Some Submodular Function Optimization Problems in Image Segmentation

Qiaoliang Li,
Joint work with Zili Peng and Shaojun Qu

College of Mathematics and Statistics, Hunan Normal University

The 10th Cross-Strait on Graph Theory and Combinatorics

Content

Image Segmentation

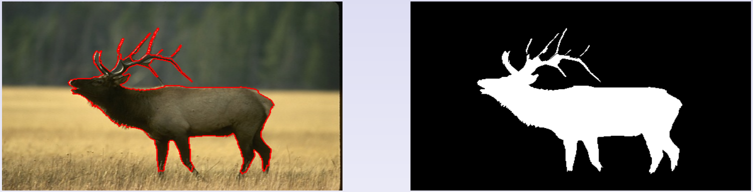


Figure: An Example of Image Segmentation

Image Segmentation

- Image segmentation is the separation of the target region corresponding to the object of the real world from the background of the image.
- The target region is based on the needs of specific applications and usually conforms to the subjective cognition and experience of the operator
- Without image segmentation, it is hard to switch into image analysis from image processing and get further image understanding.

Submodular Function-Definitions

Definition 1

A function $f : 2^N \rightarrow R$ *is submodular if for any* $S, T \subseteq N$,

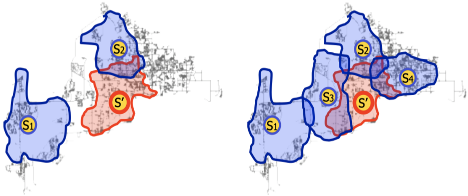
$$f(S \cup T) + f(A \cap T) \leq f(S) + f(T)$$

Definition 2

(decreasing marginal values) For any $A \subseteq B \subseteq N$ *and* $x \in N \backslash B$,

$$f(B \cup \{x\}) - f(B) \leq f(A \cup \{x\}) - f(A).$$

Submodular Function-definitions



(a) *Adding s′ to set* $\{s_1, s_2\}$ (b) *Adding s′ to superset* $\{s_1, \ldots, s_4\}$

Figure: An Illustration of the diminishing returns effect[1]

Submodular Function-definitions

Definition 3

Consider a function $f : 2^N \to R_{\geq 0}$.

- f *is monotone if* $f(A) \leq f(B)$ *for every two sets* $A \subseteq B \subseteq N$.
- f *is symmetric if* $f(A) = f(N \backslash A)$ *for every set* $A \subseteq N$.
- f *is normalized if* $f(\phi) = 0$

Submodular Function-Examples

Example 1

Cut Function: *Let* $G = (V, E)$ *be a directed graph with capacities* $c_e \geq 0$ *on the edges. For every subset of vertices* $A \subseteq V$, *let* $\delta(A) = \{e = uv | u \in A, v \in V \backslash A\}$. *The cut capacity function is defined as the total capacity of edges that cross the cut* $(A, V \backslash A)$. *Formally, the cut function is defines as*

$$f(A) = \sum_{e \in \delta(A)} c_e$$

Submodular Function-Examples

- (**Max** $k-$ **Coverage**) A number k and a collection of sets $S = \{S_1, S_2, \cdots, S_m\}$, Find a subsets $S' \subseteq S$ of sets, such that $|S'| \leq k$ and the number of covered elements $|\bigcup_{S_i \in S'} S_i|$ is maximized.
- (Weighted Coverage Function) Fix a set X, a nonnegative modular function $g : 2^X \to R$ and a collection V of subsets of X. Then for a subcollection $S \subseteq V$, the function

$$f(S) := g(\bigcup_{v \in S} v) = \sum_{x \in \cup_{v \in S} v} w(x),$$

  is monotone submodular. When $g(A) = |A|$, it is the well-known max-cover problem. $f(S)$ is submodular even for arbitrary submodular function g.

Submodular Function-Examples

- (**The Rank Function of a Matroid**) A matroid is a pair $(V, \mathcal{I})$ such that V is a finite set, and $\mathcal{I} \subseteq 2^V$ is a collection of subsets of V satisfying the following two properties:
  (1) $A \subseteq B \subseteq V$ and $B \in \mathcal{I}$ implies $A \in \mathcal{I}$;
  (2) $A, B \in \mathcal{I}$ and $|B| > |A|$ implies $\exists e \in B \backslash A$ such that $A \cup \{e\} \in \mathcal{I}$.
  Sets in $\mathcal{I}$ are called independent, and matroids generalize the concept of linear independence in linear algebra. Rank function $f(S) := \max\{|U| : U \subseteq S, U \in \mathcal{I}\}$. The rank function of any matroid is monotone submodular [2].

Submodular Function-Examples

- (**Facility Location**) Suppose we wish to select, out of a set $V = \{1, 2, \cdots, n\}$, some locations to open up facilities in order to serve a collection of m customers. If we open up a facility at location j, then it provides service of value M_{ij} to customer i, where $M \in R^{m \times n}$. If each customer chooses the facility with highest value, the total value provided to all customers is modeled by the set function

$$f(S) = \sum_{i=1}^{m} \max_{j \in S} M_{i,j}.$$

Here $f(\Phi) = 0$. If $M_{i,j} > 0$ for all i, j, then $f(S)$ is monotone submodular [3].

Submodular Function-Examples

- (**Entropy**) Given a joint probability distribution $P(X)$ over a discrete-valued random vector $X = [X_1, X_2, \cdots, X_n]$, the function $f(S) = H(X_S)$ is monotone submodular [4], where H is the Shannon entropy, i.e.,

$$H(X_S) = -\sum_{X_S} p(x_S) \log_2 P(x_S)$$

where we use the notational convention that X_S is the random vector consisting of the coordinates of X indexed by S, and likewise x_S is the vector consisting of the coordinates of an assignment x indexed by S.

Minimizing submodular functions

- Let EO be the maximum amount of time it takes to evaluate $f(S)$ for subset $S \subseteq V$.
- Grötschel et. al. [5], [6] showed that a set U minimizing $f(S)$ can be found in strongly polynomial time, if f is given by a value-giving oracle, that is, an oracle that returns $f(U)$ for any given $U \subseteq V$.
- Schrijver [7] developed an submodular function minimization algorithm that proved runs in $O(n^8 EO + n^9)$ time.
- Vygen [8] improved the run time analysis of Schrijver's algorithm and showed that the running time is $O(n^7 EO + n^8)$.
- Iwata, Fleischer and Fujishige [9] presented two algorithms that run in $O(n^5 EO \log M)$ and $O(n^7 EO \log n)$ time respectively

Minimizing submodular functions

- Fleischer and Iwata [10] gave an alternative to Schrijver's algorithm that runs in $O(n^7 EO + n^8)$ time.
- Iwata[11] developed a scaling based algorithm whose running time is $O(n^4 EO \log M + n^5 \log M)$.
- In 2008, Iwata[12] gave a new submodular function minimization algorithm, the running time is $O(n^6 EO + n^7)$
- In 2009, Orlin[13] gave a combinatorial algorithm that runs in $O(n^5 EO + n^6)$.

Maximizing submodular functions

Unconstrained Submodular Maximiztion (USM) has been studied since the sixties in the operations research community. USM captures NP-hard problems, the research concentrate on:

- Solve special cases of the problem.
- Provide exact algorithms whose times complexity cannot be efficiently bounded.
- Provide efficient algorithms whose output has no provable guarantee.

Maximizing submodular functions

- In 1962, V. P. Cherenin[14] , Solving some combinatorial problems of optimal planning by the method of successive calculations.
- The first rigorous study of USM was conducted by Feige, Mirrokni, and Vondr$\acute{a}$k [15]. They attempts to design efficient algorithms and prove the corresponding lower-bounds for unconstrained maximization of non-negative functions that are not necessarily monotone. They propose a (2/5)-approximation algorithm he also proved that the optimal approximation factor in this case is 1/2.

Maximizing submodular functions

- The hard results (optimal approximation factors) in both monotone and non-monotone cases are re-derived using a joint framework by Vondr$\acute{a}$k[16] in which the so-called multilinear continuous extension of the set functions is used.
- Buchbinder et al. [17](2015) gave a (1/3)-approximation deterministic and (1/2)-approximation randomized algorithms for unconstrained maximization of nonnegative submodular functions.

Maximizing submodular functions

- For many special cases of USM, better approximation factors are known. For example, Goemans and Williamsom[18] provides a 0.878-approximation for Max-Cut based on a semidefinite programming formulation.
- Ageev and Sviridenko [19] provide an approximation of 0.828 for the maximum facility location problem.

Maximizing submodular functions

- For Monotome submodular function subject to a matroid constraint, Calinescu et. al [20]. provide a randomized $(1 - 1/e)$-approximation for any monotone submodular function and an arbitrary matroid.
- Lee et. al[21] consider the problem of maximizing non-monotone submodular functions under matroid or knapsack constraints.

Submodular Function and Image Segmentation

- Many of the problems that arise in computer vision, especially, image segmentation, can be naturally expressed in terms of energy minimization.
- One of the most important method in solving energy minimization problems is based on graph cut. The basic technique is to construct a specialized graph for the energy function to be minimized such that the minimum cut on the graph also minimizes the energy. The minimum cut can be computed very efficiently by max flow algorithms[22].

Submodular Function and Image Segmentation

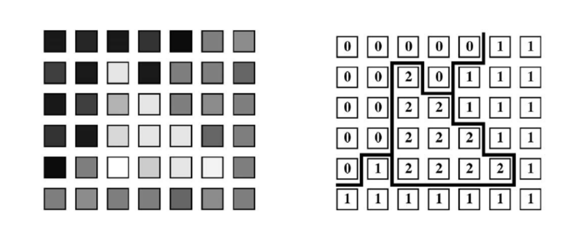


Figure: An example of image labeling.

Submodular Function and Image Segmentation

- Grig et. al were the first to discover the powerful min-cut/max flow algorithms from combinatorial optimization can be used to minimize certain important energy functions in vision.
- The energies addressed by Greig et al. and by most later graph-based methods can be represented as

$$E(L) = \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q) \quad (1)$$

- where $L = \{L_p | p \in \mathcal{P}\}$ is a labeling of image $\mathcal{P}$, $D_p(\cdot)$ is a data penality function, $V_{p,q}$ is an interaction potential, N is a set of all pairs of neighboring pixels.

Submodular Function and Image Segmentation

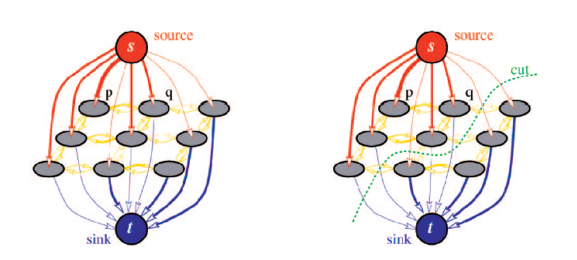


Figure: An example of directed capacitated graph, (a) A graph $\mathcal{G}$. (b) A cut on $\mathcal{G}$.

Submodular Function and Image Segmentation

- Let $\{x_1, x_2, \cdots, x_n\}, x_i \in \{0, 1\}$ be a set of binary-valued variables. We define the class $\mathcal{F}^2$ to be functions that can be written as a sum of functions of up to two binary variables at a time

$$E(x_1, x_2, \cdots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j).$$

- We define the class $\mathcal{F}^3$ to be functions that can be written as a sum of functions of up to three binary variables at a time,

$$E(x_1, x_2, \cdots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j) + \sum_{i<j<k} E^{i,j,k}(x_i, x_j, x_k)$$

- Obviously, the class $\mathcal{F}^2$ is a strict subset of the class $\mathcal{F}^3$.

Submodular Function and Image Segmentation

Definition 4

[22] A function E of n binary variables is called graph-representatable if there exists a graph $\mathcal{G} = (V, E)$ with the terminals s and t and a subset of vertices $\mathcal{V}_0 = \{v_1, v_2, \cdots, v_n\} \subset V - \{s, t\}$ such that, for any configuration $x_1, x_2, \cdots, x_n$, the value of the energy $E(x_1, x_2, \cdots, x_n)$ is equal to a constant plus the cost of the minimum $s-t$-cut among all cuts $C = S, T$ in which $v_i \in S$ iif $x_i = 0$, and $v_i \in T$, if $x_i = 1 (1 \leq i \leq n)$, We say that E is exactly represented by $\mathcal{G}, V_0$ if the constant is zero.

Submodular Function and Image Segmentation

Lemma 1

[22] the energy function E is graph-representable by a graph $\mathcal{G}$ and a subset V_0. Then, it is possible to find the exact minimum of E in polynomial time by computing the minimum $s-t-cut$ on $\mathcal{G}$.

Submodular Function and Image Segmentation

Lemma 2

($\mathcal{F}^2$ Theorem)[22] Let E be a function of n binary variables from the class of $\mathcal{F}^2$, i.e.,

$$E(x_1, x_2, \cdots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j).$$

Then, E is graph-representable if and only if each term $E^{i,j}$ satisfies the inequality

$$E^{i,j}(0, o) + E^{i,j}(1, 1) \leq E^{i,j}(0, 1) + E^{i,j}(1, 0)$$

Namely E is a submodular function

Submodular Function and Image Segmentation

Theorem 1

[22] Let E^2 be a nonregular function of two variables. Then, minimizing function of the form

$$E(x_1, x_2, \cdots, x_n) = \sum_i E^i(x_i) + \sum_{(i,j)\in\mathcal{N}} E^2(x_i, x_j),$$

where E^i are arbitrary function of one variable and $\mathcal{N} \subset \{(i,j)|i \leq i < j \leq n\}$ is NP-hard.

Definition 5

[22] Let $E(x_1, x_2, \cdots, x_n)$ *be a function of* n *binary variables and let* I, J *be a disjoint partition of the set of indices* $\{1, 2, \cdots, n\}$; $I = \{i(1), \cdots, i(m)\}$, $J = \{j(1), \cdots, j(n-m)\}$. *Let* $\alpha_{i(1)}, \cdots, \alpha_{i(m)}\}$ *be binary constants. A projection* $E' = E(x_{i(1)} = \alpha_{i(1)}, \cdots, x_{i(m)} = \alpha_{i(m)}]$ *is a function of* $n-m$ *variables defined by*

$$E'(x_{j(1)}, \cdots, x_{j(n-m)}) = E(x_1, \cdots, x_n),$$

where $x_i = \alpha_i$ *for* $i \in I$. *We say that we fix the variables* $x_{i(1)}, \cdots, x_{i(m)}$.

Submodular Function and Image Segmentation

Definition 6

[22]

- *All functions of one variable are regular.*
- *A function E of two variables is called regular if $E(0,0) + E(1,1) \leq E(0,1) + E(1,0)$.*
- *A function E of more than two variables is called regular if all projection of E of two variables are regular.*

Submodular Function and Image Segmentation

Theorem 2

[22] Let E be a function of n binary variables from $\mathcal{F}^3$, i.e.,

$$E(x_1, x_2, \cdots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j) + \sum_{i<j<k} E^{i,j,k}(x_i, x_j, x_k)$$

Then, E is graph-representable if and only if E is regular.

Theorem 3

[22] (regularity) Let E be a function of binary variables. If E is not regular, then E is not graph-representable.

Submodular Function and Image Segmentation

- How to construct meaningful energy function?
- How to approximate submodular function?
- How ti design approximation algorithm for minimizing or maximizing the submodular functions (unconstrained or constrained)?

Submodular Function and Image Segmentation

- The graph cut method is prone to cause the isolated points problem due to the lack of the distance information between pixels and seeds.
- Our method makes full use of color information and geodesic distance information; the method does not need iterative process, and can give good results in many cases.
- Besides, the approach can continue to interact to improve the previous segmentation results after getting poor segmentations in the condition of inadequate seed points marked.

Our method

- Our approach incorporates geodesic-distance and appearance overlap constraints properly in the graph-cut optimization framework:

$$E(L) = \lambda \cdot \sum_{x_i \in I} R_i(L_i) + \sum_{(x_i, x_j) \in N} B_{i,j}(L_i, L_j)$$

.

- The coefficient $\lambda \geq 0$ specifies a relative importance of regional term R_i in comparison with boundary term $B_{i},_{j}$.
- $L = (L_i)$ is a binary vector whose components L_i specify assignments to pixels x_i in I (i.e., $L_i = 0$ if x_i is a background pixel and $L_i = 1$ if x_i is a foreground object pixel).
- I is the set of pixels in the image, and N is the set of adjacent pixel pairs.

Our method

- For weighting of the relative importance of each term, the regional term can be expressed as:
- x_i represents a pixel in a image, α_1 and α_2 are used to specify the importance of $M_l(x_i)$ and $G_l(x_i)$ separately;

Our method

- $s_l(x_i)$ is a term to represent user strokes,
- $M_l(x_i)$ is a global color model, and $G_l(x_i)$ is based on the geodesic distance . Namely,

$$s_l(x_i) = \begin{cases} \infty & \text{if } x_i \in \Omega_{\bar{l}} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

$$M_l(x_i) = P_{\bar{l}}(C(x_i)),\ G_l(x_i) = \frac{D_l(x_i)}{D_F(x_i) + D_B(x_i)}, \tag{3}$$

Our method

- Ω_l is the set of seeds with label $l \in \{F, B\}$; $C(x_i) = c$ is the color at pixel x_i; $P_l(C(x_i))$ is the color model at x_i for the label l, and can be computed by normalizing the color probability density function (PDF) $Pr(c|l)$ of Ω_l, mathematically,

$$P_l(c) = \frac{\Pr(c|l)}{\Pr(c|F) + \Pr(c|B)}; \tag{4}$$

$D_l(x_i)$ is the geodesic distance that is defined as the smallest geodesic distance $d_l(s, x_i)$ from pixel x_i to the foreground (F) and background (B) seeds s in form of:

$$D_l(x_i) = \min_{s \in \Omega_l} d_l(s, x_i), \tag{5}$$

Our method

- The boundary term of our method is expressed by using the combination of the appearance overlap penalty in OneCut with the usual smoothing terms:

$$B(x_i, x_j) = -\beta \left\| \theta^L - \theta^{\bar{L}} \right\|_{L_1} + |\partial S| \,, \tag{6}$$

- $L \subset \Omega$ is a segment, and $\overline{L} = \Omega \backslash L$, θ^L and $\theta^{\bar{L}}$ are the unnormalized color histograms inside object L and background $\bar{L}$, $\|\cdot\|_{L_1}$ is the L_1 norm, β is the weight of appearance overlap penalty;

Our method

- The second term is a commonly used contrast-sensitive smoothness term as:

$$|\partial S| = \sum \omega_{x_i,x_j} \left| l_{x_i} - l_{x_j} \right| \text{ with } \omega_{x_i,x_j} = \exp \frac{-\Delta I^2}{2\sigma^2} \Big/ d, \quad (7)$$

- l_{x_i} is the label for pixel x_i, σ^2 is the average ΔI^2 over the image, d is the distance between pixel x_i and x_j.
- According to GeoGC, OneCut, and Kolmogorov 2004 above, our objective energy function is submodular that can be optimized with graph-cut.

Our method

- As the irrational weight of the appearance overlap penalty term (β) used in OneCut algorithm prone to cause the problem of isolated points, we need to adjust the value of β properly according to the ambiguity degree of the estimated appearance model.
- an error term ε,

$$\varepsilon = \frac{1}{2}\left[\frac{\sum_{x\in F} P_B(C(x))}{|\Omega_F|} + \frac{\sum_{x\in B} P_F(C(x))}{|\Omega_B|}\right]. \qquad (8)$$

Our method

- there is no error ($\varepsilon = 0$), we would like to give the appearance overlap term a big weight; when the color models become more indistinct (i.e. the foreground and background color models will overlap more and more), the error ε will grow and we would like to provide less weight to the appearance overlap term:

$$\beta_{t+1} = \begin{cases} 0.25 \times \beta_t & \text{if } \varepsilon > 0.9 \\ (1-\varepsilon) \times \beta_t & \text{otherwise} \end{cases} \quad with\ \beta_0 = 1.4. \tag{9}$$

Our method

The constraints of color models and geodesic distance α

- Empirically, we set α according to the error term ε computed in the previous section:

$$\kappa = \begin{cases} 1 - 2\varepsilon & \text{if } \varepsilon < 0.5 \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

$$\alpha_1 = 80 \times (1 - \kappa),\ \alpha_2 = 800 \times \kappa. \tag{11}$$

- κ can be viewed as the confidence of the estimated appearance model. the number of bins automatically when using L_1 norm to approximate the appearance overlap term: 128 for small images; medium-sized images (used in this letter), take 64; and 16 for large images.

Our method

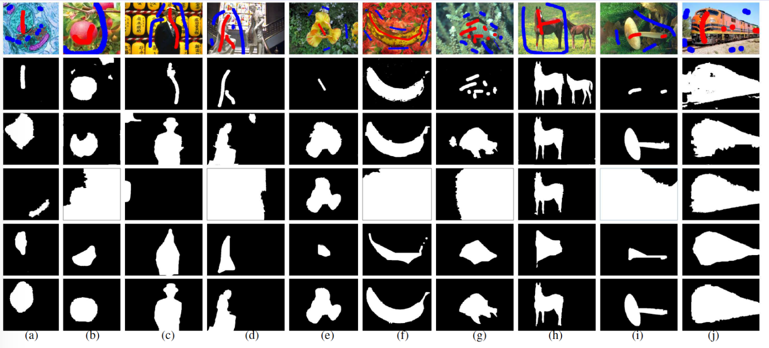


Figure: Comparison results with the same user scribbles. The first row is the original images with user scribbles, 2-6 rows are the segmentation masks of OneCut, GeoGC, SSNCuts, TRC and Ours respectively

Our method

1. Zili Peng, Shaojun Qu and Qiaoliang Li, Interactive image segmentation using geodesic appearance overlap graph cut, Signal Processing: Image Communication 78 (2019) 159-170

2. Zili Peng and Qiaoliang Li, Adaptive appearance separation for interactive image segmentation based on Dense CRF, IET Image Processing, 13(2019),, 142-151

3. Shaojun Qu, Qiaoliang Li and Ming Chen, Supervised image segmentation based on superpixel and improved normalised cuts, IET Image Processing, doi: 10.1049/iet-ipr.2018.6241

References I

- A. Krause and D. Golovin. Submodular Function Maximization. Tractability: Practical Approaches to Hard Problems. 2011.
- G. Birkhoff. On the Combination of Subalgebras. Cambridge Philosophical Society, 29(1933),441-464.
- A. M. Frieze, A cost function property for plant location problems. Mathematical Programming, 7(1974), 245-248.
- S. Fujishige, Polymatroidal Dependence Structure of a Set of Random Variables. Inform. Contr., 39(1978), 55-72.
- M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (1981), 169-197. [Corrigendum: Combinatorica 4 (1984) 291-295]

- M. Grötschel, L. Lovász, and A. Schrijver, "Geometric Algorithms and Combinatorial Optimization," Springer-Verlag, Berlin, 1988.
- A. Schrijver ,A combinatorial algorithm minimizing submodular functions in strongly polynomial time.J. Combin. Theory (B) 80(2000), 346-355.
- J. Vygen, A note on Schrijver's submodular function minimization algorithm, J. of Combinatorial Theory, B88(2003), 399-402.
- S. Iwata, L. Fleischer and S. Fujishige, A combinatorila , strongly polynomial-time algorithm for minimizing submodular functions, J. of ACM, 48(2001), 761-777.

References III

- L. Fleischer, S. Fujishige and S. Iwata. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions, Journal of the ACM 48:4 (2001), 761-777.
- S. Iwata, A faster scaling algorithm for minimizing submodular functions, SIAM J. Comput. 32(2002), 833-840.
- S. Iwata, Submodular function minimization, Math. Program. 112(2008), 45-64.
- J. B. Orlin, A faster strongly polynomial time algorithm for submodular function minimization, Math. Program, A118(2009), 237-251.

References IV

- V. Cherenin. Solving some combinatorial problems of optimal planning by the method of successive calculations, Proc. of the Conference of Experiences and Perspectives of the Applications of Mathematical Methods and Electronic Computers in Planning (in Russian), Mimeograph, Novosibirsk (1962).
- U. Feige, V. S. Mirrokni, and J. Vondr*á*k, Maximizing non-monotone submodular functions, SIAM J. Comput., 40 (2011), 1133-1153.
- J. Vondr*á*k, Symmetry and approximability of submodular maximization problems, SIAM J. on Comput., 42(2013), 265-304.
- N. Buchbinder, M. Feldman, J. Naor, A tight linear time (1/2)-approximation for unconstrained submodular maximization, SIAM J. Comput., 44(2015), 1384-1402.

References V

- M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satis?ability problems using semidefinite programming, J. ACM, 42 (1995), pp. 1115-1145.
- A. A. Ageev and M. I. Sviridenko, An 0.828 approximation algorithm for the uncapacitated facility location problem, Discrete Appl. Math., 93 (1999), pp. 149?156.
- G. Calinescu, C. Chekuri, M. Pal and J. Vondrák, Maximizing a monotone submodular function subject to a matroid constraint, SIAM J. Comput., 40(2011), 1740-1766.
- Jon Lee, S. Vahab, Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. SIAM Journal on Discrete Mathematics, 23(2010):2053?2078.

References VI

V. Kolmogorov and R. Zabih, What energy functions can be minimized via graph cuts? IEEE Trans. on Pattern Analysis and Machine Intelligence, 26(2004), 147-158.

Thank you!