

作業用BGMに特化した楽曲推薦システム

矢倉 大夢^{1,a)} 中野 倫靖² 後藤 真孝²

概要：本稿では、作業時に集中度を高めることを目的として聴取する楽曲、「作業用BGM」に特化した楽曲推薦システムを提案する。従来、ユーザが好むであろう楽曲を推薦する手法が研究されてきたが、「とても好き」な楽曲は作業者の集中を阻害することが知られており、作業用BGMとして推薦する楽曲に適していない。提案システムは、「とても好き」や「とても嫌い」ではなく、「好き」もしくは「どちらともいえない」楽曲を、BGM聴取時のユーザからのフィードバックに基づいて推薦する。具体的には、楽曲のサビ区間までをダイジェスト的に聴取する（部分的にしか再生されない）システムとして設計することで、楽曲を「スキップ」するフィードバックによって「嫌い」な楽曲を推定する従来手法に加え、「もっと聴く」フィードバックを導入して「好き」な楽曲を推定する。さらに、「好き」として推定された楽曲は、ユーザの集中度を行動ログから推定して「とても好き」か「好き」かを識別する。これは集中度が高い時のフィードバックは、低い時より嗜好度を強く表しているという仮説に基づく。そして、楽曲間類似度に基づくLabel Spreadingにより、頑健にかつ再生履歴が少ない状況でも適切に楽曲を推薦することを可能にした。

1. はじめに

本稿では、作業中に聴取する楽曲である「作業用BGM」を推薦するシステムを提案する。作業用BGMという用語は、例えば、ニコニコ動画^{*1}において「作業用BGM」というタグが付与された動画は約14万件^{*2}投稿されており、広く認知されている。また、Lonsdaleらによる189名の学生を対象とした調査[1]では、75.7%が楽曲を聴取する理由として勉強中や作業中のBGMとしての使用を挙げ、「BGMとしての使用を挙げた回答者のほとんどが、集中を高めるために聴取していると答えた」と述べている。そこで本稿では、作業用BGMを「作業時に集中を高めることを目的として再生される楽曲」として定義する。このような楽曲を自動的に推薦できれば、作業の効率や質を上げる音楽活用につながって有用である。

提案システムは、ユーザの作業に関する行動ログ（マウス操作など）と、BGM聴取時のユーザフィードバック（曲のスキップなど）によって、作業に適した楽曲をシステムが自動的に選曲し、連続再生する（図1）。その際、ユーザ

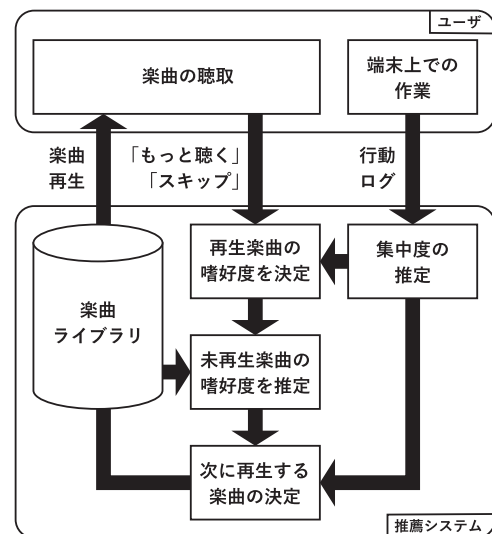


図1 提案システムの概要図。ユーザからのフィードバックと行動ログに基づく集中度推定によって嗜好度を決定する。さらに、推薦する楽曲を決定する際にも集中度を用いる。

が強く好む楽曲は集中を妨害する可能性があるため、ユーザが最も好むであろう楽曲を推薦する手法は適用できないという点で、従来の楽曲推薦と異なる。

ユーザの負担を抑えながら多様なフィードバックを得るために、複数の楽曲をダイジェスト的に（部分的に）連続再生するBGM聴取システムを設計し、ユーザによる「嫌いな曲だからスキップしたい」もしくは「好きな曲だからもっと聴きたい」という要求をフィードバック可能にする。さらに、作業に集中している時のフィードバックは、集中

¹ 筑波大学 情報学群 情報科学類
College of Information Science, University of Tsukuba
² 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
^{a)} hiromu@coins.tsukuba.ac.jp
^{*1} <http://nicovideo.jp>
^{*2} 2016年4月1日現在、138,439件。（<http://www.nicovideo.jp/tag/%E4%BD%9C%E6%A5%AD%E7%94%A8BGM>にて確認）

していない時よりも、楽曲に対するユーザの嗜好度（好き嫌い）を強く反映しているという仮説に基づき、「集中度」を推定して利用する。また、推薦に適していると判断された楽曲が複数あった場合、集中度に応じて選曲を変えることで、より作業に適した楽曲推薦を実現する。

2. 関連研究

本章では、限られたフィードバックに基づく楽曲推薦と、作業用 BGM の推薦、そして「集中度」の推定に関する関連研究について述べる。

2.1 限られたフィードバックに基づく楽曲推薦

ユーザからの限られたフィードバックに基づく手法の 1 つとして、Pampalk らの手法 [2] がある。この手法では、ユーザが楽曲の再生中に行ったスキップ操作のみをフィードバックとし、楽曲の音響特徴量を組み合わせて、以下のように推薦している。

- (1) まだ再生されていない楽曲のそれぞれについて、曲中にスキップされた楽曲との類似度のうち最大の類似度を s_s 、最後まで再生された楽曲との類似度のうち最大の類似度を s_a とする。
- (2) $s_a > s_s$ なる楽曲のうち、 s_a が最大のものを再生する。
- (3) $s_a > s_s$ なる楽曲が存在しない場合は、 $\frac{s_a}{s_s}$ が最大となる楽曲を再生する。

また、暦本による UniversalPlaylist [3] は「Yes」「No」という 2 種類のフィードバックのみから好みの楽曲を推薦するインタラクション手法を提案している。

2.2 作業用 BGM の推薦

作業用 BGM の推薦手法として、滝澤らによる LISWO [4] がある。LISWO は、ユーザの作業状況と楽曲のメタデータを入力ベクトル、楽曲長に対する再生開始からスキップされるまでの時間の割合を目的変数として、Support Vector Regression により、目的変数が大きくなるような、つまり最後までスキップされにくいような楽曲を推薦する。なお、ユーザの作業状況として、月、曜日、時刻とアクティブなアプリケーション名、キーボードの打鍵数、マウスカーソルの移動量、マウスのクリック回数を用いていた。また、楽曲のメタデータとして、アーティスト、アルバム名、発売年、ジャンルを用いていた。

この手法では、スキップされにくい楽曲を再生することはできるが、「とても好き」な楽曲によって集中が阻害されることを考慮していない。したがって、ユーザが強く好む楽曲が再生され、楽曲の聴取に意識が傾いてしまい、集中が低下してしまう可能性がある。

2.3 集中度の推定

「集中度 (concentration)」の推定については、いくつか

の研究がなされている [5–7]。例えば、立山ら [5] は、ステレオカメラで計測される視線の移動量や瞬きの回数に基づいた集中度の推定をした。また、永井ら [7] は、キーボードやマウスの操作量、及び、スクリーンショットの輝度変化量と、集中度との相関から自動推定の可能性を示唆した。

集中度に関連して、作業の「割り込みやすさ (interruptibility)」の自動推定も研究されている [8–11]。例えば、Fogarty ら [9] は、キーボードやマウスの操作量だけでなく、マイクによる会話検知や、磁気センサによるドアの開閉検知などを用いた手法を、Züger ら [10] は、生体センサで検知した脳波や皮膚電位、心拍数などを用いた手法を、そして、田中ら [11] は、キーボードやマウスの操作量とアクティブなアプリケーションの切り替え回数を組み合わせた手法を提案した。

その結果、Fogarty ら [9] と Züger ら [10] は「とても割り込んでもよい」「割り込んでもよい」「どちらともいえない」「割り込んでほしくない」「全く割り込んでほしくない」の 5 クラス分類でそれぞれ 51.5%, 32.5% の精度を得た。また、田中ら [11] は「割り込んでもよい」「どちらともいえない」「割り込んでほしくない」の 3 クラス分類で 35.4% の精度を得たと述べている。ここで、Fogarty ら、田中ら、Züger らの手法での 2 クラス分類の精度を、Züger らの基準に従い、「どちらともいえない」までをまとめて「割り込んでもよい」として混同行列 (confusion matrix) から再計算したところ、それぞれ 71.1%, 58.4%, 78.6% であった。

上記を楽曲推薦に用いる際に、ステレオカメラやドアの開閉検知センサといった外部センサの活用は、ユーザの心理的障壁やコストの点から非現実的だと考えられる。さらに、生体センサの使用は作業の邪魔となる可能性もある。また、永井ら [7]、田中ら [11] は、外部センサを用いずに収集可能な特徴量を用いたが、一定長の時間における推定を目的として、特定の操作の回数の特徴量としている。したがって、例えば、数十秒でスキップされることもあるように、時間の長さが可変である楽曲再生中の推定にそのまま適用することはできない。

3. 作業用 BGM 推薦システム

1 章で述べた通り、作業用 BGM の推薦においては楽曲の聴取が集中度に与える影響について考慮する必要がある。これは、Huang ら [12] による「とても好き (like very much)」「好き (like)」「どちらともいえない (feel passably)」「嫌い (dislike)」「とても嫌い (dislike very much)」の 5 段階で評価された楽曲への嗜好度と、楽曲の再生中の集中度との関連性についての調査からも裏付けられる。Huang らは「とても好き」または「とても嫌い」と評価した楽曲を聴取していた場合は無音環境よりも集中度が大きく低下し、「好き」「どちらともいえない」「嫌い」と評価した楽曲を聴取していた場合は集中度が向上したと結論づけている。

すなわち、集中を高めたいというモチベーションで作業用 BGM を聴取するユーザには、「とても好き」な楽曲ではなく「好き」「どちらともいえない」「嫌い」という楽曲を推薦することが求められる。ただし、「嫌い」な楽曲を推薦することは、ユーザのシステムを使用するモチベーションを損なう可能性が高いため、本稿では「好き」「どちらともいえない」という楽曲を推薦することとする。

さらに、作業中のユーザの集中を阻害しないために、推薦システムが求めるフィードバックについては、ユーザの負荷を最小限にする必要がある。しかし、ユーザに明示的な評価を要求しない、暗黙的なフィードバック手法は正確性に劣ると言われている [13]。その上、別の作業をするために副次的に音楽を聴取しているユーザからフィードバックを得る場合、評価のゆらぎや操作ミスによる誤った評価も含まれやすいと考えられる。そのため、作業用 BGM の推薦アルゴリズムには、ノイズや誤った評価に対しての耐性が求められる。

以上、作業用 BGM 推薦システムの要件をまとめると、次の通りとなる。

要件 1 「とても好き」な楽曲ではなく、「好き」または「どちらともいえない」楽曲の推薦

要件 2 ユーザの負荷を抑えたフィードバックの取得

要件 3 ノイズ耐性の高い推薦アルゴリズム

4. 手法

本章では、3 章で述べた要件を満たすよう、フィードバックからの再生楽曲の嗜好度の決定と、集中度を推定する手法を説明する。そして、未再生楽曲の嗜好度の推定と、次に再生する楽曲の決定をする推薦アルゴリズムについて述べる (図 1)。

4.1 再生楽曲の嗜好度の決定

本節では、再生された楽曲に対するユーザの嗜好度を決定するための、新たなフィードバックの実現について説明する。ただし、3 章で述べた要件 1 を満たすためには、再生楽曲の嗜好度を「とても好き」から「とても嫌い」までの 5 段階で決定する必要がある。そこで、まず「好き」「どちらともいえない」「嫌い」の 3 段階で決定するために「もっと聴く」ボタンを導入し、次に「集中度」を用いて 5 段階まで拡張することを考える (図 2)。

4.1.1 「もっと聴く」ボタンを用いたフィードバック

本節では、2.1 節で述べた Pampalk らの手法 [2] をさらに拡張した、「もっと聴く」ボタンを用いたフィードバックについて説明する。Pampalk らの手法は、要件 2 のユーザの負荷を抑えるという点では優れているが、「嫌いな曲だからスキップしたい」というフィードバックしか得られないため、「嫌い」な楽曲を排し、「好き」な楽曲を推薦することしかできず、「どちらともいえない」楽曲を選ぶことは

できなかった (図 2 (A))。

そこで、「もっと聴く」ボタンを用いて、「好きな曲だからもっと聴きたい」という要求をフィードバックできるようにすることで、「好き」「どちらともいえない」「嫌い」の 3 段階で嗜好度を決定する (図 2 (B))。そして、「もっと聴く」ボタンを実現するためには、楽曲を部分再生する必要があるため、後藤によるサビ区間検出手法 [14] を用いる。

提案するフィードバック手法での、ユーザからのフィードバックと得られる嗜好度の対応は、以下の通りである。

- 「もっと聴く」ボタンを押す

曲の末尾まで再生され、「好き」とみなす。

- 何もしない

曲の先頭から 30 秒後以降で初めてのサビ区間の終わりまで再生され、「どちらともいえない」とみなす。

- 「スキップ」ボタンを押す

直ちに次の楽曲が再生され、「嫌い」とみなす。

ここで、何も操作がなかった場合にサビ区間の終わりまでを再生するようにしたのは、後藤が「サビは楽曲中で最も多く繰り返され、印象に残る」と述べている [14] ように、楽曲への印象や嗜好度に大きく影響を与えられからである。また、好きな楽曲が再生された場合に「もっと聴く」ボタンを押すというフィードバックについても、ユーザの負荷や違和感は小さいと考えられる。これは、携帯音楽プレーヤ Walkman に楽曲をダイジェスト的に連続再生する機能が搭載されている*3ことや、濱崎らによる音楽視聴支援サービス Songrium [15] の連続再生機能 [16,17] では、ダイジェスト的な連続再生の中で、ユーザの操作に応じて末尾まで再生する機能も実装されていることから確かめられる。

また、曲の先頭から 30 秒後以降という制限を加えたのは、10 秒や 20 秒で「スキップ」や「もっと聴く」という判断を下すのが難しいと考えたためである。

4.1.2 「集中度」によるフィードバックの拡張

4.1.1 項で述べた手法によって、「好き」「どちらともいえない」「嫌い」という 3 段階で嗜好度を得た後、以下の仮説に基づき「集中度」というパラメータを用いて、「とても好き」「とても嫌い」を含めた 5 段階の嗜好度を決定する。

仮説 集中度が高い時のフィードバックは、集中度が低い時のフィードバックに対し、より強く嗜好度を表している。

つまり、集中度とフィードバックの対応に基づいて表 1 のように、嗜好度を 5 段階で決定することができる。なお、本稿における「集中度」の定義は、永井ら [7] に従って「タスクへ没頭している程度」とし、その推定手法については次節で説明する。

*3 http://www.sony.jp/walkman/products/NW-W270S_series/feature_4.html#L1_170

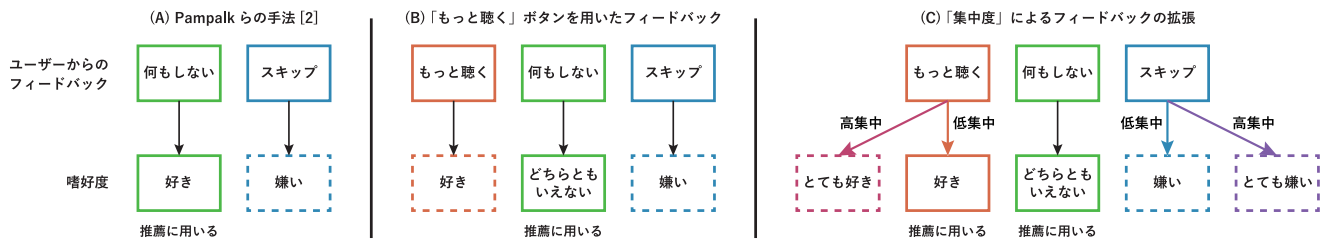


図 2 フィードバックと嗜好度の対応図. (A) の「スキップ」によるフィードバックでは「好き」と「嫌い」の 2 段階の嗜好度しか得られないが, (B) の「もっと聴く」ボタンによって「どちらともいえない」を含めた 3 段階の嗜好度が得られる. さらに, (C) のように集中度を組み合わせることで, 「とても好き」「とても嫌い」を含めた 5 段階の嗜好度が得られる.

表 1 このように集中度とフィードバックとの紐付けを行うことで, 図 2 (C) のように 5 段階の嗜好度が得られる.

集中度	フィードバック		
	もっと聴く	何もしない	スキップ
高集中	とても好き	どちらともいえない	とても嫌い
低集中	好き	どちらともいえない	嫌い

4.2 行動ログに基づく集中度の推定手法

本稿では, 表 2 に示した, キーボード, マウスの入力, HTTP/HTTPS (Web) 通信の 3 種類の行動ログの n-gram を特徴量として, 集中度を推定することを提案する. これは, 行動ログの n-gram を特徴量として機械学習に用いるという, マルウェア検知などで提案されている手法 [18–21] を基にした. 例えば, Rieck ら [19] は, Windows の API コールと引数から得られる n-gram を特徴量として, マルウェアの分類を行っている.

前述の 3 種類の行動ログは, 田中ら [11] によって提案された特徴量である, キーボードとマウスの入力情報と, アクティブなアプリケーションに関する情報を基にした. ただし, 行動ログとして記録する際に, 具体的な入力文字やクリック動作の種類を保存し, ログの情報量を増やすことで, 推定精度の向上を目指す. さらに, 集中度を推定する上では, 同じ入力内容であっても入力先のアプリケーションによって, 異なる情報として扱うべきだと考えられるため, アクティブなアプリケーションが変化したというイベントを記録するのではなく, どのアプリケーションに入力したかという形式で記録することとした. これによって, 例えば, 同じクリックという動作であっても Firefox であれば Web ブラウジングを, Skype であれば音声通話やチャットをしている等の違いを表現できる.

また, PC を利用した作業において, Web アクセスの有無やアクセス先は作業内容を反映すると考えられる. 例えば, 表 2 のように, Google に対する GET リクエストであれば Web 検索を, Twitter に対する POST リクエストであれば SNS への投稿をしているといった推察に繋がる. そのため, 新たな特徴量として Web 通信のアクセスログ (リクエストメソッド及びホスト名) を採用することとした.

そして, 以下の 2 つの理由から, 推定アルゴリズムには Crammer らによる Adaptive Regularization of Weight Vectors (以降, AROW と表記) [22] を採用する.

- 行動ログを元にした学習データを大量に集めるのは難しいため, 少ないデータからでも高い精度で推定できる必要がある.
- 「集中度」という曖昧さを持つ主観的なパラメータにも対応できるような, ノイズ耐性が求められる.

AROW は, 少ない学習データでも収束が早く, 高いノイズ耐性を持つアルゴリズムとして知られており, 村上らによるマルウェアの分類 [21] によっても用いられている.

4.3 作業用 BGM 推薦アルゴリズム

本節では, 4.1 節で述べたフィードバック手法によって得られる再生中の楽曲の嗜好度を用いて, 推薦する楽曲を決定する手法について説明する. その際, 3 章の要件 1 を満たすために, まず未再生楽曲の嗜好度を推定し, 「好き」「どちらでもない」と推定された楽曲群から集中度に応じて 1 曲を選び出すという手法を採る.

4.3.1 Label Spreading による未再生楽曲の嗜好度推定

未再生楽曲の嗜好度推定には, 以下の 2 点が求められる.

- 誤ったフィードバックやユーザからの評価にも対処できるノイズ耐性を持つこと
 - 少ないフィードバックからでも正確に推薦できること
- ここで, ノイズ耐性が必要となるのは, 3 章で述べた要件 3 に加え, 集中度の推定誤りによってもノイズが生まれる可能性があるためである. また, ある程度のフィードバックがないと推薦できないのであれば, フィードバックが蓄積されるまで, ユーザに集中を阻害するような楽曲を聞きながら作業してもらう必要があるため, 実用的でない.

そこで, 本稿では Label Spreading [23] を用いる (図 3). Label Spreading は, Shao らの楽曲推薦手法 [24] に採用されている Label Propagation [25] と同じグラフベースの半教師あり学習アルゴリズムである. 特に, Label Spreading では, Label Propagation での制約条件を緩和して学習を行うため, Wang らによる比較 [26] によっても確認できる

表 2 キーボード、マウスについては具体的な入力内容と入力先アプリケーション名を、Web 通信についてはリクエストメソッドとホスト名を記録して、その n-gram を集中度推定の特徴量に用いる。

種類	凡例	例
キーボード	“key [アプリケーション名] [キー]” (修飾キーはまとめて扱われる)	“key Google Chrome a” (Google Chrome に「a」を入力)
		“key Microsoft Excel <[Ctrl: v]>” (Microsoft Excel に「Ctrl+v」を入力)
マウス	“mouse [アプリケーション名] [番号]” (左/中/右クリック, 上下左右スクロールが 1~7 に対応)	“mouse Firefox 1” (Firefox に左クリック)
		“mouse Skype 4” (Skype で上スクロール)
HTTP/HTTPS 通信 (Web)	“net [リクエストメソッド] [ホスト名]” (GET または POST リクエストのみ)	“net GET www.google.com” (www.google.com に GET リクエスト)
		“net POST twitter.com” (twitter.com に POST リクエスト)

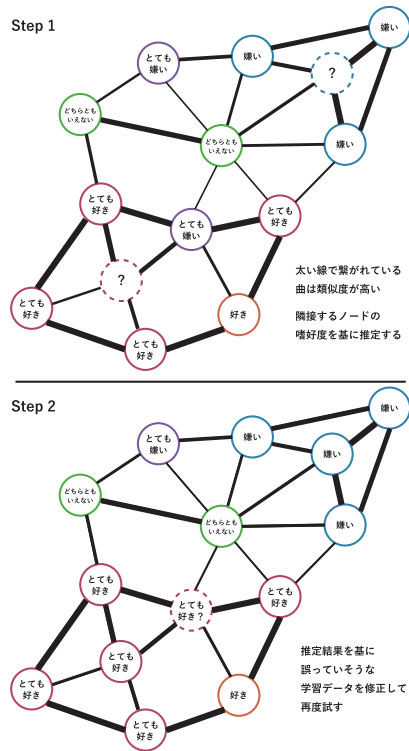


図 3 Label Spreading の処理手順 (イメージ図)。重みの大きな辺で繋がれた (類似度の大きい) 楽曲は同じ嗜好度を共有するという仮定の下で推定を行う (Step 1)。さらに、推定結果から誤っているような学習データが見つければ、修正して上で再度推定を行う (Step 2)。

通り、ノイズへの耐性が高いことが知られている。

これらは、学習データが限られていても高い精度で推定できることが知られている。(Zhou ら [23] による k-近傍法や SVM との比較によっても確認できる。) また、これら 2 つアルゴリズムはグラフ構造に基づいて上で学習するため、楽曲間類似度を辺の重みとして用いることで、嗜好度推定が可能となる。

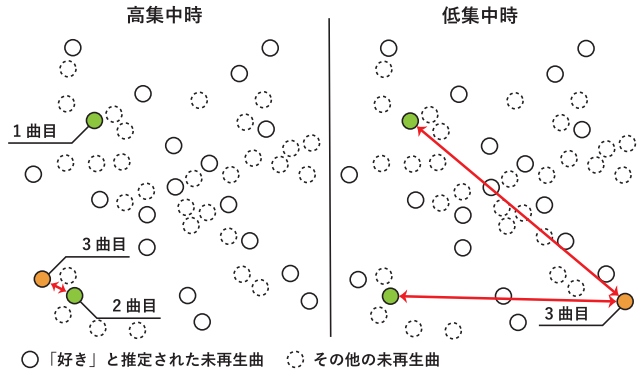


図 4 集中度を用いた楽曲推薦の処理手順 (イメージ図)。集中度が高い時には、直前の楽曲と最も似た楽曲を、集中度が低い時には、直前に再生した 2 曲と最も似ていない楽曲を選ぶ。

4.3.2 集中度に応じた楽曲推薦

ユーザの集中を高めることを目的として、同じ嗜好度を推定された楽曲から推薦する楽曲を選ぶ際にも「集中度」を用いる手法を提案する。具体的には、集中度が高い時には直前に再生した楽曲と類似した楽曲を優先し、低い時には多様な楽曲が再生されるようにする。つまり、集中している時には急な曲調の変更によって気が散るのを避け、集中できていない時には曲調の変化によって気持ちを切り替えるきっかけを作る。これは、Wells による調査 [27] で、225 人中 73.3% が音楽によって気分を切り替えるという項目に同意した一方で、その契機となる楽曲に類似性が見られなかったという結果に基づいている。

具体的な推薦手順は、以下の通りである (図 4)。

- (1) 2 曲目まではランダムに楽曲を再生する。
- (2) 3 曲目以降は、以下の手順で推薦する楽曲を選ぶ。

- (a) 推薦する楽曲の候補として、「好き」というラベルを推定された楽曲群を選ぶ。ただし、「好き」と推定された楽曲群が存在しない場合は、「どちらともいえない」「嫌い」「とても好き」「とても嫌



図 5 作業用 BGM 推薦システムのインタフェース。上部の Songle Widget によって楽曲を再生し、下部の「スキップ」「もっと聴く」ボタンによってフィードバックを取得する。

い」という優先順位で選択する。

- (b) 直前の楽曲を再生中、集中度が高いと推定された場合は、楽曲群候補から、直前に再生した 1 曲との類似度が最大となるものを推薦する。
- (c) 直前の楽曲を再生中、集中度が低いと推定された場合は、楽曲群候補から、直前に再生した 2 曲との類似度の和が最小となるものを推薦する。

ここで、集中度が低かった場合に、直前に再生した 1 曲との類似度のみを最小化すると、2 つのジャンルを交互に行き来するような選択を繰り返してしまう可能性があるため、直前に再生した 1 曲との類似度ではなく、2 曲まで考慮して楽曲を選択することとした。

5. 実装

本章では、これまで述べた手法を実現する、作業用 BGM 推薦システムの実装について述べる。

ユーザインタフェースを図 5 に示す。Songle Widget による楽曲再生画面に加え、「一時停止」と 4.1 節で述べた「スキップ」「もっと聴く」の 3 種類のボタン、そして音量調節バーを持つ。

システムの構成は図 6 に示すように、再生・制御、行動ログ収集、集中度推定、楽曲推薦の 4 つのモジュールを持つ。中心となるのが、再生・制御モジュールで、他の 3 つと連携して得られた推薦楽曲を、埋め込みブラウザ内で動作する Songle Widget [28] を用いて再生する。Songle Widget を用いる利点としては、サビ区間に基づく再生制御が容易であるということに加え、Songle [29] 内でユーザ

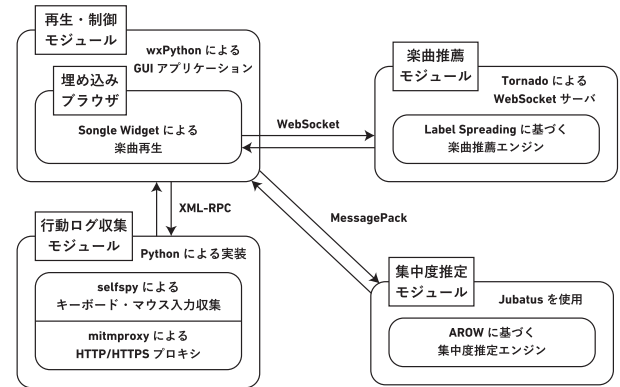


図 6 作業用 BGM 推薦システムの構成。左上部の再生・制御モジュールがインタフェース(図 5)を担い、他の 3 つのモジュールがバックグラウンドで動作する。

が誤り訂正を行った、より精度の高いサビ区間情報を利用できる可能性があるという点が挙げられる。

また、行動ログ収集モジュールでは、キーボード・マウスへの入力と Web 通信の記録を行う。キーボード・マウスへの入力については、selfspy^{*4}により、アクティブなアプリケーション名と共に記録し。Web 通信は、mitmproxy^{*5}を用いて、リクエストごとのメソッドとホスト名を記録する。

集中度推定モジュールと楽曲推薦モジュールは、機械学習に基づく 2 つのモジュールである。前者は、行動ログからの集中度の推定を行うもので、機械学習フレームワーク Jubatus^{*6}を利用した。後者は、ユーザによるフィードバックや集中度から楽曲推薦を行うもので、機械学習ライブラリ scikit-learn^{*7}による Label Spreading の実装を用いた。また、Songle Widget から通信できるようにするため、Tornado^{*8}を用いて WebSocket サーバ上に実装した。

プライバシーへの配慮として、行動ログはすべてハッシュ化した上で保存し、通信はすべて暗号化した。

6. 評価

本章では、集中度の推定精度、フィードバック手法の妥当性、及び、推薦結果の妥当性の 3 つの項目について、評価方法とその結果を述べる。

6.1 対象楽曲と音響特徴量の算出

対象とした楽曲は、ニコニコ動画において「VOCALOID」タグを付与されている楽曲のうち、再生数が上位の 50 曲(6.5 節のみ 200 曲)とした。ここで VOCALOID 楽曲を利用したのは、Songle 上に多くの楽曲が登録されており、ユーザによる訂正を経たサビ区間情報が得やすいことと、様々なジャンルの楽曲が CGM (Consumer Generated Media) 的

*4 <https://github.com/gurgeh/selfspy>

*5 <https://mitmproxy.org/>

*6 <http://jubat.us/>

*7 <http://scikit-learn.org/>

*8 <http://www.tornadoweb.org/>

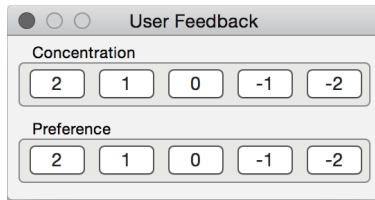


図 7 集中度と嗜好度の正解データを入力するダイアログ. 集中度の推定精度とフィードバックの妥当性の検証に用いるため, 各楽曲を再生するごとに入力した.

に多数投稿されており [30,31], 楽曲推薦システムを評価する上で適していると考えられることによる. VOCALOID 楽曲のジャンルの多様さは, VOCALOID に関連するタグとしてニコニコ動画で用いられているタグのうち楽曲ジャンルを示すために用いられているタグの数^{*9}からも分かる.

推薦で用いる楽曲間類似度の算出は, Songrium [15] と同様の手法を用いた. 具体的には, MARSYAS [32] を用いて, スペクトル特徴量やメル周波数ケプストラム係数, テンポに関する 35 次元の音響特徴量ベクトルを求め, 主成分分析によりその第 1~3 主成分を用いた.

6.2 評価のためのデータ収集

被験者は 10~20 代の男性 5 名 (楽曲を聴取しながら作業する習慣がある) に, 5 章で述べたシステムを使用してもらい, 楽曲を再生中の行動ログと「スキップ」「もっと聴く」といったユーザからのフィードバックを収集した.

さらに, 各楽曲を再生するごとに図 7 のダイアログを表示し, 集中度の推定精度とフィードバック手法の妥当性を検証するために, 正解データとして楽曲の嗜好度と再生中の集中度を入力した. 「2」が「とても好き」「とても集中していた」に, 「-2」が「とても嫌い」「全く集中していなかった」に対応した Likert 尺度である.

6.3 「集中度」の推定精度

はじめに, 集中度の推定精度の評価を行った. 行動ログから推定される集中度と, 正解データとしてユーザが入力した集中度の適合度を 5-fold cross validation によって確かめた結果, 5 クラス分類では 35.2%, 2 クラス分類では 69.6%の精度で, 混同行列 (confusion matrix) はそれぞれ表 3, 表 4 の通りとなった. なお, 2 クラス分類については Züger ら [10] と同様に, 「とても集中していた」「集中していた」「どちらともいえない」を高集中, 「集中していなかった」「全く集中していなかった」を低集中とした.

2 クラス分類の結果精度は, 物理センサを用いた手法 (71.1%) [9] や, 生体センサを用いる手法 (78.6%) [10] よ

表 3 集中度推定の混同行列 (5 クラス分類)

推定結果	正解ラベル				
	2	1	0	-1	-2
2	16	13	4	6	2
1	9	22	5	6	7
0	3	7	10	5	5
-1	4	10	9	22	24
2	5	11	6	18	21

表 4 集中度推定の混同行列 (2 クラス分類)

推定結果	正解ラベル	
	高集中 (2~0)	低集中 (-1,-2)
高集中	89	21
低集中	45	85

表 5 「もっと聴く」ボタンを用いたフィードバックと楽曲の嗜好度の対応 (相関係数 0.60)

フィードバック	楽曲の嗜好度		
	好き (2,1)	どちらともいえない (0)	嫌い (-1,-2)
もっと聴く	42	9	5
何もしない	36	80	23
スキップ	3	7	45

り低いものの, 外部センサを用いずに収集可能な特徴量のみを用いた, 田中らの手法 (58.4%) [11] より高かったことから, 提案手法の有効性が確認できた.

6.4 フィードバック手法の妥当性

次に, 4.1 節で述べたフィードバック手法について, ユーザの嗜好度を正しく反映しているかを評価した.

まず, ユーザが「スキップ」とした楽曲への嗜好度は低く, 「もっと聴く」とした楽曲への嗜好度は高くなっているかどうか, つまり「もっと聴く」ボタンを用いたフィードバックがユーザの嗜好度を正しく反映しているかを評価する. ユーザのフィードバックから得られる嗜好度と, 正解としてユーザが入力した嗜好度の相関を表 5 に示す.

ユーザのフィードバックから得られる嗜好度とユーザが入力した嗜好度について, ピアソンの相関係数は 0.60 ($p < 0.01$) であった. ただし, ユーザが入力した嗜好度については「とても好き」「好き」をまとめて「好き」に, 「とても嫌い」「嫌い」をまとめて「嫌い」とした.

さらに, 「もっと聴く」ボタンに「集中度」の推定を組み合わせたフィードバックが, ユーザの嗜好度を正しく反映しているかについてを評価した. 表 6 に示すように, ユーザのフィードバックから得られる嗜好度と, 正解としてユーザが入力した嗜好度について, ピアソンの相関係数は 0.67 ($p < 0.01$) であった.

以上の結果から, 4.1.2 項で述べた仮説は支持され, 表 1 に表されるフィードバックは, ユーザの嗜好度を正しく反映していることが分かる.

^{*9} 2016 年 4 月 1 日現在, ニコニコ動画に関する記事をまとめた Wiki システム「ニコニコ大百科」には 141 件が挙げられている. (<http://dic.nicovideo.jp/a/vocaloid%E9%96%A2%E9%80%A3%E3%81%AE%E3%82%BF%E3%82%B0%E4%B8%80%E8%A6%A7>にて確認)

表 6 「もっと聴く」ボタンに「集中度」の推定を用いたフィードバックと楽曲の嗜好度（相関係数 0.67）

フィードバック	楽曲の嗜好度				
	2	1	0	-1	-2
もっと聴く（高集中）	12	2	1	1	0
もっと聴く（低集中）	9	19	8	3	1
何もしない	9	27	80	19	4
スキップ（低集中）	0	3	6	21	10
スキップ（高集中）	0	0	1	2	12

6.5 作業用 BGM 推薦の結果の妥当性

最後に、4.3 節で提案した推薦アルゴリズムは、Pampalk ら [2] のようにシミュレーションによって評価した。なぜなら、複数の手法を比較するためには同一の集中度系列を作り出す必要があるが、主観評価において集中度を操作することができないためである。

そこで、まずニコニコ動画のタグ情報を用いてユーザの嗜好をシミュレーションし、それを基にフィードバックもシミュレーションした。また、指標としては、滝澤ら [4] と同様に、ユーザが「スキップ」や「もっと聴く」といったアクションをせず「何もしない」というフィードバックを与えた楽曲の割合を用いた。これは、ユーザがアクションを行いたくなる楽曲再生は集中の障害に繋がり、何もしなければ集中しているということだからである。

6.5.1 比較対象とするアルゴリズム

まず、評価を行う上で、比較対象とする 2 つの推薦アルゴリズムを実装した。1 つは、未再生の楽曲からランダムに選ぶアルゴリズムである。もう 1 つは 4.1.1 項で述べた「もっと聴く」ボタンを用いたフィードバックに、2.1 節で述べた Pampalk らのアルゴリズム [2] を組み合わせたもので、以下の手順で推薦する。

- (1) まだ再生されていない楽曲のそれぞれについて、「スキップ」または「もっと聴く」のフィードバックがされた楽曲との類似度のうち最大の類似度を s_s 、「何もしない」ままサビ区間まで再生された楽曲との類似度のうち最大の類似度を s_a とする。
- (2) $s_a > s_s$ なる楽曲のうち、 s_a が最大のものを再生する。
- (3) $s_a > s_s$ なる楽曲が存在しない場合は、 $\frac{s_a}{s_s}$ が最大となる楽曲を再生する。

6.5.2 嗜好のシミュレーション

嗜好のシミュレーションには、ニコニコ動画のタグが楽曲のカテゴリやコンテキストを反映しているため、それを利用した。楽曲間類似度を用いないのは、比較手法と提案手法の両方が楽曲間類似度を用いて推薦するために、同一の情報に基づいて嗜好度をシミュレーションしても、正しい評価ができないからである。具体的な手順は、以下の通りである（表 7）。

- (1) ニコニコ動画の VOCALOID 楽曲のうち、再生数が上位 200 位以内のものに付与されたタグを列挙する。

表 7 ロック調やテクノポップ調の楽曲は好み、和風やバラード風な楽曲は好まないというユーザの嗜好がシミュレーションされた場合、それぞれのタグの印象とスコアは以下の通りとなる。

(1) タグ	(2) 割合	(3) 印象	(4) 嗜好スコア
VOCAROCK	0.27 (54 曲)	ポジティブ	0.27
VOCALOID 和風曲	0.23 (46 曲)	ネガティブ	-0.23
ボカロバラード	0.21 (42 曲)	ネガティブ	-0.21
ミクノポップ	0.18 (36 曲)	ポジティブ	0.18

- (2) それぞれのタグについて、200 曲中でそのタグが用いられている曲の割合を求める。
- (3) それぞれのタグについて、「ポジティブ」か「ネガティブ」かの印象をランダムに割り振る。
- (4) それぞれのタグについて、「ポジティブ」ならその割合を、「ネガティブ」ならその割合に -1 を掛けたものを嗜好スコアとする。
- (5) それぞれの楽曲について、付与されたタグの嗜好スコアを合計したものを、その楽曲の嗜好スコアとする。

例えば、表 7 のようにタグの嗜好スコアが算出された場合、ロック調やテクノポップ調の楽曲は好むが、和風やバラード風な楽曲は好まないというユーザの嗜好をシミュレーションしていることになる。そのようなユーザの「VOCAROCK」「VOCALOID 和風曲」の 2 つのタグが付与された楽曲に対する嗜好スコアは $0.04 (= 0.27 - 0.23)$ となる。

なお、対象としたすべての楽曲に付与されている「VOCALOID」タグは除外した上で算出を行った。また、タグごとのスコアをランダムに割り振るのではなく、タグの使用されている割合を用いたのは、似た意味を持つタグに大きく矛盾したスコアが割り振られてしまい、嗜好シミュレーションの妥当性が失われてしまう可能性を低減するためである。タグごとの使用曲数の分布は、図 8 のようにロングテール上になっていることから、似た意味を持つタグは淘汰され、より多くの楽曲で使われているタグに集約されていると期待できることによる。これは、楽曲をより広めたいという投稿者、聴取者にとっては、多くの楽曲で使われているタグを用いることで、検索性が高まり、より多くの人の目に触れやすくなるというメリットがあることから説明できる。そのため、似た意味を持つタグが 2 つ以上あっても、最も多くの楽曲で使われているもの以外はほとんど使われていないために、嗜好スコアの算出に大きな影響を与えない。

6.5.3 フィードバックのシミュレーション

次に、集中度の系列を生成し、それに基づいてフィードバックのシミュレーションを行った。

集中度は、2（とても集中している）～-2（全く集中していない）の間で、同一または隣接したラベルへと等しい確率

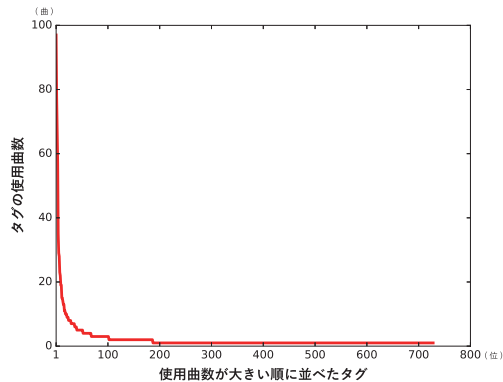


図 8 対象とした 200 曲におけるタグごとの使用数の分布

で遷移するようにした。そして、集中度に基づいて、「スキップ」または「もっと聴く」の閾値となる嗜好スコアを変更するようにした。具体的には、高集中 (2~0) の時は、嗜好スコアが 0.75 以上の楽曲に「もっと聴く」、-0.75 以下の楽曲に「スキップ」というフィードバックを生成し、低集中 (-1,-2) の時は、嗜好スコアが 0.5 以上の楽曲に「もっと聴く」、-0.5 以下の楽曲に「スキップ」というフィードバックを生成した。

このシミュレーションによって生成されるフィードバック例を図 9 に示す。この例では、3 曲目と 7 曲目は同じ嗜好スコア (0.58) だが、3 曲目では高集中状態なので「何もしない」を、7 曲目では低集中状態なので「もっと聴く」を生成する。同様に、2 曲目と 9 曲目も同じ嗜好スコア (-0.69) だが、2 曲目では高集中状態なので「何もしない」を、9 曲目では低集中状態なので「スキップ」を生成する。

6.5.4 シミュレーションによる評価結果

評価は、フィードバックデータがない状態から推薦を開始し、1 曲ごとにシミュレーションされたフィードバックの情報を生成するという手法で行った。50 回の試行 (異なる 50 ユーザの嗜好と集中度系列のシミュレーション) について、100 曲分の推薦を行い、平均を取った結果が表 8 である。これは、推薦された楽曲のうち、「スキップ」または「もっと聴く」ではなく「何もしない」というフィードバックがシミュレーションされたものの割合を表している。つまり、提案手法は比較手法より、集中を阻害しない「何もしない」ような楽曲を推薦できているということである。

この結果から、提案手法は適切な楽曲を推薦できている、フィードバックの数が限られている状態でも、高い精度で推薦できているということが分かる。

7. まとめ

本稿では、作業時の集中を高めるための作業用 BGM 楽曲推薦システムについて述べた。「とても好き」な楽曲ではなく「好き」や「どちらともいえない」楽曲を優先するという、作業用 BGM に特有の推薦アルゴリズムを実現するた

めに、「スキップ」だけでなく「もっと聴く」ボタンも用いた。それによって、ユーザの心理的抵抗や負荷を抑えながら「嫌い」という評価だけでなく「好き」という評価を得るフィードバック手法を実現した。さらに、ユーザの行動からの「集中度」の自動推定を行い、「好き」や「嫌い」の度合いを推定し、より細かい尺度でのユーザの嗜好度を取得することを可能にした。物理センサなどを用いずにユーザの集中度を推定する手法についても提案し、既存手法より高い精度で推定できることを確認した。最後に、タグ情報を用いて嗜好をシミュレーションし、フィードバックを生成した結果、提案した作業用 BGM 推薦システムが高い推薦精度を持つことを示した。

今後は、「集中度」というパラメータの導入によって生まれる全く新たなインタラクションの可能性についても模索していきたい。例えば、予め設定された時刻に近くなるにつれ、集中を阻害するような楽曲を再生する割合を増やしていくというインタラクションを考えている。これにより、出かけなければならない時間になってもどうしても作業を続けたいというシチュエーションや、予定の時間を過ぎて気づかないという失態を回避できる。

謝辞 本研究の一部は JST CREST の支援を受けた。

参考文献

- [1] Lonsdale, A. J. and North, A. C.: Why do we listen to music? A uses and gratifications analysis, *British Journal of Psychology*, Vol. 102, No. 1, pp. 108–134 (2011).
- [2] Pampalk, E., Pohle, T. and Widmer, G.: Dynamic Playlist Generation Based on Skipping Behavior, *ISMIR 2005*, pp. 634–637 (2005).
- [3] 暦本純一: UniversalPlaylist: 利用者の嗜好に動的に適合するメディア再生機構, *インタラクション 2005*, pp. 237–244 (2005).
- [4] 滝澤勇介, 西本一志: 作業状況に基づく「ながら聴き」用自動選曲プレーヤ“LISWO”, *情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI)*, Vol. 2009, No. 28, pp. 115–122 (2009).
- [5] Tateyama, Y., Matsumoto, Y. and Kagami, S.: Concentration detection by eye movements: towards supporting a human, *Proc. of the IEEE Intl. Conf. on Systems, Man & Cybernetics*, pp. 1544–1548 (2004).
- [6] 本田新九郎, 富岡展也, 木村尚亮, 大澤隆治, 岡田謙一, 松下 温: 作業者の集中度に応じた在宅勤務環境の提供—仮想オフィスシステム Valentine, *情報処理学会論文誌*, Vol. 39, No. 5, pp. 1472–1483 (1998).
- [7] 永井有希, 徐 建鋒, 酒澤茂之: デスクワークにおける主観的集中度と PC 操作ログの関係分析, *映像情報メディア学会年次大会講演予稿集*, Vol. 2015, pp. 14-5-1–14-5-2 (2014).
- [8] Fogarty, J., Hudson, S. E. and Lai, J.: Examining the robustness of sensor-based statistical models of human interruptibility, *CHI 2004*, pp. 207–214 (2004).
- [9] Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S. B., Lee, J. C. and Yang, J.: Predicting human interruptibility with sensors, *ACM Trans. Comput.-Hum. Interact.*, Vol. 12, No. 1, pp. 119–146 (2005).

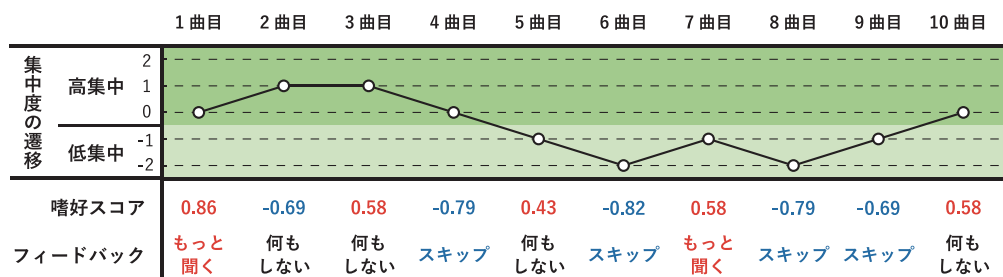


図 9 フィードバックのシミュレーション例。高集中状態か低集中状態かによって、3 曲目と 7 曲目、2 曲目と 9 曲目のように同じくらい好き(嫌い)とされる楽曲に対しても、生成されるフィードバックは変わる。

推薦した曲数	10 曲	20 曲	30 曲	40 曲	50 曲	60 曲	70 曲	80 曲	90 曲	100 曲
ベースライン (ランダム)	0.65	0.67	0.67	0.66	0.66	0.67	0.66	0.66	0.67	0.67
比較手法 (「もっと聴く」ボタンのみ)	0.65	0.70	0.72	0.71	0.72	0.71	0.69	0.70	0.69	0.68
提案手法	0.72	0.72	0.73	0.73	0.74	0.74	0.75	0.76	0.78	0.78

表 8 推薦された楽曲のうち、「何もしない」というフィードバックが与えられた楽曲の割合を表しており、提案手法が最も高い数値を示していることから、最も集中を妨害しにくい推薦手法だと言える。

- [10] Züger, M. and Fritz, T.: Interruptibility of Software Developers and its Prediction Using Psycho-Physiological Sensors, *CHI 2015*, pp. 2981–2990 (2015).
- [11] Tanaka, T. and Fujita, K.: Study of user interruptibility estimation based on focused application switching, *CSCW 2011*, pp. 721–724 (2011).
- [12] Huang, R. H. and Shih, Y. N.: Effects of background music on concentration of workers, *Work*, Vol. 38, No. 4, pp. 383–387 (2011).
- [13] 神島敏弘: 推薦システムのアルゴリズム (1), 人工知能学会誌, Vol. 22, No. 6, pp. 826–837 (2007).
- [14] Goto, M.: A chorus section detection method for musical audio signals and its application to a music listening station, *IEEE Trans. Audio, Speech & Language Processing*, Vol. 14, No. 5, pp. 1783–1794 (2006).
- [15] Hamasaki, M. and Goto, M.: Songrium: a music browsing assistance service based on visualization of massive open collaboration within music content creation community, *Proc. of the 9th Intl. Symposium on Open Collaboration*, pp. 4:1–4:10 (2013).
- [16] Hamasaki, M., Goto, M. and Nakano, T.: Songrium: a music browsing assistance service with interactive visualization and exploration of protect a web of music, *WWW '14*, pp. 523–528 (2014).
- [17] 濱崎雅弘, 中野倫靖, 後藤真孝: Songrium RelayPlay: リレー再生による同曲異唱コンテンツ視聴支援インタフェース, *WISS 2015*, pp. 25–30 (2015).
- [18] Firdausi, I., lim, C., Erwin, A. and Nugroho, A. S.: Analysis of Machine Learning Techniques Used in Behavior-Based Malware Detection, *ACT 2010*, pp. 201–203 (2010).
- [19] Rieck, K., Trinius, P., Willems, C. and Holz, T.: Automatic analysis of malware behavior using machine learning, *Journal of Computer Security*, Vol. 19, No. 4, pp. 639–668 (2011).
- [20] Canali, D., Lanzi, A., Balzarotti, D., Kruegel, C., Christodorescu, M. and Kirda, E.: A quantitative study of accuracy in system call-based malware detection, *ISSTA 2012*, pp. 122–132 (2012).
- [21] 村上純一, 鵜飼裕司: 類似度に基づいた評価データの選別によるマルウェア検知精度の向上, *CSS2013 論文集*, Vol. 2013, No. 4, pp. 870–876 (2013).
- [22] Crammer, K., Kulesza, A. and Dredze, M.: Adaptive regularization of weight vectors, *Machine Learning*, Vol. 91, No. 2, pp. 155–187 (2013).
- [23] Zhou, D., Bousquet, O., Lal, T. N., Weston, J. and Schölkopf, B.: Learning with Local and Global Consistency, *NIPS 2003*, pp. 321–328 (2003).
- [24] Shao, B., Ogihara, M., Wang, D. and Li, T.: Music Recommendation Based on Acoustic Features and User Access Patterns, *IEEE Trans. Audio, Speech & Language Processing*, Vol. 17, No. 8, pp. 1602–1611 (2009).
- [25] Zhu, X., Ghahramani, Z. and Lafferty, J. D.: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions, *ICML 2003*, pp. 912–919 (2003).
- [26] Wang, F., Li, T., Wang, G. and Zhang, C.: Semi-supervised Classification Using Local and Global Regularization, *AAAI 2008*, pp. 726–731 (2008).
- [27] Wells, A.: Popular Music: Emotional Use and Management, *The Journal of Popular Culture*, Vol. 24, No. 1, pp. 105–117 (1990).
- [28] Goto, M., Yoshii, K. and Nakano, T.: Songle Widget: Making Animation and Physical Devices Synchronized with Music Videos on the Web, *IEEE ISM 2015*, pp. 85–88 (2015).
- [29] Goto, M., Yoshii, K., Fujihara, H., Mauch, M. and Nakano, T.: Songle: A Web Service for Active Music Listening Improved by User Contributions, *ISMIR 2011*, pp. 311–316 (2011).
- [30] Hamasaki, M., Takeda, H. and Nishimura, T.: Network analysis of massively collaborative creation of multimedia contents: case study of hatsune miku videos on nico nico douga, *UXTV 2008*, pp. 165–168 (2008).
- [31] 後藤真孝: CGM の現在と未来: 初音ミク, ニコニコ動画, ビアプロの切り拓いた世界: 1. 初音ミク, ニコニコ動画, ビアプロが切り拓いた CGM 現象, 情報処理, Vol. 53, No. 5, pp. 466–471 (2012).
- [32] Tzanetakis, G. and Cook, P.: MARSyas: a framework for audio analysis, *Organised Sound*, Vol. 4, pp. 169–175 (2000).