2: A failure case of triplet loss.

- margin はデータ依存
- モデルが向上するためには failure case を避ける.
- そのためには大きな margin が必要.



Adaptive-Bound Triplet Loss (2/3)

- l_{th} metric model $E_l \in \{E_0, \dots, E_L\}$ での distance measure $D^{(l)}(x_i, x_j)$ は以下のように定義される.

$$D^{(l)}(\mathbf{x}_i, \mathbf{x}_j) = \|f^{(l)}(\mathbf{x}_i) - f^{(l)}(\mathbf{x}_j)\|_2, \quad \forall l = 0, 1, 2, \dots, L$$

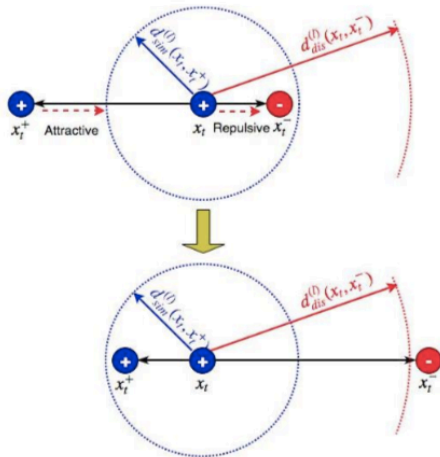


Figure 3: Schematic illustration of ABTL.

$$\begin{cases} D_{\text{update}}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) \leq d_{\text{sim}}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) \\ D_{\text{update}}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) \geq d_{\text{dis}}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) \end{cases}$$



Adaptive-Bound Triplet Loss (2/3)

- Attractive loss L_{attr}^l repulsive loss L_{rep}^l は,

$$\begin{cases} \mathcal{L}_{attr}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) = \max \left\{ 0, \frac{1}{2-c_1} D_{orig}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) - \frac{c_1}{2-c_1} \right\} \\ \mathcal{L}_{rep}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) = \max \left\{ 0, \frac{-1}{c_2} D_{orig}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) + 1 \right\} \end{cases}$$

where $c_1 = d_{sim}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+)$ and $c_2 = d_{dis}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-)$.

と表され,



Adaptive-Bound Triplet Loss (3/3)

- Local loss は, L_{attr}^l と L_{rep}^l を平均したもので,

$$\mathcal{L}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) = \frac{1}{2} \left(\mathcal{L}_{attr}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) + \mathcal{L}_{rep}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) \right)$$

と表される.



ABTLの利点

- Triplet loss と同様に，同時に類似・非類似誤差を評価する．
- 全ての制約がモデルの更新に寄与できる．
(failure caseを削除し，制約の使用率をあげる)



Adaptive Hedge Update (1/3)

- ABTLを適用すると， OAHU の Overall loss は，

$$\begin{aligned}\mathcal{L}_{overall}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) &= \sum_{l=0}^L \alpha^{(l)} \cdot \mathcal{L}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) \\ &= \sum_{l=0}^L \frac{\alpha^{(l)}}{2} \left(\mathcal{L}_{attr}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) + \mathcal{L}_{rep}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) \right)\end{aligned}$$

- Hedge アルゴリズムを用いて， $\alpha^{(l)}$ を学ぶ方法を提案.



Adaptive Hedge Update (2/3)

- 全ての重みを均一な分布に初期化.
- $\alpha^{(l)}$ は, local loss に基づいて更新:

$$\alpha_{t+1}^{(l)} \leftarrow \begin{cases} \alpha_t^{(l)} \beta^{\mathcal{L}^{(l)}} & \beta^{\min_l \mathcal{L}^{(l)}} \log \mathcal{L}^{(l)} > \beta - 1 \\ \alpha_t^{(l)} [1 - (1 - \beta) \mathcal{L}^{(l)}] & \text{otherwise} \end{cases}$$

- $\beta \in (0, 1)$ は discount factor.
- それぞれのステップで $\alpha^{(l)}$ が最大化されるように更新.



Adaptive Hedge Update (3/3)

- パラメタは, Online Gradient Descent (OGD) で更新される.

$$\begin{aligned}\Theta_{t+1}^{(l)} &\leftarrow \Theta_t^{(l)} - \eta \nabla_{\Theta_t^{(l)}} \mathcal{L}_{overall}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) \\ &= \Theta_t^{(l)} - \eta \frac{\alpha^{(l)}}{2} \left(\nabla_{\Theta_t^{(l)}} \mathcal{L}_{attr}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+) + \nabla_{\Theta_t^{(l)}} \mathcal{L}_{rep}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-) \right)\end{aligned}\quad (21)$$

$$\begin{aligned}W_{t+1}^{(l)} &\leftarrow W_t^{(l)} - \eta \nabla_{W_t^{(l)}} \mathcal{L}_{overall}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-) \\ &= W_t^{(l)} - \eta \sum_{j=l}^L \frac{\alpha^{(j)}}{2} \left(\nabla_{W_t^{(l)}} \mathcal{L}_{attr}^{(j)}(\mathbf{x}_t, \mathbf{x}_t^+) + \nabla_{W_t^{(l)}} \mathcal{L}_{rep}^{(j)}(\mathbf{x}_t, \mathbf{x}_t^-) \right)\end{aligned}\quad (22)$$



OAHU アルゴリズム

Algorithm 1 OAHU: Online Metric Learning with Adaptive Hedge Update

Require: Discount Factor $\beta \in (0, 1)$; Smooth Factor s ; Control Parameter $\tau \in (0, \frac{2}{3})$; Learning rate η ; A randomly initialized ANN with L hidden layers that is parameterized by $\Theta^{(l)}, W^{(l)}$ and $\alpha^{(l)}$.

Ensure: $\alpha^{(l)}, \Theta^{(l)}$ and $W^{(l)}$

- 1: Initialize $\alpha^{(l)} = \frac{1}{L+1}, \forall l = 0, 1, \dots, L$
 - 2: **for** $t = 1, 2, \dots, T$ **do**
 - 3: Receive a triplet constraint $(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-)$
 - 4: Transform and retrieve the metric embedding $f^{(l)}(\mathbf{x}_t)$, $f^{(l)}(\mathbf{x}_t^+)$ and $f^{(l)}(\mathbf{x}_t^-)$ from each model E_l .
 - 5: Evaluate $\mathcal{L}_{attr}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+)$ and $\mathcal{L}_{rep}^{(l)}(\mathbf{x}_t, \mathbf{x}_t^-)$ (Eq. 12), $\forall l = 0, 1, \dots, L$.
 - 6: Compute $L^{(l)}(\mathbf{x}_t, \mathbf{x}_t^+, \mathbf{x}_t^-), \forall l = 0, 1, \dots, L$ as per Eq. 13.
 - 7: Update $\Theta_{t+1}^{(l)}, \forall l = 0, 1, 2, \dots, L$ as per Eq. 21.
 - 8: Update $W_{t+1}^{(l)}, \forall l = 0, 1, 2, \dots, L$ as per Eq. 22.
 - 9: Update $\alpha_{t+1}^{(l)}, \forall l = 0, 1, 2, \dots, L$ as per Eq. 20.
 - 10: $\alpha_{t+1}^{(l)} = \max(\alpha_{t+1}^{(l)}, \frac{s}{L+1}), \forall l = 0, 1, 2, \dots, L$.
 - 11: Normalize $\alpha_{t+1}^{(l)}$, i.e., $\alpha_{t+1}^{(l)} = \frac{\alpha_{t+1}^{(l)}}{\sum_{j=0}^L \alpha_{t+1}^{(j)}}, \forall l = 0, 1, 2, \dots, L$.
 - 12: **end for**
-



画像検索

- Dataset
 - CARS-196, CIFAR-100
 - 特徴量は VGG-19 で抽出.
- 制約
 - 訓練データから500個のシード(triplet 制約)をサンプリング.
 - これらのシードから推移的にまた500の制約を構築.
- 評価手法
 - Recall@K



画像検索

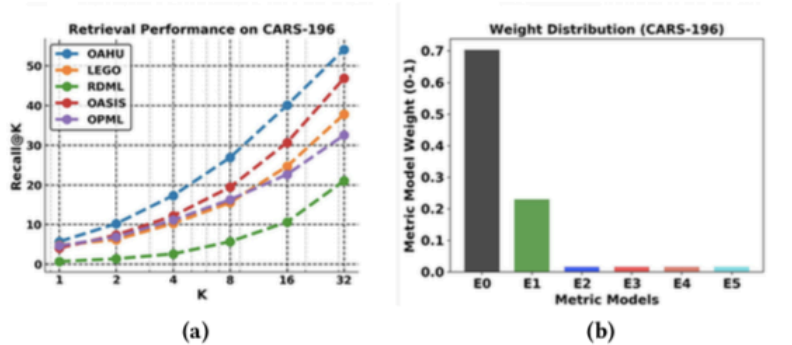


Figure 7: (a) Recall@K score on the test split of CARS-196.
(b) Metric weight distribution of OAHU on CARS-196.

(a)

- 他手法より高い結果.
- 成長速度も速い.

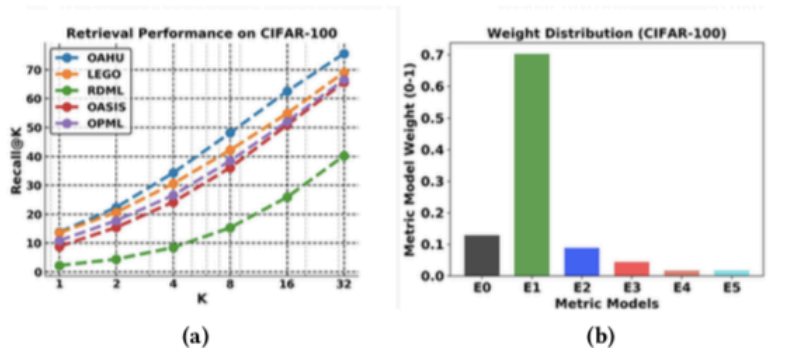


Figure 8: (a) Recall@K score on the test split of CIFAR-100.
(b) Metric weight distribution of OAHU on CIFAR-100.

(b)

- metric weight 分布



まとめ

- 新しい OML フレームワーク, OAHU の提案.
- 最先端のベースラインと比較して, ベストパフォーマンスを出した.
- また, 入力の制約の質に対してよりロバスト性を得た.