Multidimensional Clustering of Massiv Open Online Course (MOOC) offers

Applying unsupervised learning algorithms
FCM and SOM to MOOC textual descriptions

Bachelor thesis – final presentation

Kai-Henning Wilker

08.12.2016

Agenda

- Introduction / goals
- MOOC clustering application
- Clustering process
- Evaluation
- Conclusions & future work

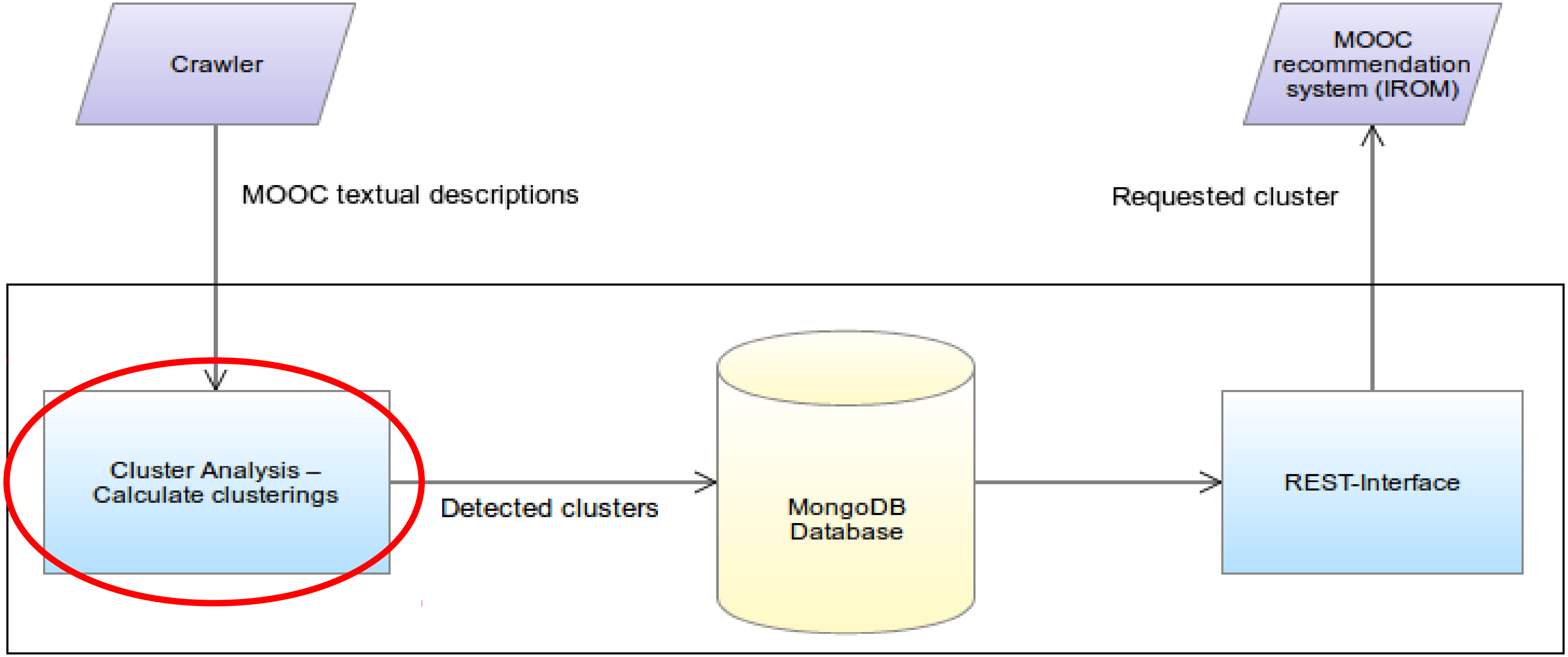
Goals

- Vision: build MOOC recommendation system for students
- Making recommendations using clusters
- Goal: cluster analysis of MOOC textual descriptions with Fuzzy C-Means (FCM) and Self-organizing Maps (SOM)
- Questions:
  - Can valid clusters be found?
  - Which clustering algorithm performs better?
  - What are the best meta-parameters for the algorithms?
  - What is the best vector representation of the documents?
  - How to evaluate a cluster's quality?

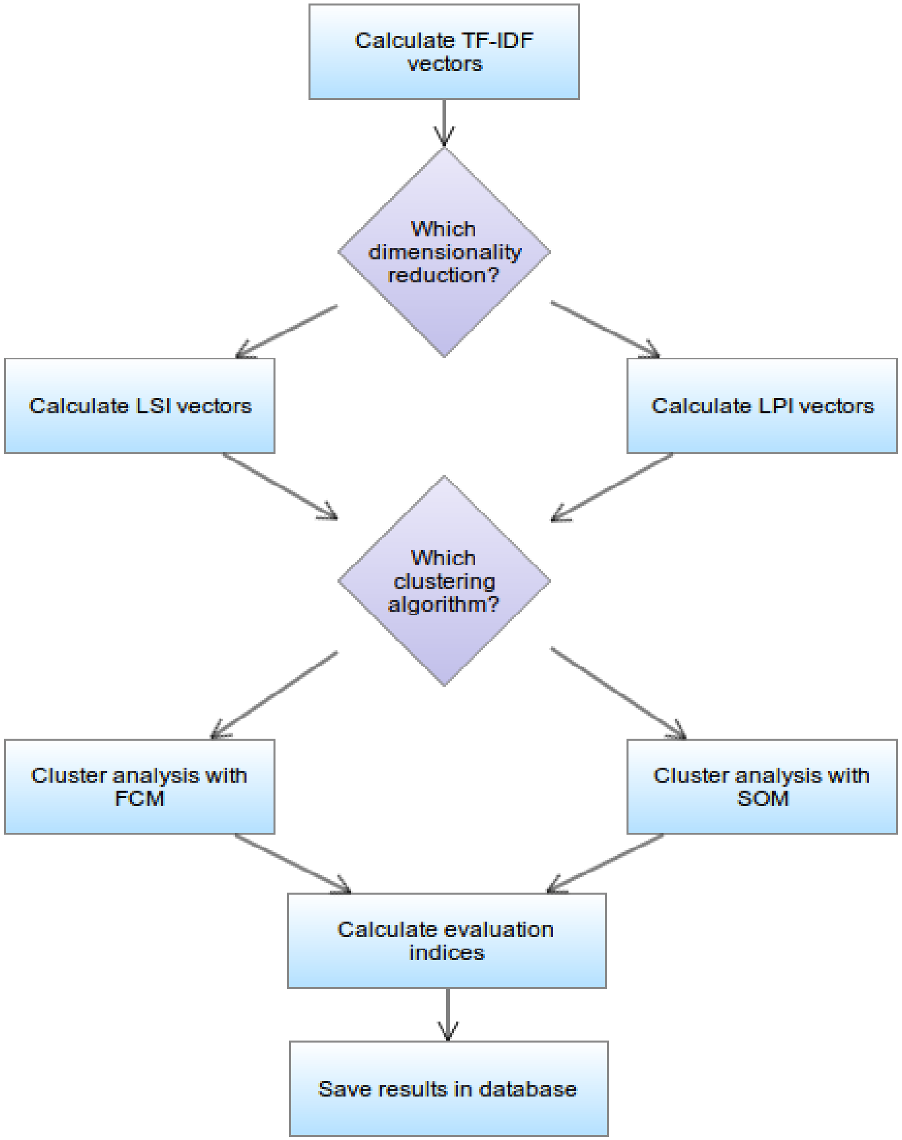
Agenda

- Introduction / goals
- MOOC clustering application
- Clustering process
- Evaluation
- Conclusions & future work

System integration



Cluster analysis process



Agenda

- Introduction / goals
- MOOC clustering application
- Clustering process
- Evaluation
- Conclusions & future work

Vector representation

- Consider the MOOC textual descriptions as „bag of words“ (→ each dimension represents one term) [~22,000 dimensions]
- Normalization by TF-IDF
- Reduce number of dimensions of the vectors with
  - *Latent Semantic Indexing* (LSI) or
  - *Locality Preserving Indexing* (LPI)

  [~10 dimensions]
- Insights:
  - General term blacklist needs to be extended (e.g. filter terms like *Illinois State University* or *capstone*)
  - No clear winner between LSI and LPI

Clustering algorithms

- Fuzzy C-Means (FCM)
  - Derivative of *k-Means* using fuzzy sets
  - Cluster centers are initialized randomly and are improved iteratively by calculating a weighted mean of each cluster
- Challenges with FCM
  - Results of FCM highly depend on the initialization
  - Solution: run FCM multiple times, return best result
  - Even after dimension reduction: *concentration of norm phenomenon*
- Meta-parameters:
  - *c* – Number of clusters
  - *m* – „fuzzyness“ parameter

Clustering algorithms

- Self-organizing Maps (SOM)
  - SOM is a type of artifical neural network
  - Map = two-dimensional grid of neurons
  - Each neuron holds a weight vector that represents it's position in the input data vector space (→ with dimension higher than two!)
  - Self-organization:
    - Input vectors are propagated through the map
    - For each vector, the nearest neuron is determined (the *winning* neuron)
    - The weights of the winning neuron and the winning neuron's neighbours (on the map) are adjusted

Clustering algorithms

- Insights on SOM
  - SOM is less dependend on initialization than FCM
  - SOM performs generally better than FCM
- Meta-parameters of SOM
  - *N x M* – map dimensions (corresponds to number of clusters)
  - α – initial learning parameter
  - δ – initial neighbourhood radius

Agenda

- Introduction / goals
- MOOC clustering application
- Clustering process
- Evaluation
- Conclusions & future work

Internal Evaluation

- Internal evaluation: calculate „validity index“ using only the input vectors and the found clusters
- No external information is used
- The validity index computes a real number, which represents the quality of a clustering
- Aim of internal evaluation: tweak meta-parameters
- Method: compute clusterings for all values of the meta-parameter within a suitable range
  → the clustering with the best index value is selected
  → this determines the value of the meta-parameter
- Validity indices might be biased against one algorithm
  → one should *not* use internal validity indices to compare two clustering algorithms

Internal evaluation: Validity Indices

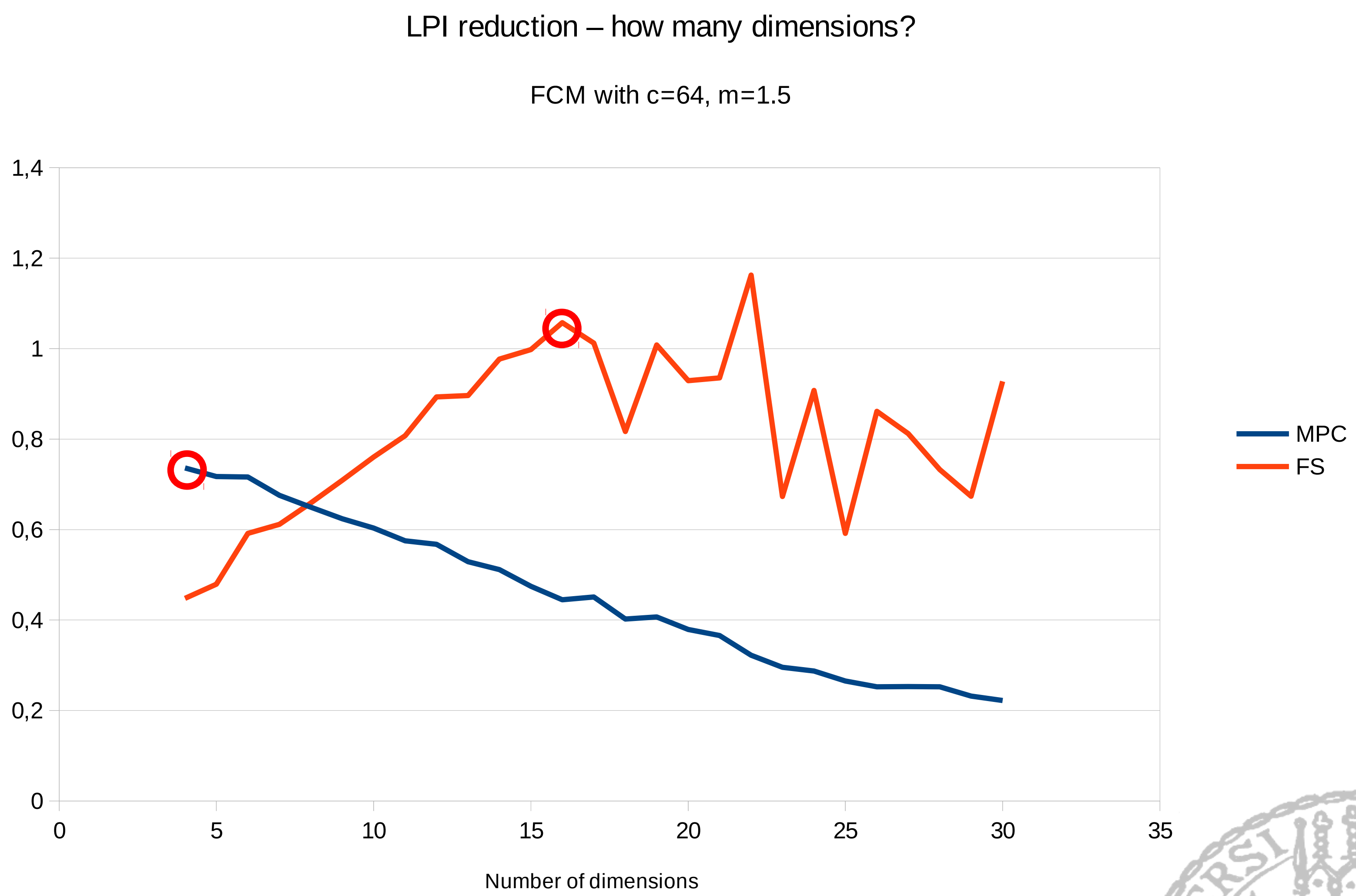
- Defining „good“ clusters is to some extent subjective
  → There are many different validity indices available
- Validity indices measure the *compactness* and *separation* of clusters
- One exemplary index: *Dunn index*

$$Dunn(C_1, \cdots, C_k) := \frac{\min_{1 \leq i < j \leq k} distance(C_i, C_j)}{\max_{1 \leq i \leq k} diameter(C_i)}$$

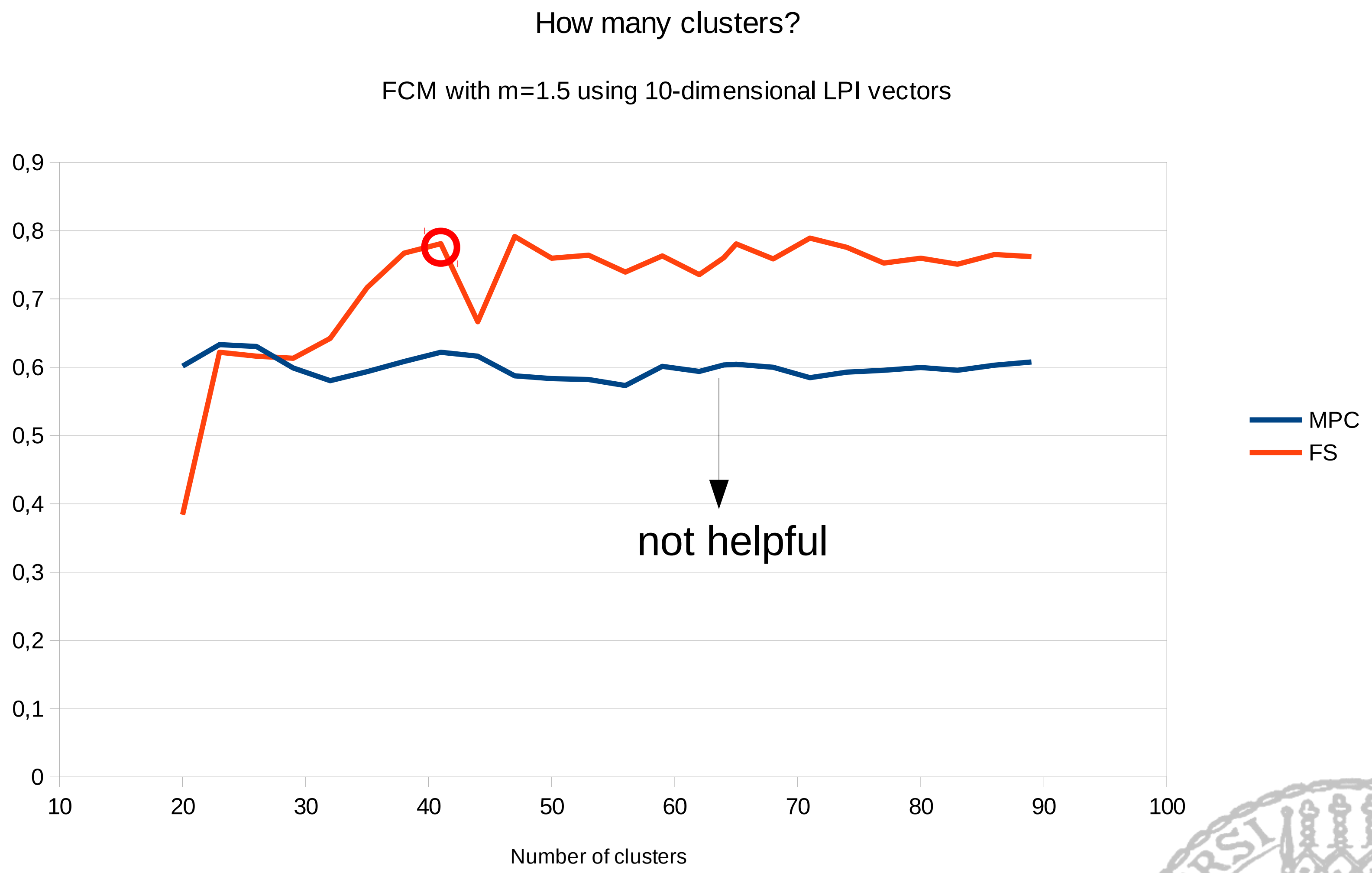
$$diameter(C_i) := \max_{x, y \in C_i} \|x - y\|$$

$$distance(C_i, C_j) := \min_{x \in C_i, y \in C_j} \|x - y\|$$

Exemplary results



Exemplary results



External Evaluation

- Use additional, external information
- Create clusters manually as „golden standard“ (in the following, these clusters are called *classes*)
- Compare clusterings with the manually created one
- *Purity*:
  - Assign each cluster to the class, which is most frequent in the cluster
  - Count the number of correctly assigned input vectors
- Downside:
  - „Golden standard“ created by only one single person → very subjective
  - This method is hardy applicable for fuzzy clustering

Agenda

- Introduction / goals
- MOOC clustering application
- Clustering process
- Evaluation
- Conclusions & future work

Conclusions

- SOM performed generally better than FCM on our data
- Even with small *m*, FCM was too fuzzy (e.g. one MOOC belongs to too many clusters)
- FCM has problems with vectors of higher dimension
- SOM worked *better* with vectors of higher dimension
- Internal evaluation has strong limits
  - Evaluation indices sometimes contradict each other
  - Which index is suitable? → hard to decide
- External evaluation needs more feedback by different users (→ see future work)

Future Work

- Use more data (syllabus, category)
- Smarter initialization for FCM
- Other distance functions except Euclidean, different vector representations
- How do the clusters change over time?
- Utilize user feedback:
  - Create ranking *within* each cluster
  - Semi-supervised clustering: improve clusters using the user feedback
  - Use the feedback for external evaluation

SOM – Further Details

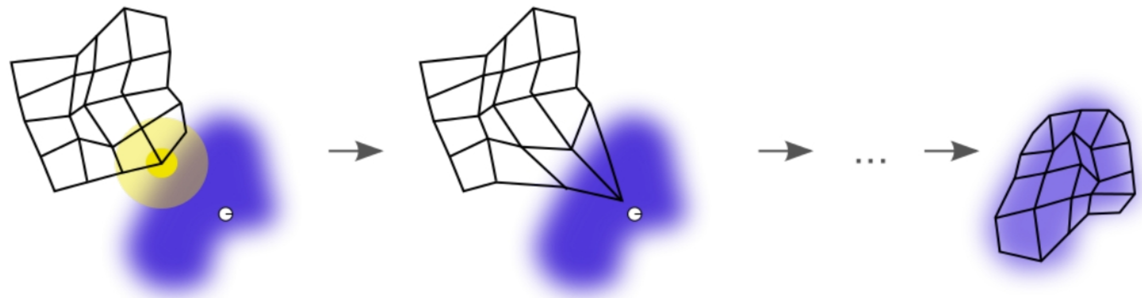
1. **Find winning neuron** Let j' be the neuron whose weight vector is closest to the input vector (using Euclidean distance):

$$j' := \arg\min_j \|x - w_j(t)\|$$

2. **Update all weight vectors** using the following formula:

$$w_j(t+1) := w_j(t) + \alpha(t) \cdot \eta_{j,j'}(t) \cdot [x - w_j(t)]$$

where $\alpha(t)$ is the learning rate and $\eta_{j,j'}(t)$ the neighbourhood function.



(Image source: Wikipedia)

SOM – Further Details

$$\eta_{j,j'}(t) := \exp\left(-\frac{\|c_j - c_{j'}\|}{2 \cdot \sigma(t)^2}\right)$$