Parameter Learning for Log-supermodular Distributions

Authored by:

Francis Bach
Tatiana Shpakova

Abstract

We consider log-supermodular models on binary variables, which are probabilistic models with negative log-densities which are submodular. These models provide probabilistic interpretations of common combinatorial optimization tasks such as image segmentation. In this paper, we focus primarily on parameter estimation in the models from known upper-bounds on the intractable log-partition function. We show that the bound based on separable optimization on the base polytope of the submodular function is always inferior to a bound based on "perturb-and-MAP" ideas. Then, to learn parameters, given that our approximation of the log-partition function is an expectation (over our own randomization), we use a stochastic subgradient technique to maximize a lower-bound on the log-likelihood. This can also be extended to conditional maximum likelihood. We illustrate our new results in a set of experiments in binary image denoising, where we highlight the flexibility of a probabilistic model to learn with missing data.

1 Paper Body

Submodular functions provide efficient and flexible tools for learning on discrete data. Several common combinatorial optimization tasks, such as clustering, image segmentation, or document summarization, can be achieved by the minimization or the maximization of a submodular function [1, 8, 14]. The key benefit of submodularity is the ability to model notions of diminishing returns, and the availability of exact minimization algorithms and approximate maximization algorithms with precise approximation guarantees [12]. In practice, it is not always straightforward to define an appropriate submodular function for a problem at hand. Given fully-labeled data, e.g., images and their foreground/background segmentations in image segmentation, structured-output prediction methods such as the structured-SVM may be used [18]. However, it is common (a) to have missing data, and (b) to embed submodular function minimization within a larger model. These are two situations well tackled by

probabilistic modelling. Log-supermodular models, with negative log-densities equal to a submodular function, are a first important step toward probabilistic modelling on discrete data with submodular functions [5]. However, it is well known that the log-partition function is intractable in such models. Several bounds have been proposed, that are accompanied with variational approximate inference [6]. These bounds are based on the submodularity of the negative log-densities. However, parameter learning (typically by maximum likelihood), which is a key feature of probabilistic modeling, has not been tackled yet. We make the following contributions: ? In Section 3, we review existing variational bounds for the log-partition function and show that the bound of [9], based on ?perturb-and-MAP? ideas, formally dominates the bounds proposed by [5, 6]. ? In Section 4.1, we show that for parameter learning via maximum likelihood the existing bound of [5, 6] typically leads to a degenerate solution while the one based on ?perturb-and-MAP? ideas and logistic samples [9] does not. 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

? In Section 4.2, given that the bound based on ?perturb-and-MAP? ideas is an expectation (over our own randomization), we propose to use a stochastic subgradient technique to maximize the lower-bound on the log-likelihood, which can also be extended to conditional maximum likelihood. ? In Section 5, we illustrate our new results on a set of experiments in binary image denoising, where we highlight the flexibility of a probabilistic model for learning with missing data.

2

Submodular functions and log-supermodular models

In this section, we review the relevant theory of submodular functions and recall typical examples of log-supermodular distributions. 2.1

Submodular functions

We consider submodular functions on the vertices of the hypercube {0, 1}D . This hypercube representation is equivalent to the power set of {1, . . . , D}. Indeed, we can go from a vertex of the hypercube to a set by looking at the indices of the components equal to one and from set to vertex by taking the indicator vector of the set. For any two vertices of the hypercube, x, y ? {0, 1}D , a function f : {0, 1}D ? R is submodular if f (x) + f (y) ¿ f (min{x, y}) + f (max{x, y}), where the min and max operations are taken component-wise and correspond to the intersection and union of the associated sets. Equivalently, the function x 7? f (x + ei) ? f (x), where ei ? RD is the i-th canonical basis vector, is non-increasing. Hence, the notion of diminishing returns is often associated with submodular functions. Most widely used submodular functions are cuts, concave functions of subset cardinality, mutual information, set covers, and certain functions of eigenvalues of submatrices [1, 7]. Supermodular functions are simply negatives of submodular functions. In this paper, we are going to use a few properties of such submodular functions (see [1, 7] and references therein). Any submodular function f can be extended from {0, 1}D to a convex function on RD , which is called the Lov?sz extension. This extension has the same value on {0, 1}D , hence we use the same notation f . Moreover, this function

is convex and piecewise linear, which implies the existence of a polytope B(f) ? RD , called the base polytope, such that for all x ? RD , f (x) = maxs?B(f) x¿ s, that is, f is the support function of B(f). The Lov?sz extension f and the base polytope B(f) have explicit expressions that are, however, not relevant to this paper. We will only use the fact that f can be efficiently minimized on {0, 1}D , by a variety of generic algorithms, or by more efficient dedicated ones for subclasses such as graph-cuts. 2.2

Log-supermodular distributions

Log-supermodular models are introduced in [5] to model probability distributions on a hypercube, x ? {0, 1}D , and are defined as 1 p(x) = exp(?f (x)), Z(f) where f : {0,P 1}D ? R is a submodular function such that f (0) = 0 and the partition function is Z(f) = (x)). It is more convenient to deal with the convex log-partition x?{0,1}D exp(?f P function A(f) = log Z(f) = log x?{0,1}D exp(?f (x)). In general, the calculation of the partition function Z(f) or the log-partition function A(f) is intractable, as it includes simple binary Markov random fields?the exact calculation is known to be #P -hard [10]. In Section 3, we review upperbounds for the log-partition function. 2.3

Examples

Essentially, all submodular functions used in the minimization context can be used as negative log-densities [5, 6]. In computer vision, the most common examples are graph-cuts, which are essentially binary Markov random fields with attractive potentials, but higher-order potentials have been considered as well [11]. In our experiments, we use graph-cuts, where submodular function minimization may be performed with max-flow techniques and is thus efficient [4]. Note that there are extensions of submodular functions to continuous domains that could be considered as well [2]. 2

3

Upper-bounds on the log-partition function

In this section, we review the main existing upper-bounds on the log-partition function for logsupermodular densities. These upper-bounds use several properties of submodular functions, in particular, the Lov?sz extension and the base polytope. Note that lower bounds based on submodular maximization aspects and superdifferentials [5] can be used to highlight the tightness of various bounds, which we present in Figure 1. 3.1

Base polytope relaxation with L-Field [5]

This method exploits the fact that any submodular function f (x) can be lower bounded by a modular function s(x), i.e., a linear function of x ? {0, 1}D in the hypercube representation. The submodular function and its lower bound are related by f (x) = maxs?B(f) s¿ x, leading to: P P A(f) = log x?{0,1}D exp (?f (x)) = log x?{0,1}D mins?B(f) exp (?s¿ x), which, by swapping the sum and min, is less than P PD def mins?B(f) log x?{0,1}D exp (?s¿ x) = mins?B(f) d=1 log (1 + e?sd) = AL-field (f). (1) Since the polytope B(f) is tractable (through its membership oracle or by maximizing linear functions efficiently), the bound AL-field (f) is tractable, i.e., computable in polynomial time. Moreover, it has a nice interpretation through convex duality as the logistic function log(1 + e?sd) may be represented as max?d ?[0,1] ??d sd ? ?d log ?d ? (1 ? ?d) log(1 ? ?d

), leading to: AL-field (f) = min
max ??¿ s + H(?) = max H(?) ? f (?),
s?B(f) ??[0,1]D
??[0,1]D

PD where H(?) = ? d=1 ?d log ?d + (1 ? ?d) log(1 ? ?d) . This shows in particular the convexity of f 7? AL-field (f). Finally, [6] shows the remarkable result that the minimizer s ? B(f) may be obtained by minimizing a simpler function on B(f), namely the squared Euclidean norm, thus leading to algorithms such as the minimum-norm-point algorithm [7]. 3.2

?Pertub-and-MAP? with logistic distributions

Estimating the log-partition function can be done through optimization using ?pertub-and-MAP? ideas. The main idea is to perturb the log-density, find the maximum a-posteriori configuration (i.e., perform optimization), and then average over several random perturbations [9, 17, 19]. The Gumbel distribution on R, whose cumulative distribution function is F (z) = exp(? exp(?(z + c))), where c is the Euler constant, is particularly useful. Indeed, if {g(y)}y?{0,1}D is a collection of independent random variables g(y) indexed by y ? {0, 1}D , each following the Gumbel distribution, then the random variable maxy?{0,1}D g(y) ? f (y) is such that we have

log Z(f) = Eg maxy?{0,1}D {g(y) ? f (y)} [9, Lemma 1]. The main problem is that we need 2D such variables, and a key contribution of [9] is to show that if we consider a factored collection {gd (yd)}yd ?{0,1},d=1,...,D of i.i.d. Gumbel variables, then we get an upper-bound on the log PD partition-function, that is, log Z(f) ? Eg maxy?{0,1}D { d=1 gd (yd) ? f (y)}. Writing gd (yd) = [gd (1) ? gd (0)]yd + gd (0) and using the fact that (a) gd (0) has zero expectation and (b) the difference between two independent Gumbel distributions has a logistic distribution (with cumulative distribution function z 7? (1 + e?z)?1) [15], we get the following upper-bound:

ALogistic (f) = Ez1 ,...,zD ?logistic max {z ¿ y ? f (y)} , (2) y?{0,1}D

where the random vector z ? RD consists of independent elements taken from the logistic distribution. This is always an upper-bound on A(f) and it uses only the fact that submodular functions are efficient to optimize. It is convex in f as an expectation of a maximum of affine functions of f . 3.3

Comparison of bounds

In this section, we show that AL-field (f) is always dominated by ALogistic (f). This is complemented by another result within the maximum likelihood framework in Section 4. 3

Proposition 1. For any submodular function f : {0, 1}D ? R, we have: A(f) 6 ALogistic (f) 6 AL-field (f).

(3)

Proof. The first inequality was shown by [9]. For the second inequality, we have:

ALogistic (f) = Ez max z ¿ y ? f (y) D y?{0,1}

= Ez max z ¿ y ? max s¿ y from properties of the base polytope B(f), D s?B(f) y?{0,1}

= Ez maxy?{0,1}D mins?B(f) z ¿ y ? s¿ y ,

= Ez mins?B(f) max z ¿ y ? s¿ y by convex duality, D y?{0,1}

6 mins?B(f) Ez maxy?{0,1}D (z ? s)¿ y by swapping expectation and minimization, PD

= mins?B(f) Ez d=1 (zd ? sd)+ by explicit maximization, PD

= mins?B(f) d=1 Ezd (zd ? sd)+ by using linearity of expectation, PD R +?

= mins?B(f) d=1 ?? (zd ? sd)+ P (zd)dzd by definition of expectation,

PD R +? e?zd = min (zd ? sd) (1+e ?zd 2 dzd by substituting the density function, d=1 sd) s?B(f) PD = mins?B(f) d=1 log(1 + e?sd), which leads to the desired result. In the inequality above, since the logistic distribution has full support, there cannot be equality. However, if the base polytope is such that, with high probability ?d, —sd — ? —zd —, then the two bounds are close. Since the logistic distribution is concentrated around zero, we have equality when —sd — is large for all d and s ? B(f). Running-time complexity of AL-field and Alogistic . The logistic bound Alogistic can be computed if there is efficient MAP-solver for submodular functions (plus a modular term). In this case, the divide-and-conquer algorithm can be applied for L-Field [5]. Thus, the complexity is dedicated to the minimization of O(—V —) problems. Meanwhile, for the method based on logistic samples, it is necessary to solve M optimization problems. In our empirical bound comparison (next paragraph), the running time was the same for both methods. Note however that for parameter learning, we need a single SFM problem per gradient iteration (and not M). Empirical comparison of AL-field and Alogistic . We compare the upper-bounds on the log-partition function AL-field and Alogistic , with the setup used by [5]. We thus consider data from a Gaussian mixture model with 2 clusters in R2 . The centers are sampled from N([3, 3], I) and N([?3, ?3], I), respectively. Then we sampled n = 50 points for each cluster. Further, these 2n points are used as nodes in a complete weighted graph, where the weight between points x and y is equal to e?c——x?y—— . We consider the graph cut function associated to this weighted graph, which defines a logsupermodular distribution. We then consider conditional distributions, one for each k = 1, . . . , n, on the events that at least k points from the first cluster lie on the one side of the cut and at least k points from the second cluster lie on the other side of the cut. For each conditional distribution, we evaluate and compare the two upper bounds. We also add the tree-reweighted belief propagation upper bound [23] and the superdifferential-based lower bound [5]. In Figure 1, we show various bounds on A(f) as functions of the number on conditioned pairs. The logistic upper bound is obtained using 100 logistic samples: the logistic upper-bound Alogistic is close to the superdifferential lower bound from [5] and is indeed significantly lower than the bound AL-field . However, the tree-reweighted belief propagation bound behaves a bit better in the second case, but its calculation takes more time, and it cannot be applied for general submodular functions. 3.4

From bounds to approximate inference

Since linear functions are submodular functions, given any convex upper-bound on the log-partition function, we may derive an approximate marginal probability for each xd ? {0, 1}. Indeed, following [9], we consider an exponen-

tial family model p(x—t) = exp(?f (x) + t¿ x ? A(f ? t)), where 4

200 150

Superdifferential lower bound L?field upper bound Tree?reweighted BP Logistic upper bound

Log?Partition Function

Log?Partition Function

250

100 50 0 0

Superdifferential lower bound L?field upper bound Tree?reweighted BP Logistic upper bound

100 80 60 40 20 0

20 40 Number of Conditioned Pairs

60

0

(a) Mean bounds with confidence intervals, c = 1.

10 20 30 40 Number of Conditioned Pairs

50

(b) Mean bounds with confidence intervals, c = 3.

Figure 1: Comparison of log-partition function bounds for different values of c. See text for details. f ? t is the function x 7? f (x) ? t¿ x. When f is assumed to be fixed, this can be seen as an exponential family with the base measure exp(?f (x)), sufficient statistics x, and A(f ? t) is the log-partition function. It is known that the expectation of the sufficient statistics under the exponential family model Ep(x—t) x is the gradient of the log-partition function [23]. Hence, any approximation of this log-partition gives an approximation of this expectation, which in our situation is the vector of marginal probabilities that an element is equal to 1. For the L-field bound, at t = 0, we have ?td AL-field (f ? t) = ?(s?d), where s? is the minimizer of PD ?sd), thus recovering the interpretation of [6] from another point of view. d=1 log(1 + e For the logistic bound, this is the inference mechanism from [9], with ?td Alogistic (f ? t) = Ez y ? (z), where y ? (z) is the maximizer of maxy?{0,1}D z ¿ y ? f (y). In practice, in order to perform approximate inference, we only sample M logistic variables. We could do the same for parameter learning, but a much more efficient alternative, based on mixing sampling and convex optimization, is presented in the next section.

4

Parameter learning through maximum likelihood

An advantage of log-supermodular probabilistic models is the opportunity to learn the model parameters from data using the maximum-likelihood principle. In this section, we consider that we are given N observations x1 , . . . , xN ? {0, 1}D , e.g., binary images such as shown in Figure 2. PK We consider a submodular function f (x) represented as f (x) = k=1 ?k fk (x) ? t¿ x. The modular term t¿ x is explicitly taken into account with t ? RD , and K base submodular functions are assumed to be given with ? ? RK + so that the function f remains submodular. Assuming the data x1 , . . . , xN are

independent and identically (i.i.d.) distributed, then maximum likelihood is equivalent to minimizing: N N K

1 X 1 XX min ? ?k fk (xn) ? t¿ xn + A(f) , log p(xn —?, t) = min K , t?RD N D N ??RK , t?R ??R + + n=1 n=1 k=1

which takes the particularly simple form P P

PK N N 1 ¿ 1 min??RK ? f (x) ? t x + A(?, t), D k k n n k=1 n=1 n=1 N N + , t?R

(4)

where we use the notation A(?, t) = A(f). We now consider replacing the intractable log-partition function by its approximations defined in Section 3. 4.1

Learning with the L-field approximation

In this section, we show that if we replace A(f) by AL-field (f), we obtain a degenerate solution. Indeed, we have D D X X ?sd AL-field (?, t) = min log (1 + e) = log (1 + e?sd +td). Pmin s?B(f)

s?B(

d=1

5

K k=1

?K fK)

d=1

This implies that Eq. (4) becomes K N N D 1 X X

X X ¿ 1 min ? min f (x) ? t x + log (1 + e?sd +td). k k n n PK D s?B(N N , t?R ??RK ? f) K K + k=1 n=1 n=1 k=1

d=1

hxid The minimum with respect to td may be performed in closed form with td ? sd = log 1?hxi , where d P N 1 hxi = N n=1 xn . Putting this back into the equation above, we get the equivalent problem: K N N 1 X X

X ¿ 1 min min ? f (x) ? s xn + const , k k n P K N n=1 N n=1 ??RK + s?B(k=1 ?K fK) k=1

which is equivalent to, using the representation of f as the support function of B(f): 1 PN

PK PN 1 min??RK k=1 ?k N n=1 fk (xn) ? fk N n=1 xn . + Since fk is convex, by Jensen?s inequality, the linear term in ?k is non-negative; thus maximum likelihood through L-field will lead to a degenerate solution where all ??s are equal to zero. 4.2

Learning with the logistic approximation with stochastic gradients

In this section we consider the problem (4) and replace A(f) by ALogistic (f): K K X X

min ?k hfk (x)iemp. ?t¿ hxiemp. +Ez?logistic max z ¿ y + t¿ y ? ?k f (y) , D ??RK + , t?R

y?{0,1}D

k=1

(5)

k=1

where hM (x)iemp. denotes the empirical average of M (x) (over the data). PK Denoting by y ? (z, t, ?) ? {0, 1}D the maximizers of z ¿ y + t¿ y ? k=1 ?k f (y), the objective function may be written: K X

?k hfk (x)iemp. ?hfk (y ? (z, t, ?))ilogistic ?t¿ hxiemp. ?hy ? (z, t, ?)ilogistic]+hz ¿ y ? (z, t, ?)ilogistic . k=1

This implies that at optimum, for ?k ¿ 0, then hfk (x)iemp. = hfk (y ? (z, t, ?))ilogistic , while, hxiemp. = hy ? (z, t, ?)ilogistic , the expected values of the sufficient statistics match between the data and the optimizers used for the logistic approximation [9]. In order to minimize the expectation in Eq. (5), we propose to use the projected stochastic gradient method, not on the data as usually done, but on our own internal randomization. The algorithm then becomes, once we add a weighted ‘2 -regularization ?(t, ?): ? Input: functions fk , k = 1, . . . , K, and expected sufficient statistics hfk (x)iemp. ? R and hxiemp. ? [0, 1]D , regularizer ?(t, ?). ? Initialization: ? = 0, t = 0 ? Iterations: for h from 1 to H ? Sample z ? RD as independent logistics PK ? Compute y ? = y ? (z, t, ?) ? arg max z ¿ y + t¿ y ? k=1 ?k f (y) D y?{0,1}

? Replace t by t ? ?Ch y ? ? hxiemp. + ?t ?(t, ?)

? Replace ?k by ?k ? ?Ch hfk (x)iemp. ? fk (y ?) + ??k ?(t, ?) + . ? Output: (?, t). Since our cost function is convex and ? Lipschitz-continuous, the averaged iterates are converging to the global optimum [16] at rate 1/ H (for function values). 4.3

Extension to conditional maximum likelihood

In experiments in Section 5, we consider a joint model over two binary vectors x, z ? RD , as follows D Y ?(z 6=x) p(x, z—?, t, ?) = p(x—?, t)p(z—x, ?) = exp(?f (x) ? A(f)) ?d d d (1 ? ?d)?(zd =xd) , (6) d=1

6

(a) original image

(b) noisy image

(c) denoised image

Figure 2: Denoising of a horse image from the Weizmann horse database [3].

which corresponds to sampling x from a log-supermodular model and considering z that switches the values of x with probability ?d for each d, that is, a noisy observation of x. We have:

PD log p(x, z—?, t, ?) = ?f (x) ? A(f) + d=1 ? log(1 + eud) + xd ud + zd ud ? 2xd zd ud , with ud = log

?d 1??d

which is equivalent to ?d = (1 + e?ud)?1 .

Using Bayes rule, we have p(x—z, ?, t, ?) ? exp(?f (x) ? A(f) + x¿ u ? 2x¿ (u ? z)), which leads to the log-supermodular model p(x—z, ?, t, ?) = exp(?f (x) + x¿ (u ? 2u ? z) ? A(f ? u + 2u ? z)). Thus, if we observe both z and x, we can consider a conditional maximization of the log-likelihood (still a convex optimization problem), which we do in our experiments for supervised image denoising, where we assume we know both noisy and original images at training time. Stochastic gradient on the logistic samples can then be used. Note that our conditional ML estimation can be seen as a form of approximate conditional random fields [13]. While supervised learning can be achieved by

other techniques such as structured-output-SVMs [18, 20, 22], our approach also applies when we do not observe the original image, which we now consider. 4.4

Missing data through maximum likelihood

In the model in Eq. (6), we now assume we only observed the noisy output z, and we perform parameter learning for ?, t, ?. This is a latent variable model for which maximum likelihood can be readily applied. We have: P log p(z—?, t, ?) = log x?{0,1} p(z, x—?, t, ?) QD P ?(z 6=x) = log x?{0,1}D exp(?f (x) ? A(f)) d=1 ?d d d (1 ? ?d)?(zd =xd) P D = A(f ? u + 2u ? z) ? A(f) + z ¿ u ? d=1 log(1 + eud). In practice, we will assume that the noise probability ? (and hence u) is uniform across all elements. While we could use majorization-minization approaches such as the expectation-minimization algorithm (EM), we consider instead stochastic subgradient descent to learn the model parameters ?, t and u (now a non-convex optimization problem, for which we still observed good convergence).

5

Experiments

The aim of our experiments is to demonstrate the ability of our approach to remove noise in binary images, following the experimental set-up of [9]. We consider the training sample of Ntrain = 100 images of size D = 50 ? 50, and the test sample of Ntest = 100 binary images, containing a horse silhouette from the Weizmann horse database [3]. At first we add some noise by flipping pixels values independently with probability ?. In Figure 2, we provide an example from the test sample: the original, the noisy and the denoised image (by our algorithm). We consider the model from Section 4.3, with the two functions f1 (x), f2 (x) which are horizontal and vertical cut functions with binary weights respectively, together with a modular term of dimension D. To perform minimization we use graph-cuts [4] as we deal with positive or attractive potentials. Supervised image denoising. We assume that we observe N = 100 pairs (xi , zi) of original-noisy images, i = 1, . . . , N . We perform parameter inference by maximum likelihood using stochastic subgradient descent (over the logistic samples), with regularization by the squared ‘2 -norm, one 7

noise ? 1% 5% 10% 20%
max-marg. 0.4% 1.1% 2.1% 4.2%
std ¡0.1% ¡0.1% ¡0.1% ¡0.1%
mean-marginals 0.4% 1.1% 2.0% 4.1%
std ¡0.1% ¡0.1% ¡0.1% ¡0.1%
SVM-Struct 0.6% 1.5% 2.8% 6.0%
std ¡0.1% ¡0.1% 0.3% 0.6%
Table 1: Supervised denoising results.
? 1% 5% 10% 20%
max-marg. 0.5% 0.9% 1.9% 5.3%
? is fixed std mean-marg. ¡0.1% 0.5% 0.1% 1.0% 0.4% 2.1% 2.0% 6.0%
std ¡0.1% 0.1% 0.4% 2.0%
max-marg. 1.0% 3.5% 6.8% 20.0%
? is not fixed std mean-marg. 1.0% 0.9% 3.6% 2.2% 7.0% 20.0%
std 0.8% 2.0% -

Table 2: Unsupervised denoising results. parameter for t, one for ?, both learned by cross-validation. Given our estimates, we may denoise a new image by computing the ?max-marginal?, e.g., the maximum a posteriori maxx p(x—z, ?, t) through a single graph-cut, or computing ?mean-marginals? with 100 logistic samples. To calculate the error we use the normalized Hamming distance and 100 test images. Results are presented in Table 1, where we compare the two types of decoding, as well as a structured output SVM (SVM-Struct [22]) applied to the same problem. Results are reported in proportion of correct pixels. We see that the probabilistic models here slightly outperform the max-margin formulation1 and that using mean-marginals (which is optimal given our loss measure) lead to slightly better performance. Unsupervised image denoising. We now only consider N = 100 noisy images z1 , . . . , zN to learn the model, without the original images, and we use the latent model from Section 4.4. We apply stochastic subgradient descent for the difference of the two convex functions Alogistic to learn the model parameters and use fixed regularization parameters equal to 10?2 . We consider two situations, with a known noise-level ? or with learning it together with ? and t. The error was calculated using either max-marginals and mean-marginals. Note that here, structuredoutput SVMs cannot be used because there is no supervision. Results are reported in Table 2. One explanation for a better performance for max-marginals in this case is that the unsupervised approach tends to oversmooth the outcome and max-marginals correct this a bit. When the noise level is known, the performance compared to supervised learning is not degraded much, showing the ability of the probabilistic models to perform parameter estimation with missing data. When the noise level is unknown and learned as well, results are worse, still better than a trivial answer for moderate levels of noise (5% and 10%) but not better than outputting the noisy image for extreme levels (1% and 20%). In challenging fully unsupervised case the standard deviation is up to 2.2% (which shows that our results are statistically significant).

6

Conclusion

In this paper, we have presented how approximate inference based on stochastic gradient and ?perturband-MAP? ideas could be used to learn parameters of log-supermodular models, allowing to benefit from the versatility of probabilistic modelling, in particular in terms of parameter estimation with missing data. While our experiments have focused on simple binary image denoising, exploring larger-scale applications in computer vision (such as done by [24, 21]) should also show the benefits of mixing probabilistic modelling and submodular functions. Acknowledgements. We acknowledge support the European Union?s H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement no 642685 MacSeNet, and thank Sesh Kumar, Anastasia Podosinnikova and Anton Osokin for interesting discussions related to this work. 1 [9] shows a stronger difference, which we believe (after consulting with authors) is due to lack of convergence for the iterative algorithm solving the max-margin formulation.

8

2 References

[1] F. Bach. Learning with submodular functions: a convex optimization perspective. Foundations and Trends in Machine Learning, 6(2-3):145 ? 373, 2013. [2] F. Bach. Submodular functions: from discrete to continuous domains. Technical Report 1511.00394, arXiv, 2015. [3] E. Borenstein, E. Sharon, and S. Ullman. Combining Top-down and Bottom-up Segmentation. In Proc. ECCV, 2004. [4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222?1239, 2001. [5] J. Djolonga and A. Krause. From MAP to Marginals: Variational Inference in Bayesian Submodular Models. In Adv. NIPS, 2014. [6] J. Djolonga and A. Krause. Scalable Variational Inference in Log-supermodular Models. In Proc. ICML, 2015. [7] S. Fujishige. Submodular Functions and Optimization. Annals of discrete mathematics. Elsevier, 2005. [8] D. Golovin and A. Krause. Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization. Journal of Artificial Intelligence Research, 42:427?486, 2011. [9] T. Hazan and T. Jaakkola. On the Partition Function and Random Maximum A-Posteriori Perturbations. In Proc. ICML, 2012. [10] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. SIAM Journal on Computing, 22(5):1087?1116, 1993. [11] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision, 82(3):302?324, 2009. [12] Andreas Krause and Daniel Golovin. Submodular function maximization. In Tractability: Practical Approaches to Hard Problems. Cambridge University Press, February 2014. [13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. ICML, 2001. [14] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In Proc. NAACL/HLT, 2011. [15] S. Nadarajah and S. Kotz. A generalized logistic distribution. International Journal of Mathematics and Mathematical Sciences, 19:3169 ? 3174, 2005. [16] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. SIAM Journal on Optimization, 19(4):1574?1609, 2009. [17] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In Proc. ICCV, 2011. [18] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In Proc. ECCV, 2008. [19] D. Tarlow, R.P. Adams, and R.S. Zemel. Randomized optimum models for structured prediction. In Proc. AISTATS, 2012. [20] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. 2003. [21] S. Tschiatschek, J. Djolonga, and A. Krause. Learning probabilistic submodular diversity models via noise contrastive estimation. In Proc. AISTATS, 2016. [22] I. Tsochantaridis, Thomas Joachims, T., Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research, 6:1453?1484, 2005. [23] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning, 1(1-2):1?305, 2008. [24]

J. Zhang, J. Djolonga, and A. Krause. Higher-order inference for multi-class log-supermodular models. In Proc. ICCV, 2015.
9